MFi Accessory Interface Specification

Release R18

Developer

Contents

1. Introduction 40
1.1 Purpose of This Specification 40
1.2 Requirements, Recommendations, and Permissions 40
1.3 Applicability 41
1.4 Terminology 41
1.4.1 Accessory, Device, and Product 41
1.4.2 Authentication Coprocessor 42
1.4.3 I2C Bus 42
1.4.4 Challenge 42
1.4.5 Challenge Response 42
1.4.6 X.509 Certificate 42
1.4.7 Component 42
1.4.8 Feature 43
1.4.9 USB Device and Host Mode 43
1.4.10 iAP 43
1.4.11 Direct User Action 43

2. General Requirements and Recommendations 44
2.1 Development Tools and Emulators 44
2.2 Reference Designs & Development Kits 44
2.3 Accessory Authentication and Accessory Identification 44
2.4 iAP1 45
2.5 iAP2 45
2.6 Detachable Connector Assemblies 45
2.7 Adapters and Proxies 45
2.8 Mixed 30-pin and Lightning Connectors 46
2.9 Mixed Headphone Jack and Lightning Connectors 48
2.10 Apple USB Power Adapters 48
2.11 Apple Device Detection 48
2.12 Multiple Simultaneous iAP2 Connections 49
2.13 Presentation of Apple Device Updates 49
2.14 Relationships Between Multiple Accessories 49
2.15 iBeacon 50
2.16 Feature Duplication 50

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2.17 Temperature Range 50
2.18 Magnetic Fields 50
2.19 Cables with USB Connectors 50
2.20 Cables with Non-USB Connectors 51
2.21 Integrated USB Receptacles 51
2.22 Integrated Non-USB Receptacles 51
2.23 User Supplied Cables and Power Supplies 52
2.24 Removable Storage 52
2.25 RF Transmission and Reception 52
2.26 TDMA Noise 53

3. Apple Authentication Coprocessor 2.0C 54
3.1 Coprocessor 2.0C Overview 54
3.2 Coprocessor 2.0C Authentication Protocol 54
3.3 Coprocessor 2.0C Signals and Pinouts 54
3.4 Coprocessor 2.0C Address Selection 55
3.5 Coprocessor 2.0C Reference Circuit 56
3.6 Coprocessor 2.0C System Voltage 56
3.7 Coprocessor 2.0C I2C Interface 57
3.7.1 I2C Startup On Power On 57
3.7.2 I2C Startup On Warm Reset 58
3.7.3 I2C Communications Process 59
3.7.4 I2C Sleep Mode 60
3.8 Coprocessor 2.0C Registers 60
3.8.1 Register Addresses 60
3.8.2 Register Descriptions 63
3.9 Coprocessor 2.0C I2C Protocol 69
3.9.1 Slave Selection and Reset 70
3.9.2 Coprocessor Busy 70
3.9.3 Writing to the Coprocessor 70
3.9.4 Reading from the Coprocessor 70
3.10 Coprocessor 2.0C Device Characteristics 71
3.10.1 Physical Configuration 71
3.10.2 Maximum Environmental Conditions 77
3.10.3 Recommended Operating Conditions 77
3.10.4 I2C Interface Characteristics 77
3.10.5 DC Electrical Characteristics 78
3.10.6 Timing Characteristics 78

4. Apple Lightning Audio Module 80

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

4.1 Overview 82
4.2 Firmware Updates 82
4.3 Authentication 82
4.4 Lightning (C10E) or Lightning (C68E) Connector 83
4.5 Mechanical 83
4.6 Pad Layout and Assignments 85
4.7 Input/Output Electrical Characteristics 87
4.8 Cable 88
4.9 Power 89
4.9.1 Power States 89
4.9.2 Module Power 90
4.9.3 Accessory Power 90
4.9.4 Device Power 91
4.10 Audio 93
4.10.1 DAC 93
4.10.2 I2S 93
4.11 Control 95
4.11.1 S0-S2 Headphone Remote Inputs 95
4.11.2 HID Headphone Remote 95
4.11.3 I2C 95
4.11.4 Serial 96
4.12 Serial Protocol 97
4.12.1 Packet Format 97
4.12.2 Initialization Packets 97
4.12.3 Configuration Packets 98
4.12.4 Operation Packets 104
4.12.5 Examples 109
4.13 Test Procedures 112
4.13.1 Connectivity 112
4.13.2 Audio Quality 112
4.13.3 Power 112
4.13.4 Charge/Sync 112
4.13.5 Firmware Update 113

5. Apple Lightning Connector 114
5.1 Comparison with the 30-pin Connector 114
5.1.1 Dimensions 114
5.1.2 Active Component 114
5.1.3 Reversability 115

Copyright © 2014 Apple Inc. All Rights Reserved.

5.1.4 Accessory Identify 115
5.1.5 Accessory Detect 115
5.1.6 Apple Device Detect 115
5.1.7 Analog Audio 115
5.1.8 Analog Video 115
5.1.9 DisplayPort Video 115
5.2 Connector Versions 115
5.3 Connector Pad/Pin Configuration 125
5.4 Connector Mechanical Requirements 131
5.4.1 All Accessories 131
5.4.2 Cable Accessories 133
5.4.3 Dock Accessories 144
5.4.4 Dongle Accessories 153
5.4.5 Form-Fitting Accessories 153
5.5 Connector Electrical Requirements 154
5.5.1 Connector Ground Pad/Pin Connection Requirements 154
5.5.2 Connector Shielding Requirements 154
5.5.3 Connector ESD Protection Requirements 154
5.5.4 Connector Signal Integrity Requirements 154
5.5.5 Lightning (C48)/Lightning (C68) Accessory Testing Requirements 155
5.6 Test Procedures 155
5.6.1 All Accessories 155
5.6.2 Lightning (C48)/Lightning (C68) Accessories 156
5.6.3 Mixed 30-pin/Lightning Connector Accessories 158
5.6.4 Cable Accessories 159
5.6.5 Dock Accessories 165
5.6.6 Dongle Accessories 170

6. Apple Lightning Receptacle 172
6.1 Lightning Receptacle (C37) Requirements 172
6.2 Lightning Receptacle (C37) Mechanical Requirements 172
6.2.1 Mounting 174
6.2.2 Receptacle Encapsulation 174
6.3 Lightning Receptacle (C37) Electrical Requirements 174
6.3.1 Pins and Assignments 176
6.4 Lightning Receptacle (C37) Example Integrations 177
6.4.1 Lightning Audio Module Accessories 177
6.4.2 Wireless Accessories 177
6.5 Test Procedures 177

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

6.5.1 Static Receptacle Opening 178
6.5.2 Receptacle Mounting Under Insertion Force 179
6.5.3 Receptacle Mounting Under Extraction Force 179
6.5.4 Bend Test 179

7. Apple Lightning Receptacle Controller 180
7.1 Mechanical 180
7.2 Pad Layout and Assignments 180
7.3 Electrical 183
7.3.1 Reference Circuit 184
7.3.2 Input/Output Electrical Characteristics 184

8. Apple 30-pin Connector 185
8.1 Connector Versions 185
8.1.1 Mid-Plane Plug 185
8.1.2 Mid-Plane Plug with Friction Latches 185
8.1.3 Mid-Plane Plug with Friction Latches, Power Only 186
8.1.4 Mid-Plane Plug (Top Shell) 186
8.1.5 Mid-Plane Plug (Bottom Shell) 186
8.1.6 Cable-End Plug 186
8.1.7 Cable-End Plug, Power Only 186
8.1.8 Cable-End Plug (Top Shell) 187
8.1.9 Cable-End Plug (Bottom Shell) 187
8.1.10 Vertical Dock Plug 187
8.1.11 Angled Dock Plug 187
8.1.12 Receptacle 188
8.2 Connector Dimensional Drawings 189

9. Accessory Authentication 201
9.1 Accessory Authentication Requirements 201
9.1.1 Accessory Authentication over iAP2 Requirements 201
9.2 Accessory Authentication Usage 202
9.2.1 Accessory Authentication over iAP2 Usage 202
9.3 Accessory Authentication Examples 203
9.3.1 Typical Accessory Authentication over iAP2 203
9.3.2 Accessory Authentication over iAP2 Failure Due To Invalid Certificate 204
9.3.3 Accessory Authentication over iAP2 Failure Due To Invalid Response 204

10. Accessory Identification 205
10.1 Accessory Identification Requirements 205

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

10.2 Accessory Identification Usage 205
10.2.1 Accessory Identification of Manufacturing Information 206
10.2.2 Accessory Identification of Sent/Received iAP2 Control Session Messages 207
10.2.3 Accessory Identification of Power Capabilities 207
10.2.4 Accessory Identification of iAP2 Transport Components 207
10.2.5 Additional Accessory Identification Parameters 208
10.3 Accessory Identification Examples 208
10.3.1 Typical Accessory Identification 208
10.3.2 Successful Accessory Identification With Two Tries 208
10.3.3 Unsuccessful Accessory Identification After Two Tries 209
10.3.4 Accessory Identification With Information Update 209

11. App Launch 210
11.1 App Launch Requirements 210
11.2 App Launch Usage 211

12. App Match 212
12.1 App Match Requirements 212
12.1.1 External Accessory Protocol 213
12.1.2 Team ID 213
12.2 App Match Usage 213
12.2.1 External Accessory Protocol 213
12.2.2 Team ID 213

13. AssistiveTouch 214
13.1 AssistiveTouch Requirements 214
13.2 AssistiveTouch Usage 215

14. Bluetooth 216
14.1 Conformity With Bluetooth Specifications 216
14.1.1 Enhanced Data Rate 216
14.1.2 Sniff Mode for Low Power Consumption 216
14.1.3 Role and Topology Management 217
14.1.4 Extended Inquiry Response 218
14.1.5 Secure Simple Pairing 218
14.2 Profiles 218
14.2.1 Device ID Profile (DID) 218
14.2.2 Hands-Free Profile (HFP) 219
14.2.3 Message Access Profile (MAP) 221
14.2.4 Audio/Video Remote Control Profile (AVRCP) 222

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

14.2.5 Advanced Audio Distribution Profile (A2DP) 224
14.3 Audio Routing 226
14.3.1 Audio Data Received via HFP Profile 226
14.3.2 Audio Data Received via A2DP Profile 227
14.4 iAP2 229
14.5 Test Procedures 230
14.5.1 Establishing a Connection 230
14.5.2 Role and Topology Management 230
14.5.3 Secure Simple Pairing (SSP) 231
14.5.4 Hands-Free Profile (HFP) 231
14.5.5 Advanced Audio Distribution Profile (A2DP) 232

15. Bluetooth Accessory Identification 233
15.1 HFP Command AT+XAPL 233

16. Bluetooth Connection 235
16.1 Bluetooth Connection Requirements 235
16.2 Bluetooth Connection Usage 236

17. Bluetooth Headset Battery Level Indication 237
17.1 HFP Command AT+IPHONEACCEV 237

18. Bluetooth Low Energy 238
18.1 Role 238
18.2 Advertising Channels 238
18.3 Advertising PDU 238
18.4 Advertising Data 238
18.5 Advertising Interval 239
18.6 Connection Parameters 240
18.7 Privacy 240
18.8 Permissions 240
18.9 Pairing 241
18.10 MTU Size 241
18.11 Services 241
18.11.1 Generic Access Profile Service 241
18.11.2 Generic Attribute Profile Service 241
18.11.3 Device Information Service 242
18.11.4 Available Services 242
18.12 GATT Server 242

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

19. CarPlay 244
19.1 Additional Specifications 244
19.2 General Requirements 245
19.2.1 Overview 245
19.2.2 Hardware Requirements 246
19.2.3 Architecture 249
19.2.4 Media Types and Formats 258
19.3 CarPlay Communication Protocol 268
19.3.1 Discovery 269
19.3.2 Setup and Control 270
19.3.3 Resource Management 287
19.3.4 Modes 290
19.3.5 User Input 295
19.3.6 Siri User Input 315
19.3.7 Commands 316
19.3.8 CarPlay Communication Plug-in 324
19.4 Examples of Modes and Mode Changes 329
19.4.1 Accessory Showing Vehicle Info 329
19.4.2 User Turns on FM Radio 330
19.4.3 Case 1: User Switches to the Apple device Music 332
19.4.4 Case 2: User Answers an Incoming Phone Call 333
19.5 Test Procedures 335

20. Cases 336
20.1 Product Design 336
20.1.1 Device Layouts and Dimensions for Apple Devices 336
20.1.2 Access to Controls 336
20.1.3 Access to the Headphone Jack and 30-pin or Lightning Connector 336
20.1.4 Device Protection 337
20.1.5 Cover Glass Contact 337
20.1.6 Dock Compatibility 337
20.2 Acoustics 337
20.2.1 Speaker and Microphone Openings 337
20.2.2 Speaker to Microphone Coupling 337
20.2.3 Call Quality 338
20.3 Sensors 338
20.3.1 Ambient Light and Proximity Sensor Interference 338
20.3.2 Magnetic Interference 338
20.3.3 Touch ID Sensor 338

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

20.4 Camera 339
20.4.1 Geometry 339
20.4.2 Color 339
20.4.3 Surface Finish 339
20.4.4 Image Degradation Examples 340
20.5 Reliability 341
20.5.1 Device Insertion and Removal 341
20.5.2 Colorfastness 341
20.6 Environmental 342
20.7 RF 342
20.7.1 Materials and Coatings 342
20.7.2 Near Field Communication (NFC) 343
20.8 Touchscreen 343
20.8.1 Overlay 343
20.8.2 Edge Swipe Gestures 343
20.9 Test Procedures 343
20.9.1 Apple Device Compatibility 343
20.9.2 Apple Device Configurations 346
20.9.3 Product Design 347
20.9.4 RF (OTA) 349

21. Device Authentication 362
21.1 Device Authentication Requirements 362
21.2 Device Authentication Usage 362
21.3 Device Authentication Examples 363
21.3.1 Typical Device Authentication 363
21.3.2 Device Authentication Failure Due To Invalid Certificate 364
21.3.3 Device Authentication Failure Due To Invalid Response 364

22. Device Notifications 365
22.1 Device Notifications Requirements 365
22.2 Device Notifications Usage 365

23. Digital Audio 366
23.1 Digital Audio Requirements 366
23.1.1 Readiness for Audio Streaming 366
23.1.2 Multiple Audio Connections 366
23.1.3 Audio Input Source Switching 367
23.1.4 Copy Protection of Digital Audio Output 367
23.1.5 USB Host Mode Audio 367

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

23.1.6 USB Device Mode Audio 369
23.2 Digital Audio Usage 369
23.2.1 USB Host Mode Audio 369
23.2.2 USB Device Mode Audio Usage 370
23.3 Digital Audio Examples 370
23.3.1 Typical USB Device Mode Digital Audio 370

24. External Accessory Protocol 372
24.1 External Accessory Protocol Requirements 372
24.1.1 iAP2 EA Session Requirements 373
24.1.2 EA Native Transport (USB Host Mode) Requirements 373
24.2 External Accessory Protocol Usage 375
24.2.1 iAP2 EA Session Usage 375
24.2.2 EA Native Transport (USB Host Mode) Usage 375
24.3 External Accessory Protocol Examples 376
24.3.1 iAP2 EA Session Example 376
24.3.2 EA Native Transport (USB Host Mode) Example 377

25. Headphones 378
25.1 Product Design 378
25.2 Audio and Data Interfaces 378
25.3 Remote Controls 379
25.4 Native Voice Recognition 379
25.5 Test Procedures 379
25.5.1 Product Design 379
25.5.2 Headphone Jack 379
25.5.3 Lightning 380
25.5.4 Bluetooth 380
25.5.5 Controls 380
25.5.6 Native Voice Recognition 381

26. Headphone Jack 382
26.1 Pin Assignments 383
26.2 Extension Cables 383
26.3 Example Circuitry 384
26.4 Test Procedures 386

27. Headphone Remote and Mic 387
27.1 Headphone Remote and Mic Requirements 387
27.2 Headphone Remote Transmitter Chip Usage 390

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

27.2.1 Transmitter Chip Pin Assignments and Physical Packaging 391
27.2.2 Transmitter Chip Maximum Voltage and Current Ratings 393
27.2.3 Transmitter Chip Thermal Impedance 393
27.2.4 Transmitter Chip Moisture Sensitivity 394
27.2.5 Transmitter Chip Electrical Characteristics 394
27.2.6 Transmitter Chip Theory of Operation 396
27.2.7 Transmitter Chip Button Mode 397
27.2.8 Transmitter Chip Tone Mode 398
27.3 Button Detection Circuitry Usage 401
27.3.1 Button Detection Circuitry Adjustments 403
27.4 Test Procedures 404
27.4.1 Product Design 404
27.4.2 Electrical 405

28. HID (Human Interface Device) 407
28.1 HID Requirements 407
28.1.1 HID Report Descriptor Requirements 408
28.1.2 HID over iAP2 Requirements 409
28.1.3 HID Native Transport (USB Host Mode) Requirements 409
28.2 HID Usage 409
28.2.1 HID over iAP2 Usage 409

29. HID Assistive Switch Control 410
29.1 HID Assistive Switch Control Requirements 410
29.2 HID Assistive Switch Control Usage 411
29.3 HID Assistive Switch Control Examples 411
29.3.1 Assistive Switch Control HID Report Descriptor 411
29.3.2 Assistive Switch Control Usage Example 412

30. HID AssistiveTouch Pointer 413
30.1 HID AssistiveTouch Pointer Requirements 413
30.2 HID AssistiveTouch Pointer Examples 414
30.2.1 AssistiveTouch HID Report Descriptor 414
30.2.2 AssistiveTouch Usage Example 415

31. HID Game Controller 416
31.1 HID Game Controller Requirements 416
31.1.1 Standard Gamepad 420
31.1.2 Extended Gamepad 421
31.1.3 Switches and Displacement Triggers 423

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

31.1.4 Labels 424
31.1.5 Face Button Group 424
31.1.6 Directional Pad 426
31.1.7 Shoulder Button 427
31.1.8 Pause Button 427
31.1.9 Joystick 428
31.1.10 LED Array 429
31.1.11 Bluetooth Button 429
31.1.12 Bluetooth Major/Minor Device Class 430
31.1.13 App Match for Controller-Enabled Games 430
31.1.14 Testing 430
31.2 HID Game Controller Examples 431
31.2.1 Standard Gamepad Example HID Report Descriptor 431
31.2.2 Extended Gamepad Example HID Report Descriptor 432
31.3 Test Procedures 435
31.3.1 General Requirements 435
31.3.2 Bluetooth Controllers 435
31.3.3 Form-Fitting Controllers 435
31.3.4 Non Form-Fitting Controllers 436
31.3.5 Control Surface Layout and Labeling 436
31.3.6 Control Surface Switches 439

32. HID Headphone Remote 441
32.1 HID Headphone Remote Requirements 441
32.2 HID Headphone Remote Examples 442
32.2.1 Headphone Remote Example HID Report Descriptor (Telephony) 442
32.2.2 Headphone Remote Example HID Report Descriptor (Media Playback) 443
32.2.3 Headphone Remote Example HID Report Descriptor (Telephony and Media Playback) 444

33. HID Keyboard 446
33.1 HID Keyboard Requirements 446
33.2 HID Keyboard Examples 452
33.2.1 Keyboard Example HID Report Descriptor 452
33.2.2 Keyboard Usage Example 454
33.3 Test Procedures 454
33.3.1 General Requirements 454

34. HID Media Playback Remote 456
34.1 HID Media Playback Remote Requirements 456
34.2 HID Media Playback Remote Examples 457

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

34.2.1 Media Playback Remote Example HID Report Descriptor 457
34.2.2 Media Playback Remote Example Usage 459

35. Location Information 460
35.1 Location Information Requirements 460
35.1.1 Global Navigation Satellite System (GNSS) Mode 461
35.1.2 Sensors Mode 462
35.1.3 NMEA Sentence Fields 462
35.2 Location Information Usage 468
35.3 Location Information Examples 469
35.3.1 GPGGA Sentence Example 469
35.3.2 GPRMC Sentence Example 469
35.3.3 GPGSV Sentence Example 469
35.3.4 GPHDT Sentence Example 469
35.3.5 PASCD Sentence Examples 470
35.3.6 PAGCD Sentence Examples 470
35.3.7 PAACD Sentence Example 471

36. Media Library Access 472
36.1 Media Library Access Requirements 472
36.1.1 Media Library Information Requirements 472
36.1.2 Media Library Updates Requirements 472
36.1.3 Media Library Playback Requirements 473
36.2 Media Library Access Usage 474
36.2.1 Media Library Information Usage 474
36.2.2 Media Library Updates Usage 474
36.2.3 Media Library Playback Usage 476

37. Musical Instrument Digital Interface (MIDI) 477

38. Now Playing Updates 478
38.1 Now Playing Updates Requirements 478
38.2 Now Playing Updates Usage 479

39. Power 480
39.1 Power Requirements 480
39.1.1 Neither Providing Power nor Drawing Power 481
39.1.2 Providing Power to the Apple Device 481
39.1.3 Drawing Power From the Apple Device 486
39.2 Power Usage 488

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

39.2.1 Providing Power to the Apple Device 488
39.2.2 Drawing Power from the Apple Device 489
39.3 Test Procedures 490
39.3.1 Accessory Power Sources 490

40. Serial 494

41. Siri 496
41.1 Enabling Custom Siri Commands 496
41.2 Obtaining Siri Availability Information 496
41.2.1 Obtaining Status Information at Connection 496
41.2.2 Receiving Siri Availability Updates from the Apple Device 497
41.3 Initiating a Siri Session 497
41.3.1 Initiating a Session from the Accessory 498
41.3.2 Initiating a Session from the Apple Device 499
41.3.3 Ending a Session from the Accessory 499
41.4 Siri Eyes Free Mode 500
41.4.1 HFP Command AT+APLEFM 500
41.5 Improving Voice Recognition 501
41.5.1 Wide Band Speech Support 502
41.6 Optimizing the Siri Experience 502
41.7 Common Siri Applications 502
41.7.1 Initialization Procedure After Connection is Established 502
41.7.2 Phone Dialing Using Siri 503
41.7.3 Audio Routing and Media Playback Using Siri 504
41.7.4 Turn-By-Turn Directions Using Siri 504
41.8 User Interaction with Siri Eyes Free in a Vehicle 504
41.9 Enabling/Disabling Siri from the Apple Device 506
41.10 Test Procedures 507
41.10.1 Siri Eyes Free 507

42. Telephony 512
42.1 Telephony Requirements 512
42.2 Telephony Usage 512

43. USB 514
43.1 USB Host Mode Passthroughs 514
43.2 USB Signal Integrity 514
43.3 Test Procedures 514
43.3.1 Signal Integrity 514

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

43.3.2 High-Speed USB 515
43.3.3 Full-Speed USB 519

44. USB Device Mode 522
44.1 USB Embedded Host Implementation 522
44.2 Power 522
44.3 Enumeration 522
44.4 iAP2 Configuration 523
44.5 HID Interface 524

45. USB Host Mode 527
45.1 iAP2 Interface Descriptor 528
45.2 iAP2 Data Transfers 528
45.3 iAP2 Performance Optimization 528

46. USB Role Switch 529
46.1 USB Role Switch Requirements 529
46.2 USB Role Switch Usage 529

47. Vehicle Status 532
47.1 Vehicle Status Requirements 532
47.2 Vehicle Status Usage 532

48. VoiceOver 534
48.1 VoiceOver Requirements 534
48.2 VoiceOver Usage 535

49. Wi-Fi Information Sharing 536
49.1 Wi-Fi Information Sharing Requirements 536
49.2 Wi-Fi Information Sharing Usage 537

50. iAP2 Link 538
50.1 Packet Structure 538
50.1.1 Start of Packet 539
50.1.2 Packet Length 539
50.1.3 Control Byte 539
50.1.4 Packet Sequence Number 540
50.1.5 Packet Acknowledgement Number 540
50.1.6 Session Identifier 541
50.1.7 Header Checksum 541

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

50.1.8 Payload Data 541
50.1.9 Payload Checksum 541
50.2 Link Synchronization Payload 542
50.2.1 Link Version 542
50.2.2 Maximum Number of Outstanding Packets 543
50.2.3 Maximum Packet Length 543
50.2.4 Retransmission Timeout 543
50.2.5 Cumulative Acknowledgement Timeout 543
50.2.6 Maximum Number of Retransmissions 544
50.2.7 Maximum Cumulative Acknowledgements 544
50.2.8 iAP2 Sessions 544
50.3 iAP2 Session Payload 544
50.4 Extended Acknowledgement Payload 544
50.5 Reset 545
50.6 Sleep 545
50.7 Operation 546
50.7.1 Record 546
50.7.2 Initialization 546
50.7.3 Synchronization 547
50.7.4 Acknowledgements 550
50.7.5 Retransmissions 550
50.7.6 Flow Control 550
50.7.7 Reset 551
50.8 Examples 552
50.8.1 Typical Link Initialization 552
50.8.2 Connection Initialization When Device is Busy 553
50.8.3 Connection Requiring Multiple Negotiation Attempts 553
50.8.4 Connection With Failed Negotiation 555
50.8.5 Normal Connection Traffic 555
50.8.6 Device Reset of Transport Connection 556
50.8.7 Cumulative Ack Timeout Expired 556
50.8.8 Continuous Data Transmission with ACKs 557
50.8.9 Resend of missing packets using EAK 558
50.8.10 Receiving Packets Out Of Order 559
50.8.11 iAP2 Link Packet Structure Example 559
50.8.12 iAP2 Link Synchronization Payload Example 560

51. iAP2 Sessions 561
51.1 Attributes 561

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

51.1.1 Type 561
51.1.2 Version 561
51.2 Control Session 562
51.2.1 Message Structure 562
51.2.2 Message Parsing 564
51.2.3 Parameter Types 564
51.2.4 Message Example 566
51.3 File Transfer Session 568
51.3.1 TransferIdentifier 568
51.3.2 Setup Datagram 569
51.3.3 Start 570
51.3.4 FirstData 570
51.3.5 FirstAndOnlyData 570
51.3.6 Data 571
51.3.7 LastData 571
51.3.8 Cancel 572
51.3.9 Pause 572
51.3.10 Success 573
51.3.11 Failure 573
51.3.12 Examples 573
51.4 External Accessory Session 575
51.4.1 Setup 576
51.4.2 ExternalAccessorySession Datagram 576

52. iAP2 Control Session Messages 577
52.1 Accessory Authentication 577
52.1.1 RequestAuthenticationCertificate 577
52.1.2 AuthenticationCertificate 577
52.1.3 RequestAuthenticationChallengeResponse 577
52.1.4 AuthenticationResponse 578
52.1.5 AuthenticationFailed 578
52.1.6 AuthenticationSucceeded 578
52.2 Accessory Identification 579
52.2.1 StartIdentification 579
52.2.2 IdentificationInformation 579
52.2.3 IdentificationAccepted 587
52.2.4 IdentificationRejected 587
52.2.5 CancelIdentification 589
52.2.6 IdentificationInformationUpdate 589

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

52.3 App Launch 590
52.3.1 RequestAppLaunch 590
52.4 AssistiveTouch 591
52.4.1 StartAssistiveTouch 591
52.4.2 StopAssistiveTouch 591
52.4.3 StartAssistiveTouchInformation 591
52.4.4 AssistiveTouchInformation 592
52.4.5 StopAssistiveTouchInformation 592
52.5 Bluetooth Connection 592
52.5.1 BluetoothComponentInformation 592
52.5.2 StartBluetoothConnectionUpdates 593
52.5.3 BluetoothConnectionUpdate 593
52.5.4 StopBluetoothConnectionUpdates 594
52.6 Device Authentication 594
52.6.1 RequestDeviceAuthenticationCertificate 594
52.6.2 DeviceAuthenticationCertificate 595
52.6.3 RequestDeviceAuthenticationChallengeResponse 595
52.6.4 DeviceAuthenticationResponse 595
52.6.5 DeviceAuthenticationFailed 596
52.6.6 DeviceAuthenticationSucceeded 596
52.7 Device Notifications 596
52.7.1 DeviceInformationUpdate 596
52.7.2 DeviceLanguageUpdate 597
52.7.3 DeviceTimeUpdate 597
52.8 External Accessory Protocol 597
52.8.1 StartExternalAccessoryProtocolSession 598
52.8.2 StopExternalAccessoryProtocolSession 598
52.8.3 StatusExternalAccessoryProtocolSession 598
52.9 Human Interface Device 599
52.9.1 StartHID 599
52.9.2 DeviceHIDReport 600
52.9.3 AccessoryHIDReport 600
52.9.4 StopHID 600
52.10 Location 601
52.10.1 StartLocationInformation 601
52.10.2 LocationInformation 602
52.10.3 StopLocationInformation 602
52.11 Media Library Access 602
52.11.1 StartMediaLibraryInformation 603

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

52.11.2 MediaLibraryInformation 603
52.11.3 StopMediaLibraryInformation 604
52.11.4 StartMediaLibraryUpdates 604
52.11.5 MediaLibraryUpdate 606
52.11.6 StopMediaLibraryUpdates 610
52.11.7 PlayMediaLibraryCurrentSelection 610
52.11.8 PlayMediaLibraryItems 610
52.11.9 PlayMediaLibraryCollection 611
52.11.10 PlayMediaLibrarySpecial 612
52.12 Now Playing 612
52.12.1 StartNowPlayingUpdates 612
52.12.2 NowPlayingUpdate 614
52.12.3 StopNowPlayingUpdates 617
52.12.4 SetNowPlayingInformation 617
52.13 Power 617
52.13.1 StartPowerUpdates 617
52.13.2 PowerUpdate 618
52.13.3 StopPowerUpdates 619
52.13.4 PowerSourceUpdate 619
52.14 USB Device Mode Audio 620
52.14.1 StartUSBDeviceModeAudio 620
52.14.2 USBDeviceModeAudioInformation 620
52.14.3 StopUSBDeviceModeAudio 621
52.15 Telephony 621
52.15.1 StartTelephonyCallStateInformation 621
52.15.2 TelephonyCallStateInformation 621
52.15.3 StopTelephonyCallStateInformation 622
52.15.4 StartTelephonyUpdates 623
52.15.5 TelephonyUpdate 623
52.15.6 StopTelephonyUpdates 625
52.16 Vehicle Status 625
52.16.1 StartVehicleStatusUpdates 625
52.16.2 VehicleStatusUpdate 625
52.16.3 StopVehicleStatusUpdates 626
52.17 VoiceOver 626
52.17.1 StartVoiceOver 626
52.17.2 StopVoiceOver 627
52.17.3 RequestVoiceOverMoveCursor 627
52.17.4 RequestVoiceOverActivateCursor 627

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

52.17.5 RequestVoiceOverScrollPage 628
52.17.6 RequestVoiceOverSpeakText 628
52.17.7 RequestVoiceOverPauseText 629
52.17.8 RequestVoiceOverResumeText 629
52.17.9 StartVoiceOverUpdates 629
52.17.10 VoiceOverUpdate 630
52.17.11 StopVoiceOverUpdates 630
52.17.12 RequestVoiceOverConfiguration 631
52.17.13 StartVoiceOverCursorUpdates 631
52.17.14 VoiceOverCursorUpdate 631
52.17.15 StopVoiceOverCursorUpdates 633
52.18 Wi-Fi Information Sharing 633
52.18.1 RequestWiFiInformation 633
52.18.2 WiFiInformation 633

53. Accessory Test System 635

54. Device Dimensional Drawings 636
54.1 iPhone 6 638
54.2 iPhone 6 Plus 639
54.3 iPhone 5s 640
54.4 iPhone 5c 641
54.5 iPhone 5 642
54.6 iPhone 4s 643
54.7 iPhone 4 (CDMA model) 644
54.8 iPhone 4 (GSM model) 645
54.9 iPhone 3G and iPhone 3GS 646
54.10 iPhone 647
54.11 iPad Air 2 Wi-Fi 648
54.12 iPad Air 2 Wi-Fi + Cellular 649
54.13 iPad mini 2 & 3 Wi-Fi 650
54.14 iPad mini 2 & 3 Wi-Fi + Cellular 651
54.15 iPad Air Wi-Fi 652
54.16 iPad Air Wi-Fi + Cellular 653
54.17 iPad mini with Wi-Fi 654
54.18 iPad mini with Wi-Fi + Cellular 655
54.19 iPad with Wi-Fi (4th generation) 656
54.20 iPad with Wi-Fi + Cellular (4th generation) 657
54.21 iPad with Wi-Fi (3rd generation) 658
54.22 iPad with Wi-Fi + 4G (3rd generation) 659

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

54.23 iPad 2 with Wi-Fi 660
54.24 iPad 2 with Wi-Fi + 3G 661
54.25 iPad with Wi-Fi 662
54.26 iPad with Wi-Fi + 3G 663
54.27 iPod touch (5th generation) 664
54.28 iPod touch (4th generation) 665
54.29 iPod touch (3rd generation) 666
54.30 iPod touch (2nd generation) 667
54.31 iPod touch 668
54.32 iPod nano (7th generation) 669
54.33 iPod nano (6th generation) 670
54.34 iPod nano (5th generation) 671
54.35 iPod nano (4th generation) 672
54.36 iPod nano (3rd generation) 673
54.37 iPod nano (2nd generation) 674
54.38 iPod nano 675
54.39 iPod classic 160GB 676
54.40 iPod classic 80GB 677
54.41 iPod (5th generation) 60GB/80GB 678
54.42 iPod (5th generation) 30GB 679
54.43 iPod (4th generation) 680
54.44 iPod (3rd generation) 681
54.45 iPod photo 30GB/60GB 682
54.46 iPod photo 683
54.47 iPod shuffle (4th generation) 684
54.48 iPod shuffle (3rd generation) 685
54.49 iPod shuffle (2nd generation) 686
54.50 iPod shuffle 687
54.51 iPod mini 689

55. Revision History 690

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figures and Tables

2. General Requirements and Recommendations 44

Figure 2-1 Mixed 30-pin and Lightning Connector USB Accessory 46
Figure 2-2 Mixed 30-pin and Lightning Connector USB Accessory with Multiplexer 47
Figure 2-3 Mixed 30-pin and Lightning Connector Serial Accessory 47
Figure 2-4 Mixed 30-pin and Lightning Connector Serial Accessory with Multiplexer 48

3. Apple Authentication Coprocessor 2.0C 54

Figure 3-1 Coprocessor pinout, top view 55
Figure 3-2 Coprocessor reference circuit diagram 56
Figure 3-3 Coprocessor I2C power on timing 58
Figure 3-4 Coprocessor I2C warm reset timing 59
Figure 3-5 Coprocessor I2C slave write address 59
Figure 3-6 Coprocessor I2C slave read address 59
Figure 3-7 Coprocessor Authentication Control and Status register, read-only bits 64
Figure 3-8 Coprocessor Authentication Control and Status register, write-only bits 65
Figure 3-9 Coprocessor Self-Test Control and Status register, write-only bits 68
Figure 3-10 Coprocessor Self-Test Control and Status register, read-only bits 68
Figure 3-11 Coprocessor package 71
Figure 3-12 Coprocessor Packing Carrier Tape 72
Figure 3-13 Coprocessor Packing Reel (1,000 unit) 73
Figure 3-14 Coprocessor Packing Reel (1,000 unit) details 74
Figure 3-15 Coprocessor Packing Reel (10,000 unit) 75
Figure 3-16 Coprocessor Packing Protective Band 76
Figure 3-17 Coprocessor typical I/O port input waveform 79
Table 3-1 Coprocessor signals 55
Table 3-2 Coprocessor address selection signals 55
Table 3-3 Coprocessor register map 60
Table 3-4 Coprocessor error codes 64
Table 3-5 Coprocessor Authentication ERR_SET values 65
Table 3-6 Coprocessor Authentication PROC_RESULTS values 65
Table 3-7 Coprocessor Authentication PROC_CONTROL values 66
Table 3-8 Coprocessor Self-Test PROC_CONTROL values 68
Table 3-9 Coprocessor Self-Test Results bits 68
Table 3-10 Coprocessor maximum electrical and temperature ranges 77

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 3-11 Coprocessor maximum electrical and temperature ranges 77
Table 3-12 Coprocessor I2C interface characteristics 77
Table 3-13 Coprocessor supply current into V_{CC}, excluding external current 78
Table 3-14 Coprocessor inputs 78
Table 3-15 Coprocessor outputs 78
Table 3-16 Coprocessor Values for typical I/O port input waveform 79

4. Apple Lightning Audio Module 80
Figure 4-1 Standard LAM Accessory without Power 81
Figure 4-2 Advanced LAM Accessory without Power 81
Figure 4-3 Lightning Audio Module Mechanical Package 84
Figure 4-4 Lightning Audio Module pickup offsets (bottom view) 85
Figure 4-5 Lightning Audio Module pad layout (top view) 86
Figure 4-6 I/O VREF RC circuit 88
Figure 4-7 Standard LAM Accessory with Power 92
Figure 4-8 Advanced LAM Accessory with Power 92
Figure 4-9 Lightning Audio Module I2S Format 94
Table 4-1 Lightning Audio Module pad assignments 86
Table 4-2 Lightning Audio Module I2S pad electrical characteristics 87
Table 4-3 Lightning Audio Module I/O pad (except I2S) electrical characteristics 88
Table 4-4 Lightning Audio Module Serial RX electrical characteristics 88
Table 4-5 Lightning Audio Module serial protocol packet 97
Table 4-6 Lightning Audio Module accConfigurationInformation packet payload 98
Table 4-7 Lightning Audio Module accNameInformation packet payload 99
Table 4-8 Lightning Audio Module accManufacturerInformation packet payload 100
Table 4-9 Lightning Audio Module accModelInformation packet payload 100
Table 4-10 Lightning Audio Module accSerialNumberInformation packet payload 100
Table 4-11 Lightning Audio Module accPreferredAppBundleIdentifierInformation packet payload 101
Table 4-12 Lightning Audio Module accExternalAccessoryProtocolNameInformation packet payload 101
Table 4-13 Lightning Audio Module accHIDComponentInformationInformation packet payload 101
Table 4-14 Lightning Audio Module accAudioTerminalInformation packet payload 102
Table 4-15 Lightning Audio Module Audio Output Terminal Types 103
Table 4-16 Lightning Audio Module Audio Input Terminal Types 103
Table 4-17 Lightning Audio Module Audio Terminal Gain/Attenuation Step Size Values 103
Table 4-18 Preferred Apple Device Audio Processing Options 104
Table 4-19 Lightning Audio Module lamInformation packet payload 104
Table 4-20 Lightning Audio Module Device Power State Values 105
Table 4-21 Lightning Audio Module Audio Input/Output Stream Types and Processing Requirements 106
Table 4-22 Lightning Audio Module lamAudioTerminalInformation packet payload 106

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 4-23 Lightning Audio Module lamExternalAccessoryProtocolSessionData packet payload 107
Table 4-24 Lightning Audio Module lamHIDReport packet payload 107
Table 4-25 Lightning Audio Module accExternalAccessoryProtocolSessionData packet payload 107
Table 4-26 Lightning Audio Module accHIDReport packet payload 107
Table 4-27 Lightning Audio Module accAudioTerminalStateInformation packet payload 108
Table 4-28 Lightning Audio Module accPassthroughPowerSourceInformation packet payload 109

5. Apple Lightning Connector 114
Figure 5-1 Comparison of 30-pin connector and Lightning connector dimensions 114
Figure 5-2 Lightning (C10) 117
Figure 5-3 Lightning (C11) 117
Figure 5-4 Lightning (C12) 118
Figure 5-5 Lightning (C48) 118
Figure 5-6 Lightning (C68) 119
Figure 5-7 Lightning (C10) dimensions 120
Figure 5-8 Lightning (C11) dimensions 121
Figure 5-9 Lightning (C12) dimensions 122
Figure 5-10 Lightning (C48) dimensions 123
Figure 5-11 Lightning (C68) dimensions 124
Figure 5-12 C10/C11 Connector Pads 126
Figure 5-13 C12 Connector Pins 126
Figure 5-14 C48 Connector Pads 127
Figure 5-15 C68 Connector Pads 128
Figure 5-16 Connector critical areas 132
Figure 5-17 Connector plug engagement 132
Figure 5-18 Connector plug exposure 133
Figure 5-19 Connector cable enclosure 133
Figure 5-20 Allowable USB-A Lightning Cable Integrations 134
Figure 5-21 Allowable USB-B Lightning Cable Integrations 135
Figure 5-22 Example Non-USB Lightning Cable Integration 135
Figure 5-23 Cable EMC Diagram 136
Figure 5-24 Lightning (C48) Cable Encapsulation Diagram 137
Figure 5-25 Lightning (C48) Cable Shielding Diagram 138
Figure 5-26 Lightning (C48) Cable Shield Laser Welding Diagram 139
Figure 5-27 Lightning (C48) Recommended Cable Shield, Crimp 140
Figure 5-28 Lightning (C48) Recommended Cable Shield, No Crimp 141
Figure 5-29 Lightning (C68) Recommended Cable Shield, Crimp 142
Figure 5-30 Lightning (C68) Recommended Cable Shield, No Crimp 143
Figure 5-31 iPhone/iPod touch/iPod nano backstop 145

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 5-32 iPhone/iPod touch flexible mechanism 145
Figure 5-33 iPhone/iPod touch contact 146
Figure 5-34 iPhone/iPod touch sideways angle 146
Figure 5-35 iPad backstop 147
Figure 5-36 iPad flexible mechanism 148
Figure 5-37 iPod nano flexible mechanism 149
Figure 5-38 Lightning (C48) Dock Shielding Diagram 149
Figure 5-39 Lightning (C48) Dock Shield Laser Welding Diagram 150
Figure 5-40 Lightning (C48) Recommended Dock Shield Top 151
Figure 5-41 Lightning (C48) Recommended Dock Shield Bottom 152
Figure 5-42 Dongle Maximum Allowable Force 153
Figure 5-43 Identifying Lightning (C10), Lightning (C11), or Lightning (C12) connector configuration 155
Figure 5-44 Signal Integrity Test Setup for USB cables 159
Figure 5-45 Differential impedance graph with limit lines 160
Figure 5-46 Differential insertion loss graph with limit lines 161
Figure 5-47 DCR Ground in Parallel with Shield Conductor Test Setup ($R_{GND||SHIELD}$) 162
Figure 5-48 DCR VBUS Conductor Test Setup (R_{VBUS}) 162
Figure 5-49 DCR Ground Conductor Test Setup (R_{GND}) 162
Figure 5-50 Bend Test Setup for Lightning Cable Shield 164
Figure 5-51 Bend Test Setup for Lightning Cables 164
Figure 5-52 Tester Setup for Lightning Cables 165
Figure 5-53 180° Cable Bend Test Setup for Lightning Cables 165
Figure 5-54 360° Cable Bend Test Setup for Lightning Cables 165
Figure 5-55 Bend Test Clamp Region for Lightning Dock Shield 166
Figure 5-56 Bend Test Setup for Lightning Dock Shield 166
Figure 5-57 Fitting mechanism to enable testing of certain docks that limit device access 168
Figure 5-58 Dongle testing using a force gauge 171
Table 5-1 Lightning connector modules 116
Table 5-2 Lightning (C10) connector configurations 129
Table 5-3 Lightning (C11) connector configurations 129
Table 5-4 Lightning (C12) connector configurations 130
Table 5-5 Lightning (C48) connector configurations 130
Table 5-6 Lightning (C68) connector configurations 130

6. Apple Lightning Receptacle 172
Figure 6-1 Lightning Receptacle (C37) Dimensions 173
Figure 6-2 Lightning Receptacle (C37) recommended PCB layout 176
Figure 6-3 Lightning Receptacle (C37) Mechanical Test Plug 178
Table 6-1 Lightning Receptacle (C37) pins 176

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

7. Apple Lightning Receptacle Controller 180
Figure 7-1 Lightning Receptacle Controller pad layout (top view) 181
Figure 7-2 Lightning Receptacle Controller reference circuit 184
Table 7-1 Lightning Receptacle Controller pad assignments 181

8. Apple 30-pin Connector 185
Figure 8-1 Mid-Plane Plug 189
Figure 8-2 Mid-Plane Plug, with Friction Latches 190
Figure 8-3 Mid-Plane Plug with Friction Latches, Power Only 191
Figure 8-4 Top Shell for Mid-Plane Plugs 192
Figure 8-5 Bottom Shell for Mid-Plane Plugs 193
Figure 8-6 Cable-End Plug 194
Figure 8-7 Cable-End Plug, Power Only 195
Figure 8-8 Top Shell for Cable End Plugs 196
Figure 8-9 Bottom Shell for Cable End Plugs 197
Figure 8-10 Vertical Dock Plug 198
Figure 8-11 Angled Dock Plug 199
Figure 8-12 Receptacle 200

11. App Launch 210
Figure 11-1 App Launch Alert 210

12. App Match 212
Figure 12-1 App Match Alert 212

13. AssistiveTouch 214
Figure 13-1 AssistiveTouch Pointer 214

14. Bluetooth 216
Figure 14-1 Initiate Audio Playback (e.g. music) 228
Figure 14-2 Initiate System Sound (e.g. turn-by-turn directions) 228
Table 14-1 SubBand Codec Information Elements for Apple products 224
Table 14-2 MPEG-2/4 AAC Codec Information Elements for Apple devices 225
Table 14-3 AAC audio packet for Apple devices 225

19. CarPlay 244
Figure 19-1 Topology of an automotive head unit using CarPlay 246
Figure 19-2 Process for establishing a CarPlay session 251
Figure 19-3 Telephony Audio Round-Trip Path 263
Figure 19-4 Audio Tolerance Mask For Send Frequency Response (Sensitivity (dB) vs. Frequency (Hz)) 265

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 19-5 Event timeline for audio ducking/unducking 268
Figure 19-6 Mixing CarPlay audio 288
Figure 19-7 Touchscreen 296
Figure 19-8 Touchpad 296
Figure 19-9 Example Input Report Layout for Single-Touch Screen 299
Figure 19-10 Example Input Report Layout for Multi-Touch Screen 301
Figure 19-11 Example Input Report Layout for Single-Touch Touchpad with Basic Character Gesture Recognition 304
Figure 19-12 Example Input Report Layout for Single-Touch Touchpad with Character Gesture Recognition with Alternate Interpretations 308
Figure 19-13 Multi-axis Controller 308
Figure 19-14 Example Input Report Layout for Multi-Axis Controller 311
Figure 19-15 Example Input Report Layout for Simple Media Buttons 313
Figure 19-16 Example Input Report Layout for Simple Telephone Buttons 315
Figure 19-17 CarPlay Communication Plug-in Reference Architecture 325
Figure 19-18 Control sequence to initiate a CarPlay session 326
Figure 19-19 Control sequence for switching the main audio channel from accessory to Apple device 327
Figure 19-20 Control sequence for returning main audio from Apple device to accessory 328
Figure 19-21 Control sequence for closing a CarPlay session 329
Table 19-1 USB NCM Control Interface Descriptor 252
Table 19-2 USB NCM Communication Interface Descriptor Requirements 253
Table 19-3 USB NCM Data Interface Descriptor 253
Table 19-4 CarPlay H.264 Video Stream Standard Resolutions 259
Table 19-5 CarPlay H.264 Levels 259
Table 19-6 Entertainment Stream Requirements 261
Table 19-7 Telephony Audio Stream Requirements 262
Table 19-8 FaceTime Audio Stream Requirements 263
Table 19-9 Audio Tolerance Mask Limits For Send Frequency Response 264
Table 19-10 Siri Stream Requirements 266
Table 19-11 Alternate Audio Stream Requirements 267
Table 19-12 CarPlay Bonjour Service Keys 269
Table 19-13 Status flags bitfield enum values 269
Table 19-14 Encryption type enum values 272
Table 19-15 Info Message request keys 273
Table 19-16 Info Message response keys 273
Table 19-17 Audio format keys 276
Table 19-18 Audio formats bitfield enum values 277
Table 19-19 Audio latencies keys 278
Table 19-20 Display keys 278

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 19-21 Display features bitfield enum values 279
Table 19-22 Limited UI elements string values 279
Table 19-23 Initial Setup request keys 280
Table 19-24 Initial Setup response keys 281
Table 19-25 Setup request keys 281
Table 19-26 Setup response keys 282
Table 19-27 Main/alternate audio stream descriptors request keys 282
Table 19-28 Main/alternate audio stream descriptors response keys 283
Table 19-29 Screen stream descriptors request keys 283
Table 19-30 Screen stream descriptors response keys 283
Table 19-31 Stream ID enum values 283
Table 19-32 Main Audio type string values 284
Table 19-33 Channels 284
Table 19-34 Feedback response keys 285
Table 19-35 Feedback stream keys 285
Table 19-36 URLs for CarPlay 285
Table 19-37 Entity enum values 290
Table 19-38 Resource ID enum values 290
Table 19-39 Resource constraint enum values 291
Table 19-40 Resource ownership transfer type enum values 291
Table 19-41 Resource transfer priority enum values 291
Table 19-42 App state enum values 292
Table 19-43 Speech mode enum values 292
Table 19-44 HID device description keys 295
Table 19-45 Digitizer Support HID Usages 297
Table 19-46 Character Input Gesture Support HID Usages 302
Table 19-47 Knob Support HID Usages 309
Table 19-48 Button Support HID Usages 311
Table 19-49 Siri actions enum values 315
Table 19-50 duckAudio request keys 317
Table 19-51 unduckAudio request keys 318
Table 19-52 disableBluetooth request keys 318
Table 19-53 changeModes request keys 319
Table 19-54 appState request dictionary request keys 319
Table 19-55 resource request dictionary request keys 319
Table 19-56 changeModes response keys 320
Table 19-57 modesChanged request keys 320
Table 19-58 appState dictionary keys 320
Table 19-59 resource dictionary keys 320

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 19-60 hidSendReport request keys 321
Table 19-61 hidSetInputMode request keys 321
Table 19-62 HID input modes enum values 321
Table 19-63 requestSiri request keys 322
Table 19-64 Accessory to Apple device requestUI request keys 323
Table 19-65 setNightMode request keys 323
Table 19-66 setLimitedUI request keys 323

20. Cases 336
Figure 20-1 Image degradation by color shifting 340
Figure 20-2 Image degradation by decrease of contrast 340
Figure 20-3 Image degradation by blocking 341
Figure 20-4 Image degradation by flash interference 341
Figure 20-5 Apple Device Proudness Test 349
Figure 20-6 Apple Device Gap Test 349
Figure 20-7 Apple Device Setup 350
Figure 20-8 Generic Setup for TRP/TIS(Free Space) 351
Figure 20-9 iPad with Post-it Notes 352
Table 20-1 Required device configurations for accessory case test procedures 346
Table 20-2 EIRP/EIS Measurement Matrix for case OTA testing procedure 353
Table 20-3 TRP/TIS Measurement Matrix for case OTA testing procedure 354
Table 20-4 Required Apple device models for testing of specific OTA bands 356

23. Digital Audio 366
Table 23-1 Audio Transport Connection Actions 367

24. External Accessory Protocol 372
Table 24-1 USB Interface Interface Descriptor (Alternate Setting 1 - Active Session) for External Accessory Native Transport (USB Host Mode) 374
Table 24-2 USB Interface Descriptor (Alternate Setting 0 - Zero Bandwidth) for External Accessory Native Transport (USB Host Mode) 374

26. Headphone Jack 382
Figure 26-1 Headphone/Microphone Plug Dimensional and Flange Details 382
Figure 26-2 Example Headphone Extension Cable 384
Figure 26-3 Typical circuitry in a headphone/microphone accessory 385
Table 26-1 Headphone Plug Pin Assignments 383
Table 26-2 Recommended component values for typical circuitry in a headphone/microphone accessory 385

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

27. Headphone Remote and Mic 387
Figure 27-1 Example Headphone Remote and Mic System A 388
Figure 27-2 Example Headphone Remote and Mic System B 389
Figure 27-3 Example Headphone Remote and Mic System C 390
Figure 27-4 Transmitter Chip Package 392
Figure 27-5 Transmitter Chip Block Diagram 397
Figure 27-6 Transmitter Chip Startup Timing 399
Figure 27-7 Tone Mode ACK Sequence 400
Figure 27-8 Tone Transmit/Decode Method 400
Figure 27-9 Transmitter Circuit 402
Figure 27-10 Microphone Circuit 402
Figure 27-11 Headphones Looped Around Ear Test Setup 405
Figure 27-12 Headphone Remote and Mic Test Setup 406
Table 27-1 Transmitter Chip Part Numbers 391
Table 27-2 Transmitter Chip Pin Assignments 391
Table 27-3 Transmitter Chip Package Dimensions 392
Table 27-4 Transmitter Chip Maximum Voltage and Current Ratings 393
Table 27-5 Transmitter Chip Electrical Characteristics (General) 394
Table 27-6 Transmitter Chip Electrical Characteristics (Tone Mode) 395
Table 27-7 Transmitter Chip Electrical Characteristics (Button Mode) 396
Table 27-8 Transmitter Chip DETECT Pin Voltages 397
Table 27-9 Transmitter Circuit Components 403
Table 27-10 Approved transmitter circuit MEMS digital microphone components 404
Table 27-11 Headphone Remote and Mic Expected mic_bias Voltages in Button Mode 406

29. HID Assistive Switch Control 410
Figure 29-1 HID Assistive Switch Control 410

31. HID Game Controller 416
Figure 31-1 Form-Fitting Standard Gamepad Sample 417
Figure 31-2 Form-Fitting Extended Gamepad Sample 417
Figure 31-3 Non Form-Fitting Extended Gamepad Sample 418
Figure 31-4 iPad mini Form-Fitting Extended Gamepad Sample 419
Figure 31-5 HID standard gamepad control layout 421
Figure 31-6 HID extended gamepad control layout 423
Figure 31-7 HID game controller face button group layout 425
Figure 31-8 HID game controller face button group layout (alternate) 426
Figure 31-9 HID gamepad directional pad layout 427
Figure 31-10 HID game controller shoulder button 427
Figure 31-11 HID game controller pause button 428

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 31-12 HID game controller joystick 429
Table 31-1 HID Game Controller pressure sensitive switch mechanical requirements 424
Table 31-2 HID Game Controller face button group color definitions 425
Table 31-3 HID Game Controller App Match Parameters 430
Table 31-4 Required Game Controller Control Surfaces 436

32. HID Headphone Remote 441
Table 32-1 HID Consumer Page controls for use by headphone remotes 441
Table 32-2 HID Telephony Page controls for use by headphone remotes 442

33. HID Keyboard 446
Table 33-1 Required HID Keyboard/Keypad Page controls for use by keyboard components 447
Table 33-2 Required HID Keyboard/Keypad Page controls for use by JIS keyboard components 450
Table 33-3 Optional HID Keyboard/Keypad Page controls for use by keyboard components 450
Table 33-4 HID Consumer Page controls for use by keyboard components 451

34. HID Media Playback Remote 456
Table 34-1 HID Consumer Page controls for use by media playback remote components 456

35. Location Information 460
Table 35-1 Supported NMEA Sentences and Maximum/Recommended Rates 460
Table 35-2 GPGGA Sentence Fields 463
Table 35-3 GPRMC Sentence Fields 463
Table 35-4 GPGSV Sentence Fields 464
Table 35-5 GPHDT Sentence Fields 465
Table 35-6 PASCD Sentence Fields 466
Table 35-7 PAGCD Sentence Fields 467
Table 35-8 PAACD Sentence Fields 467

39. Power 480
Figure 39-1 Typical AC adapter diode bridge circuit 483
Figure 39-2 USB D+/D- resistor networks for power-providing accessory connectors that do not implement iAP2 485
Table 39-1 Typical component values for an AC adapter diode bridge circuit 483
Table 39-2 USB cable maximum DC resistances 484
Table 39-3 USB D+/D- resistor values for power-providing accessory connectors that do not implement iAP2 485
Table 39-4 Maximum allowable Low Power Mode current draw 487
Table 39-5 USB Events that permit Intermittent High Power Mode 489
Table 39-6 USB Events that exit Intermittent High Power Mode 490

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 39-7 Power Supply Voltage Outputs 490

40. Serial 494
Table 40-1 Serial transport mark and space levels for Apple devices 494

41. Siri 496
Figure 41-1 Siri is Disabled/Enabled from the Apple Device's Settings 497
Figure 41-2 Initiating a Siri Session from the Accessory 498
Figure 41-3 Initiating a Siri Session from the Apple Device 499
Figure 41-4 Ending a Siri Session from the Accessory 500
Figure 41-5 Siri Initialization Procedure 503
Figure 41-6 Siri Initialization Procedure with Siri Eyes Free 503
Figure 41-7 Siri Eyes Free User Interaction 505
Figure 41-8 Siri is Deactivated - Launching Voice Control 506
Figure 41-9 Siri is Deactivated - Displaying a Warning Message 506

43. USB 514
Figure 43-1 High-Speed Test Environment 515
Figure 43-2 USB Test Points 516
Figure 43-3 Eye Diagram Template at TP2 516
Figure 43-4 TP2 Eye Diagram Mask 519
Figure 43-5 Full-Speed Test Environment 520
Table 43-1 Eye Diagram Template Specifications 517
Table 43-2 Pass/Fail Criteria for Host Under Test 517
Table 43-3 Test Packet Layout 517
Table 43-4 Full-Speed Signal Quality Requirements 520

44. USB Device Mode 522
Figure 44-1 USB Device Mode Interface Descriptor 523
Figure 44-2 USB Device Mode Interface HID Report 525
Figure 44-3 USB Device Mode Interface Report Packing 526
Table 44-1 Link control byte usage 525

45. USB Host Mode 527
Table 45-1 USB Host Mode iAP2 interface descriptor 528

46. USB Role Switch 529
Figure 46-1 USB Role Switch Custom Vendor Request (Apple device is USB Device, Accessory is USB Host) 530
Table 46-1 USB Vendor Request for Apple Device to Host Mode Switch 530

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 46-2 USB Vendor Request for Apple Device to Host Mode Switch: Valid wValues 531

49. Wi-Fi Information Sharing 536
Figure 49-1 Wi-Fi Information Sharing Alert 536

50. iAP2 Link 538
Table 50-1 iAP2 Link Packet Structure 538
Table 50-2 iAP2 Link Control Byte Bits 540
Table 50-3 Link Synchronization Payload (Version 1) 542
Table 50-4 EAK Packet Payload (Link v1) 545
Table 50-5 iAP2 Link Operation Record Variables 546
Table 50-6 Default link parameters during synchronization 548
Table 50-7 Suggested link parameters for USB Host Mode transport (Full Speed) 548
Table 50-8 Suggested link parameters for USB Device Mode transport (Full Speed) 549
Table 50-9 Suggested link parameters for Bluetooth transport 549
Table 50-10 Suggested link parameters for 57.6 kbps serial transport 549
Table 50-11 iAP2 Link Packet Structure Example - Accessory SYN Packet 559
Table 50-12 Link Synchronization Payload Example 560

51. iAP2 Sessions 561
Figure 51-1 Control Session Message Structure 563
Figure 51-2 Control Session Message Parameter Structure 564
Figure 51-3 Group Parameter Structure 566
Figure 51-4 StartExternalAccessoryProtocolSession Control Session Message Example 567
Figure 51-5 ExternalAccessoryProtocolIdentifier Parameter 567
Figure 51-6 ExternalAccessoryProtocolSessionIdentifier Parameter 567
Figure 51-7 StartNowPlayingUpdates Control Session Message Example 568
Figure 51-8 PlaybackAttributes Parameter Group 568
Table 51-1 iAP2 Session Types 561
Table 51-2 File Transfer Session Setup Datagram 569
Table 51-3 File Transfer Session Start Datagram 570
Table 51-4 File Transfer Session FirstData Datagram 570
Table 51-5 File Transfer Session FirstAndOnlyData Datagram 571
Table 51-6 File Transfer Session Data Datagram 571
Table 51-7 File Transfer Session LastData Datagram 571
Table 51-8 File Transfer Session Cancel Datagram 572
Table 51-9 File Transfer Session Pause Datagram 572
Table 51-10 File Transfer Session Success Datagram 573
Table 51-11 File Transfer Session Failure Datagram 573
Table 51-12 ExternalAccessorySession Datagram 576

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

52. iAP2 Control Session Messages 577

Table 52-1 RequestAuthenticationCertificate message parameters 577
Table 52-2 AuthenticationCertificate message parameters 577
Table 52-3 RequestAuthenticationChallengeResponse message parameters 578
Table 52-4 AuthenticationResponse message parameters 578
Table 52-5 AuthenticationFailed message parameters 578
Table 52-6 AuthenticationResponseSucceeded message parameters 578
Table 52-7 StartIdentification message parameters 579
Table 52-8 IdentificationInformation message parameters 579
Table 52-9 PowerProvidingCapability enum 581
Table 52-10 ExternalAccessoryProtocol parameter group 581
Table 52-11 MatchAction enum 582
Table 52-12 USBDeviceTransportComponent parameter group 582
Table 52-13 USBDeviceModeAudioSampleRate enum 582
Table 52-14 USBHostTransportComponent parameter group 583
Table 52-15 SerialTransportComponent parameter group 583
Table 52-16 BluetoothTransportComponent parameter group 584
Table 52-17 iAP2HIDComponent parameter group 584
Table 52-18 VehicleInformationComponent parameter group 584
Table 52-19 EngineTypes enum 584
Table 52-20 VehicleStatusComponent parameter group 585
Table 52-21 LocationInformationComponent parameter group 585
Table 52-22 USBHostHIDComponent parameter group 586
Table 52-23 HIDComponentFunction enum 586
Table 52-24 IdentificationAccepted message parameters 587
Table 52-25 IdentificationRejected message parameters 587
Table 52-26 CancelIdentification message parameters 589
Table 52-27 IdentificationInformationUpdate message parameters 589
Table 52-28 RequestAppLaunch message parameters 590
Table 52-29 AppLaunchMethod enum 590
Table 52-30 StartAssistiveTouch message parameters 591
Table 52-31 StopAssistiveTouch message parameters 591
Table 52-32 StartAssistiveTouchInformation message parameters 591
Table 52-33 AssistiveTouchInformation message parameters 592
Table 52-34 StopAssistiveTouchInformation message parameters 592
Table 52-35 BluetoothComponentInformation message parameters 592
Table 52-36 BluetoothComponentStatus parameter group 593
Table 52-37 StartBluetoothConnectionUpdates message parameters 593
Table 52-38 BluetoothConnectionUpdate message parameters 593

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 52-39 BluetoothComponentProfiles parameter group 593
Table 52-40 StopBluetoothConnectionUpdates message parameters 594
Table 52-41 RequestDeviceAuthenticationCertificate message parameters 595
Table 52-42 DeviceAuthenticationCertificate message parameters 595
Table 52-43 RequestDeviceAuthenticationChallengeResponse message parameters 595
Table 52-44 DeviceAuthenticationResponse message parameters 595
Table 52-45 DeviceAuthenticationFailed message parameters 596
Table 52-46 DeviceAuthenticationResponseSucceeded message parameters 596
Table 52-47 DeviceInformationUpdate message parameters 596
Table 52-48 DeviceLanguageUpdate message parameters 597
Table 52-49 DeviceTimeUpdate message parameters 597
Table 52-50 StartExternalAccessoryProtocolSession message parameters 598
Table 52-51 StopExternalAccessoryProtocolSession message parameters 598
Table 52-52 StatusExternalAccessoryProtocolSession message parameters 598
Table 52-53 ExternalAccessoryProtocolSessionStatus enum 599
Table 52-54 StartHID message parameters 599
Table 52-55 DeviceHIDReport message parameters 600
Table 52-56 AccessoryHIDReport message parameters 600
Table 52-57 StopHID message parameters 600
Table 52-58 StartLocationInformation message parameters 601
Table 52-59 LocationInformation message parameters 602
Table 52-60 StopLocationInformation message parameters 602
Table 52-61 StartMediaLibraryInformation message parameters 603
Table 52-62 MediaLibraryInformation message parameters 603
Table 52-63 MediaLibraryInformation parameter group 603
Table 52-64 MediaLibraryType enum 603
Table 52-65 StopMediaLibraryInformation message parameters 604
Table 52-66 StartMediaLibraryUpdates message parameters 604
Table 52-67 MediaItemProperties parameter group 604
Table 52-68 MediaPlaylistProperties parameter group 605
Table 52-69 MediaLibraryUpdate message parameters 606
Table 52-70 MediaItem parameter group 607
Table 52-71 MediaPlaylist parameter group 609
Table 52-72 MediaType enum 609
Table 52-73 StopMediaLibraryUpdate message parameters 610
Table 52-74 PlayMediaLibraryCurrentSelection message parameters 610
Table 52-75 PlayMediaLibraryItems message parameters 610
Table 52-76 PlayMediaLibraryCollection message parameters 611
Table 52-77 MediaLibraryCollectionType enum 611

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 52-78 PlayMediaLibrarySpecial message parameters 612
Table 52-79 StartNowPlayingUpdates message parameters 612
Table 52-80 StartNowPlayingMediaItemAttributes parameter group 612
Table 52-81 StartNowPlayingPlaybackAttributes parameter group 613
Table 52-82 NowPlayingUpdate message parameters 614
Table 52-83 PlaybackAttributes parameter group 615
Table 52-84 PlaybackStatus enum 616
Table 52-85 PlaybackShuffle enum 616
Table 52-86 PlaybackRepeat enum 616
Table 52-87 StopNowPlayingUpdates message parameters 617
Table 52-88 SetNowPlayingInformation message parameters 617
Table 52-89 StartPowerUpdates message parameters 618
Table 52-90 PowerUpdate message parameters 618
Table 52-91 AccessoryPowerModes enum 619
Table 52-92 BatteryChargingState enum 619
Table 52-93 StopPowerUpdates message parameters 619
Table 52-94 PowerSourceUpdate message parameters 620
Table 52-95 StartUSBDeviceModeAudio message parameters 620
Table 52-96 USBDeviceModeAudioInformation message parameters 620
Table 52-97 StopUSBDeviceModeAudio message parameters 621
Table 52-98 StartTelephonyCallStateInformation message parameters 621
Table 52-99 TelephonyCallStateInformation message parameters 621
Table 52-100 CallStateStatusValue enum 622
Table 52-101 CallStateDirectionValue enum 622
Table 52-102 StopTelephonyCallStateInformation message parameters 623
Table 52-103 StartTelephonyUpdate message parameters 623
Table 52-104 TelephonyUpdate message parameters 623
Table 52-105 TelephonySignalStrengthValue enum 624
Table 52-106 TelephonyRegistrationStatusValue enum 624
Table 52-107 StopTelephonyUpdate message parameters 625
Table 52-108 StartVehicleStatusUpdates message parameters 625
Table 52-109 VehicleStatusUpdate message parameters 626
Table 52-110 StopVehicleStatusUpdates message parameters 626
Table 52-111 StartVoiceOver message parameters 626
Table 52-112 StopVoiceOver message parameters 627
Table 52-113 RequestVoiceOverMoveCursor message parameters 627
Table 52-114 VoiceOverCursorDirection enum 627
Table 52-115 RequestVoiceOverActivateCursor message parameters 628
Table 52-116 RequestVoiceOverScrollPage message parameters 628

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 52-117 `VoiceOverScrollDirection` enum 628
Table 52-118 `RequestVoiceOverSpeakText` message parameters 629
Table 52-119 `RequestVoiceOverPauseText` message parameters 629
Table 52-120 `RequestVoiceOverResumeText` message parameters 629
Table 52-121 `StartVoiceOverUpdates` message parameters 630
Table 52-122 `VoiceOverUpdate` message parameters 630
Table 52-123 `StopVoiceOverUpdates` message parameters 630
Table 52-124 `RequestVoiceOverConfiguration` message parameters 631
Table 52-125 `StartVoiceOverCursorUpdates` message parameters 631
Table 52-126 `VoiceOverCursorUpdate` message parameters 632
Table 52-127 VoiceOverCursorUpdate Traits 632
Table 52-128 `StopVoiceOverCursorUpdates` message parameters 633
Table 52-129 `RequestWiFiInformation` message parameters 633
Table 52-130 `WiFiInformation` message parameters 634
Table 52-131 `WiFiRequestStatus` enum 634

54. Device Dimensional Drawings 636
Figure 54-1 iPhone 6 Dimensional Drawing 638
Figure 54-2 iPhone 6 Plus Dimensional Drawing 639
Figure 54-3 iPhone 5s Dimensional Drawing 640
Figure 54-4 iPhone 5c Dimensional Drawing 641
Figure 54-5 iPhone 5 Dimensional Drawing 642
Figure 54-6 iPhone 4s Dimensional Drawing 643
Figure 54-7 iPhone 4 CDMA Dimensional Drawing 644
Figure 54-8 iPhone 4 GSM Dimensional Drawing 645
Figure 54-9 iPhone 3G and iPhone 3GS Dimensional Drawing 646
Figure 54-10 iPhone Dimensional Drawing 647
Figure 54-11 iPad Air 2 Wi-Fi Dimensional Drawing 648
Figure 54-12 iPad Air 2 Wi-Fi + Cellular Dimensional Drawing 649
Figure 54-13 iPad mini 2 & iPad mini 3 Wi-Fi Dimensional Drawing 650
Figure 54-14 iPad mini 2 & iPad mini 3 Wi-Fi + Cellular Dimensional Drawing 651
Figure 54-15 iPad Air Wi-Fi Dimensional Drawing 652
Figure 54-16 iPad Air Wi-Fi + Cellular Dimensional Drawing 653
Figure 54-17 iPad mini with Wi-Fi Dimensional Drawing 654
Figure 54-18 iPad mini Wi-Fi + Cellular Dimensional Drawing 655
Figure 54-19 iPad Wi-Fi (4th generation) Dimensional Drawing 656
Figure 54-20 iPad Wi-Fi + Cellular (4th generation) Dimensional Drawing 657
Figure 54-21 iPad Wi-Fi (3rd Generation) Dimensional Drawing 658
Figure 54-22 iPad Wi-Fi + 4G (3rd Generation) Dimensional Drawing 659

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 54-23 iPad 2 Wi-Fi Dimensional Drawing 660
Figure 54-24 iPad 2 Wi-Fi + 3G Dimensional Drawing 661
Figure 54-25 iPad Wi-Fi Dimensional Drawing 662
Figure 54-26 iPad Wi-Fi + 3G Dimensional Drawing 663
Figure 54-27 iPod touch 5th gen. Dimensional Drawing 664
Figure 54-28 iPod touch 4th gen. Dimensional Drawing 665
Figure 54-29 iPod touch 3rd gen. Fall '09 32GB and 64GB Dimensional Drawing 666
Figure 54-30 iPod touch 2nd gen. 8GB, 16GB, 32GB Dimensional Drawing 667
Figure 54-31 iPod touch Dimensional Drawing 668
Figure 54-32 iPod nano 7th gen. Dimensional Drawing 669
Figure 54-33 iPod nano 6th gen. Dimensional Drawing 670
Figure 54-34 iPod nano 5th gen. Dimensional Drawing 671
Figure 54-35 iPod nano 4th gen. Dimensional Drawing 672
Figure 54-36 iPod nano 3rd gen. Dimensional Drawing 673
Figure 54-37 iPod nano 2nd gen. Dimensional Drawing 674
Figure 54-38 iPod nano Dimensional Drawing 675
Figure 54-39 iPod classic 160GB Dimensional Drawing 676
Figure 54-40 iPod classic 80GB Dimensional Drawing 677
Figure 54-41 iPod 5th gen. 60GB/80GB Dimensional Drawing 678
Figure 54-42 iPod 5th gen. 30GB Dimensional Drawing 679
Figure 54-43 iPod 4th gen. Dimensional Drawing 680
Figure 54-44 iPod 3rd gen. Dimensional Drawing 681
Figure 54-45 iPod photo 30/60GB Dimensional Drawing 682
Figure 54-46 iPod photo Dimensional Drawing 683
Figure 54-47 iPod shuffle 4th gen. Dimensional Drawing 684
Figure 54-48 iPod shuffle 3rd gen. Dimensional Drawing 685
Figure 54-49 iPod shuffle 2nd gen. Dimensional Drawing 686
Figure 54-50 iPod shuffle Dimensional Drawing (1 of 2) 687
Figure 54-51 iPod shuffle Dimensional Drawing (2 of 2) 688
Figure 54-52 iPod mini Dimensional Drawing 689

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

1. Introduction

NOTICE OF PROPRIETARY PROPERTY: THE INFORMATION CONTAINED HEREIN IS THE PROPRIETARY PROPERTY OF APPLE INC. THE POSSESSOR AGREES TO THE FOLLOWING: (I) TO MAINTAIN THIS DOCUMENT IN CONFIDENCE, (II) NOT TO REPRODUCE OR COPY IT, (III) NOT TO REVEAL OR PUBLISH IT IN WHOLE OR IN PART, (IV) ALL RIGHTS RESERVED.

ACCESS TO THIS DOCUMENT AND THE INFORMATION CONTAINED THEREIN IS GOVERNED BY THE TERMS OF THE MFI LICENSE AGREEMENT AND/OR THE IPOD-IPHONE AIS EVALUATION AGREEMENT. ALL OTHER USE SHALL BE AT APPLE'S SOLE DISCRETION.

1.1 Purpose of This Specification

This specification details requirements and recommendations for accessories that interface with Apple devices that have the Apple Lightning™ connector.

1.2 Requirements, Recommendations, and Permissions

This specification contains statements that are incorporated by reference into legal agreements between Apple and its licensees. The use of the words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *not recommended*, *may*, *optional*, and *deprecated* in a statement have the following meanings:

- *must*, *shall*, or *required* means the statement is an absolute requirement.
- *must not*, *shall not* or *prohibited* means the statement is an absolute prohibition.
- *should* or *recommended* means the full implications must be understood before choosing a different course.
- *should not* or *not recommended* means the full implications must be understood before choosing this course.
- *may* or *optional* means the statement is truly optional, and its presence or absence cannot be assumed.
- *deprecated* means the statement is provided for historical purposes only and is equivalent to 'must not'.

The absence of requirements, recommendations, or permissions for a specific accessory design in this specification must not be interpreted as implied approval of that design. Developers are strongly encouraged to ask Apple for feedback on accessory designs that are not explicitly mentioned in this specification.

1.3 Applicability

This specification covers the accessory interface exposed by Apple devices that have the Lightning connector. At time of release, this includes the following Apple devices:

- iPod nano (7th generation)
- iPod touch (5th generation)
- iPhone 5
- iPhone 5c
- iPhone 5s
- iPad (4th generation)
- iPad Air
- iPad mini
- iPad mini 2

This specification also covers portions of the accessory interface exposed by Apple devices that have the 30-pin connector. More content can be found in the following specifications:

- *MFi Accessory Firmware Specification R46*
- *MFi Accessory Hardware Specification R9*

Any conflicts between the *MFi Accessory Interface Specification* and *MFi Accessory Firmware/Hardware Specifications* must be resolved in favor of the *MFi Accessory Interface Specification*.

1.4 Terminology

1.4.1 Accessory, Device, and Product

Throughout this specification:

- The term *device* is used to refer to an Apple iPod, iPhone, or iPad (typically running iOS, Apple's mobile operating system).

Pary Wu
Unigrand Ltd
paryw@unigrand.com.tw

- The term *accessory* is used to refer to any product intended to interface with a device via the means described in this specification.
- The term *product* is used to refer generically to either a Mac (Apple computers that run OS X) or an aforementioned *device*.

1.4.2 Authentication Coprocessor

An accessory hardware component that provides Apple device-related digital signature creation and verification services.

1.4.3 I2C Bus

A 2-wire serial bus designed by Philips to allow easy communication between components that reside on the same circuit board. The I2C specification is located at http://www.nxp.com/documents/user_manual/UM10204.pdf.

1.4.4 Challenge

A random number sent from an Apple device to an accessory, or vice versa. The device/accessory being challenged must perform a challenge response computation on the offered challenge and return the resulting Challenge Response (page 42) to the challenging device for verification.

1.4.5 Challenge Response

The result obtained by performing a challenge response process on an offered Challenge (page 42).

1.4.6 X.509 Certificate

A standard defined by the International Telecommunications Union (ITU) that governs the format of certificates used for authentication and sender identity verification in public-key cryptography. X.509 certificates contain the public keys used in the Apple device's accessory authentication process.

1.4.7 Component

An accessory is defined as a collection of functional units called *components*. Examples of a component include, but are not limited to, the following:

- Data transport
- Power source
- Human Interface Device (HID) control set

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

1.4.8 Feature

All accessories must support one or more accessory interface *features*. Each feature may have associated accessory design requirements and recommendations; an accessory must comply with all feature-specific requirements to properly support the feature. Some features require other features to be implemented.

Accessory design must take into account the possibility that an Apple device may not support all of the accessory's implemented features and react appropriately.

1.4.9 USB Device and Host Mode

Apple devices are capable of taking on both USB Host and USB Device roles when connecting to accessories. In this specification, *USB Host Mode* is always used when the Apple device is the USB host and the accessory is a USB device. *USB Device Mode* is always used when the Apple device is the USB device and the accessory is a USB host.

1.4.10 iAP

There are two iAP protocols - *iAP1* and *iAP2*. iAP1 is the *iPod Accessory Protocol* as specified in *MFi Accessory Firmware Specification R46*. iAP2 is a complete replacement for iAP1 and is not backward compatible; it is specified in this specification.

An iAP2 *connection* is composed of an iAP2 *transport*, iAP2 *link*, and one or more iAP2 *sessions*.

1.4.11 Direct User Action

Some accessory interface features or iAP2 messages have a direct user action requirement. When this requirement is specified, the accessory must not autonomously use the feature or send the message without direct action taken by the user, such as pressing a physical button on the accessory or attaching the accessory to the Apple device's Lightning receptacle. This extends to resends or retries of iAP2 control session messages; sending such a message more than once in response to one direct user action is explicitly prohibited. Retries of iAP2 link packets are not affected.

Failure to observe this requirement when specified will always result in failure to pass self certification.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2. General Requirements and Recommendations

The requirements in this section apply to all accessories regardless of their feature sets.

2.1 Development Tools and Emulators

Use of this specification to create accessory development tools, such as protocol analyzers, intended for use by other accessory developers or manufacturers is prohibited unless explicitly authorized by Apple.

Similarly, this specification must not be used to create accessories that emulate the accessory interface exposed by Apple devices.

2.2 Reference Designs & Development Kits

Reference designs and development kits may be developed using this specification to assist with the design and prototyping of accessories. These may consist of any of the following:

- mechanical designs or components
- electrical designs or components
- firmware or application software (libraries, frameworks, sample code, etc.)
- development tools

Reference designs and development kits must treat specification recommendations as requirements and pass certification audits with no warnings.

A datasheet for the reference design or development kit listing all supported accessory interface features and expected accessory design/prototyping approaches must be submitted for certification audit.

2.3 Accessory Authentication and Accessory Identification

All accessories other than simple chargers or charge/sync cables must authenticate and identify their feature sets to the Apple device.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2.4 iAP1

iAP1 is not recommended for new accessories. Its development is being placed into a maintenance mode.

Accessories that connect to Apple devices with the Lightning connector and support only iAP1 must send the iAP1 StartIDPS command once every 5 seconds until the Apple device responds.

2.5 iAP2

> **Note:** Starting in 2015, all new accessories must support iAP2.

iAP2 is recommended for new accessories.

Accessories may support both iAP1 and iAP2. This specification details how an accessory may detect whether the connected Apple device supports iAP2 in Initialization (page 546). If the Apple device does not support iAP2, the accessory may fall back to iAP1.

If an accessory implements support for both iAP1 and iAP2, it must always use only iAP2 when connected to an Apple device that supports iAP2.

The iPod nano (7th generation) does not support iAP2; accessories wishing to claim compatibility with it must implement support for iAP1.

2.6 Detachable Connector Assemblies

Docks, dongles, and form fitting accessories as defined in Apple Lightning Connector (page 114) must not implement user detachable 30-pin/Lightning connector assemblies.

2.7 Adapters and Proxies

Accessories must not directly or indirectly facilitate the connection of other products that are not compliant with this specification to Apple devices.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2.8 Mixed 30-pin and Lightning Connectors

Simple charger accessories may incorporate both a Power/Sync Only 30-pin connector and a B configuration (USB Device Mode) Lightning connector. See Multiple Connectors (page 486) for additional requirements.

Accessories that both provide power and exchange data with an Apple device may incorporate both a 30-pin connector and a Lightning connector under the following conditions:

- The accessories must use the same data transport (serial/USB Host/USB Device) to communicate with both connectors unless they implement the Digital Audio (page 366) feature on both connectors. In that situation, the accessory may use the USB Host Mode transport on the Lightning connector and USB Device Mode transport on the 30-pin connector.
- If the USB data transport is used, both connectors must meet the signal integrity requirements set forth in Connector Signal Integrity Requirements (page 154). Possible implementation approaches include a microcontroller with sufficient I/O capability to maintain two separate data transport paths or a multiplexer chip such as the Intersil ISL54233.
- All connectors that are physically accessible must provide power even when no Apple device is attached.
- If the accessory is intended to work with only one Apple device at any given time, it must switch to the last attached Apple device by looking for Accessory Power to go high on a connector.
- The accessory must use iAP1 or iAP2 to inform the device of available power.

The following diagrams illustrate these requirements.

Figure 2-1 Mixed 30-pin and Lightning Connector USB Accessory

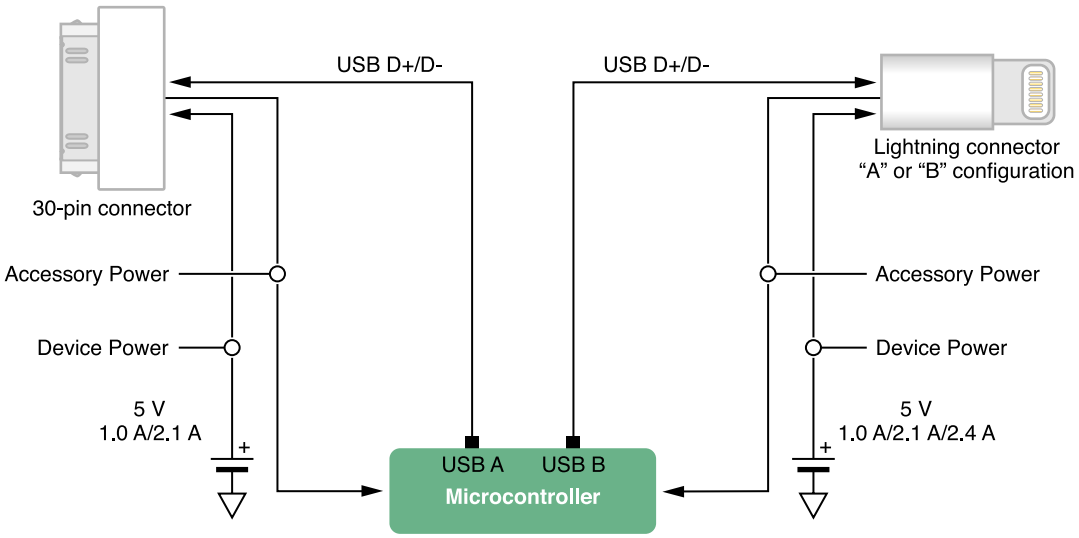


Figure 2-2 Mixed 30-pin and Lightning Connector USB Accessory with Multiplexer

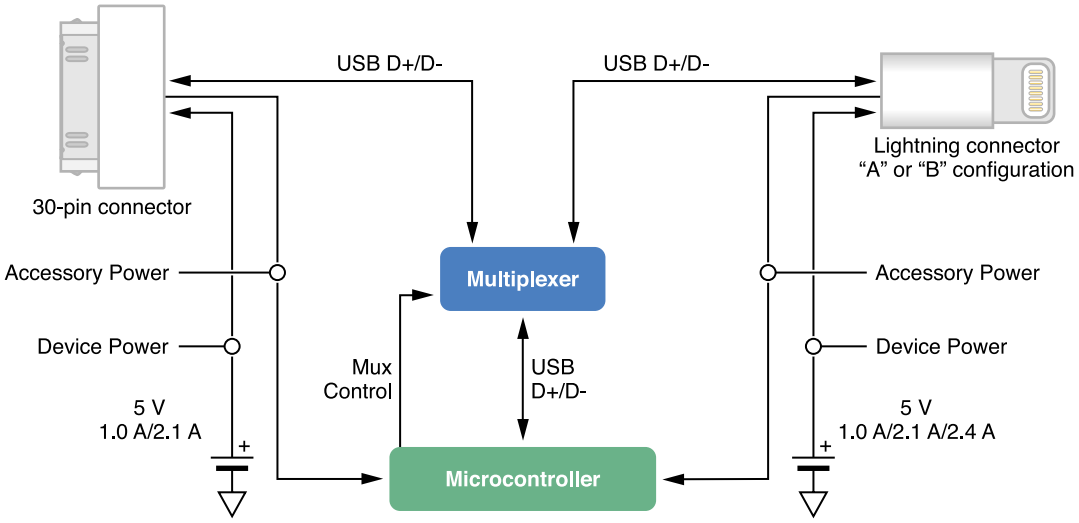


Figure 2-3 Mixed 30-pin and Lightning Connector Serial Accessory

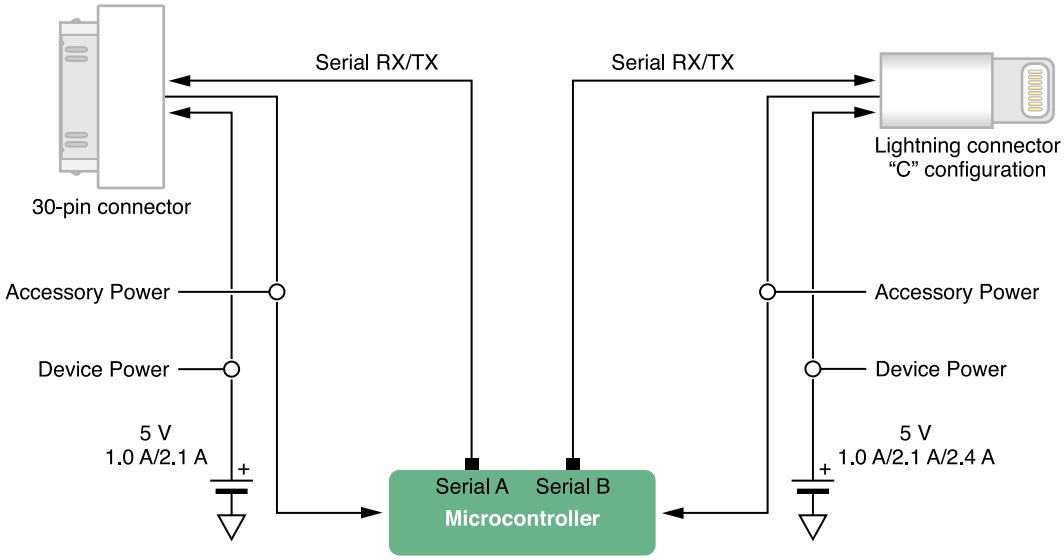
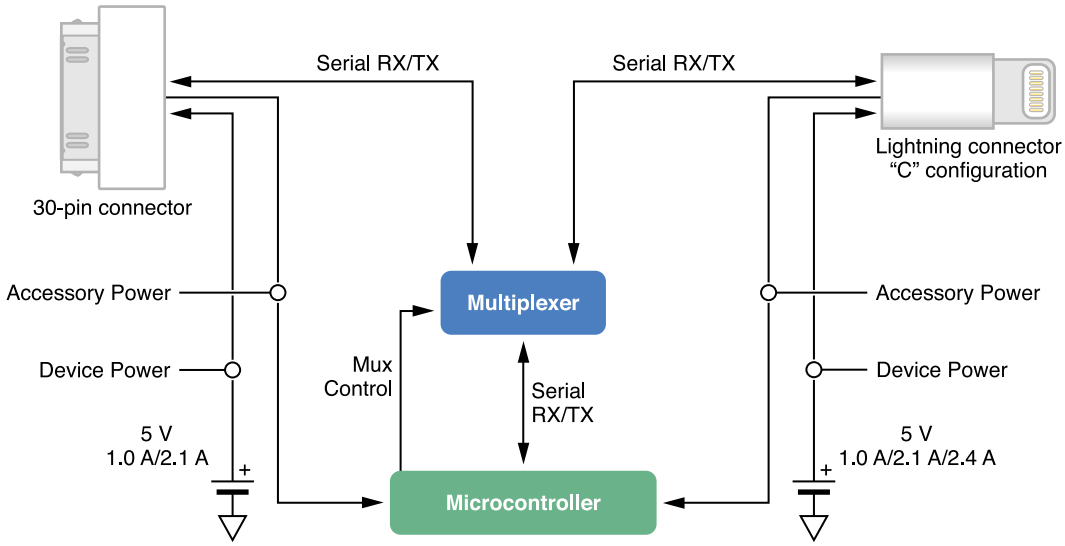


Figure 2-4 Mixed 30-pin and Lightning Connector Serial Accessory with Multiplexer



All other accessories must not integrate both 30-pin and Lightning connectors.

2.9 Mixed Headphone Jack and Lightning Connectors

Accessories must not attach to both the headphone jack and the Lightning connector on an Apple device.

2.10 Apple USB Power Adapters

Accessories must not use Apple USB power adapters to support the weight of an Apple device while plugged into a wall outlet.

2.11 Apple Device Detection

Accessories that implement iAP must not assume that an Apple device is present until the Apple device responds to the accessory's StartIDPS command (iAP1) or Link Initialization Byte Sequence (iAP2).

Accessories that both implement iAP and integrate a Lightning connector that offers Accessory Power (see Connector Pad/Pin Configuration (page 125)) may choose to wait until Accessory Power is detected before sending the StartIDPS command or Link Initialization Byte Sequence.

Accessories that do not implement iAP must assume that an Apple device is present at all times.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2.12 Multiple Simultaneous iAP2 Connections

Accessories must not implement iAP2 connections on multiple transports simultaneously. An accessory may shut down the iAP2 connection on one transport and then start up an iAP2 connection on a different transport, but it must change its identification information accordingly.

If both wired (e.g., Lightning) and wireless (e.g., Bluetooth) transports are simultaneously available to the same Apple device, the accessory must use the wired transport.

2.13 Presentation of Apple Device Updates

Accessories must not present the state of an Apple device to a user before being informed of that state by the Apple device. For example, if the user presses a 'Shuffle' button on an accessory and causes a corresponding media remote control HID usage to be sent to the Apple device, the accessory must not inform the user of a change in shuffle state until the Apple device has confirmed the state change. Assuming that the state change will happen will cause problems if the state change does not occur, such as when a third party app is running and does not respond to the user input.

Similarly, accessories must not cache Apple device information unless explicitly specified otherwise.

> **Note:** Any accessory control, whether physical or virtual, that is an analogue to a corresponding HID usage specified in HID Requirements (page 407) must send that exact HID usage to the Apple device when the user takes direct action to actuate that control. For example, use of a 'Play/Pause' button to programmatically send either the 'Play' or 'Pause' HID usage based on cached Apple device state is prohibited; a 'Play/Pause' HID usage must be sent.

2.14 Relationships Between Multiple Accessories

All accessories must not require that another accessory be connected to the same Apple device in order to function. Additionally, accessories must not take action on behalf of another accessory. In all such situations, one accessory with multiple components must be designed and implemented, and the accessory is solely responsible for managing its overall connection state with the Apple device when individual components are active or inactive.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2.15 iBeacon

MFi accessories must not support the iBeacon feature.

2.16 Feature Duplication

Accessories must not implement functionality that overlaps with a specified accessory interface feature without also offering that same functionality via the specified interface.

For example, every accessory that provides or receives any kind of MIDI data to or from an Apple device must support the MIDI feature in addition to any other mechanism it may implement, such as an External Accessory Protocol.

2.17 Temperature Range

The temperature range of the accessory must be greater than or equal to the published temperature ranges of all the Apple devices with which it claims compatibility.

2.18 Magnetic Fields

All accessories that claim to be compatible with Apple devices that contain digital compasses must minimize interference with the digital compass and must not repeatedly trigger compass recalibration.

2.19 Cables with USB Connectors

All cables included with and/or permanently tethered to or attached to an accessory that terminate in at least one USB connector must meet or exceed USB-IF specifications and ECNs.

Accessories must use the conductors in standard USB connectors as defined in the USB-IF specifications and Engineering Change Notices.

For cables incorporating a Lightning connector, see Apple Lightning Connector (page 114).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2.20 Cables with Non-USB Connectors

Accessory cables that incorporate a non-USB connector must integrate the non-USB connector in accordance with any specifications or requirements that apply to the non-USB connector. For instance:

- Cables that terminate in a 2.5 mm or 3.5 mm audio plug must only carry analog audio signals.
- Cables that terminate in a HDMI, DisplayPort, FireWire, or Thunderbolt connector must use the connector in a manner that is compliant with their respective specifications.

For cables incorporating a Lightning connector, see Apple Lightning Connector (page 114).

2.21 Integrated USB Receptacles

Accessories that incorporate a USB receptacle for the purpose of drawing power from an external USB power supply must comply with the following requirements:

- They must support the USB Dedicated Charging Port (DCP) as defined in the USB 2.0 specification.
- They must not attempt to detect Apple branded or MFi power supplies that connect the D+/D- pins to resistor networks as specified in Connectors That Do Not Implement iAP2 (page 485).
- If the accessory can also be enumerated as a USB device via the USB receptacle, it must comply with the USB 2.0 specification and not draw more than 100 mA of current until it has been successfully enumerated by the external USB host.

2.22 Integrated Non-USB Receptacles

Accessories that integrate non-USB receptacles must use them in accordance with any specifications or requirements that apply to the non-USB receptacle. For instance:

- 2.5 mm and 3.5 mm audio plug receptacles must not carry power or digital data.
- HDMI, DisplayPort, FireWire, or Thunderbolt receptacles must be implemented in a manner that is compliant with their respective specifications.

Accessory non-USB receptacles that are intended for an included accessory cable that incorporates both a Lightning connector and non-USB connector may be labeled using the compatibility icon for an iPhone 5 (without any accompanying text) to highlight to the user that the cable should be plugged into that receptacle.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2.23 User Supplied Cables and Power Supplies

Accessories that can reasonably be used with user-supplied cables and/or power supplies must be designed to work with any cable or power supply that is compliant with this specification, including Apple branded cables and power supplies. Such accessories must not declare compatibility only with Apple branded USB cables or USB power supplies.

This compatibility requirement applies to all aspects of user-supplied cables and power supplies. For example:

- Connector receptacles on accessories must accomodate all spec-compliant connector overmolds, and any accessory opening surrounding the Lightning connector on an Apple device must provide sufficient clearance for spec-compliant connector overmolds.
- Accessories must work with all cables that comply with the specification with regards to electrical DCR and SI.

> **Note:** Such accessories must be tested with a wide variety of spec-compliant cables (including various lengths of the same cable if applicable) and power supplies during accessory development, in addition to Apple branded cables and power supplies.

2.24 Removable Storage

Accessories must not integrate removable storage components, such as SD card readers, that transfer data to or from an Apple device via means other than an External Accessory Protocol (see External Accessory Protocol (page 372)).

2.25 RF Transmission and Reception

Accessories must avoid degrading the wireless performance of Apple devices. Device antenna and sensor locations must be taken into consideration during the design process. Specifically, the following wireless performance characteristics must be considered:

- **Reduction of the device's RF/antenna efficiency:** Accessories must minimize decreases in the device's total radiated power (TRP). This can be quantified by measuring TRP across all of the device's operating bands.
- **Desense of the device's RF reception:** Accessories must minimize decreases in the device's effective isotropic sensitivity (EIS). This can be quantified by measuring EIS across all of the device's operating bands.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

See RF (OTA) (page 349).

2.26 TDMA Noise

GSM phones emit radiated and conducted RF noise, which can produce time division multiple access (TDMA) sounds from audio outputs. Accessories must minimize coupling of audible interference from the Apple device (commonly known as *TDMA noise* or *chopper noise*) into an accessory's electronics.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

3. Apple Authentication Coprocessor 2.0C

3.1 Coprocessor 2.0C Overview

An Apple device verifies whether a third-party accessory attached to it is authorized for use with the Apple device by issuing an authentication challenge to the accessory. The accessory must respond to the Apple device's challenge, and it can do so only with the assistance of an Apple Authentication Coprocessor (CP) chip located in the accessory. Conversely, the accessory can use its CP chip to authenticate the Apple device.

Earlier versions of the Apple Authentication Coprocessor (1.0, 2.0A, and 2.0B) were implemented in QFN-40, QFN-20, and SOP-8 packages. The current version, 2.0C, is supplied in a smaller and more efficient PG-USON-8-1 package.

The Export Classification Control Number (ECCN) of the Apple Authentication Coprocessor 2.0C is 5A992 NLR.

3.2 Coprocessor 2.0C Authentication Protocol

The authentication protocol supported by the Apple Authentication Coprocessor 2.0C is based on standard X.509 version 3 certification. Each certificate is generated and signed by a recognized certificate authority and has a unique serial number. Information about the X.509 standard can be found at the IETF website http://tools.ietf.org/html/3280.

For information about accessory authentication of the Apple device, refer to Accessory Authentication (page 201).

For information about Apple device authentication of the accessory, refer to Device Authentication (page 362).

3.3 Coprocessor 2.0C Signals and Pinouts

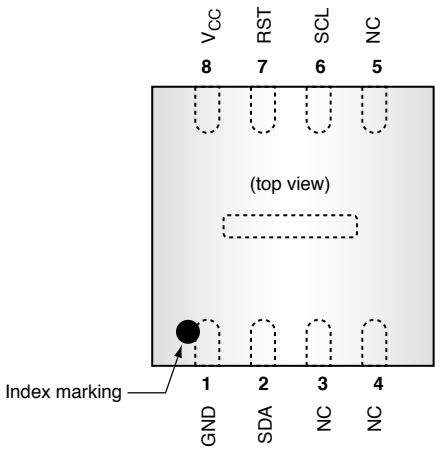
The 2.0C CP chip signal descriptions are given in Table 3-1 (page 55) and its pinouts are shown in Figure 3-1 (page 55).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 3-1 Coprocessor signals

Signal name	Pin	I/O	Description
GND	1		Supply voltage, negative terminal
SDA	2	I/O	I2C data
NC	3-5		Must not be connected
SCL	6	I	I2C clock
RST	7	I	At reset: selects I2C slave address. During operation: CP warm reset
V_{CC}	8		Supply voltage, positive terminal

Figure 3-1 Coprocessor pinout, top view



PG-USON-8-1 package

The thermal pad on the bottom of the CP may be left unconnected or optionally connected to GND.

3.4 Coprocessor 2.0C Address Selection

After power-up or in response to a warm reset, the state of RST is used to select the CP's I2C slave addresses, as shown in Table 3-2 (page 55).

Table 3-2 Coprocessor address selection signals

RST state	I2C write address	I2C read address
0	0x20	0x21

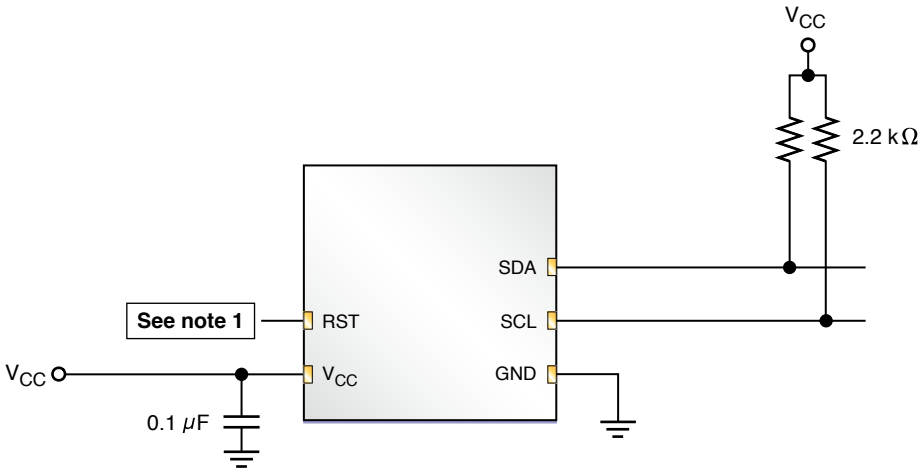
RST state	I2C write address	I2C read address
1	0x22	0x23

See I2C Communications Process (page 59) for the interface requirements of the CP's I2C slave communication transport.

3.5 Coprocessor 2.0C Reference Circuit

The reference circuit for I2C operation of the CP is shown in Figure 3-2 (page 56).

Figure 3-2 Coprocessor reference circuit diagram



Note: If the CP's warm reset function is not needed, RST can be tied to either V_{CC} or GND, depending on which of the addressing modes shown in Table 3-2 (page 55) is used. If the warm reset function is needed, RST should be connected to a general-purpose I/O line on the accessory's controller. For further details, see I2C Startup On Warm Reset (page 58).

3.6 Coprocessor 2.0C System Voltage

The 2.0C CP may be used either in an accessory powered by an attached Apple device or in an accessory that has its own power source.

3.7 Coprocessor 2.0C I2C Interface

The CP's I2C communication interface can be started up either by supplying power to the V_{CC} line or by performing a warm reset on the RST line. The accessory actions required for these two procedures are specified in the next sections.

3.7.1 I2C Startup On Power On

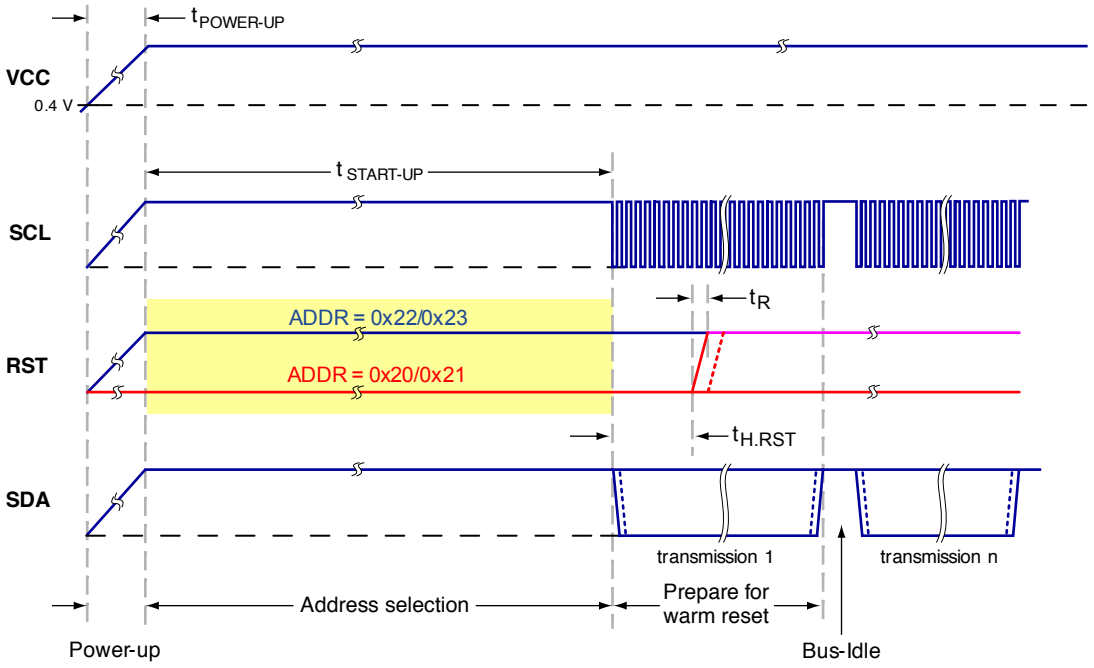
To activate the I2C interface by supplying power to the V_{CC} line, the accessory must perform the following startup procedure. This procedure is required both to support address selection by means of the RST line and to support the option of a warm reset later.

- The V_{CC} line must be supplied with power and the SDA and SCL lines must be set high during the entire procedure.
- The RST line, used to select addressing as described in Coprocessor 2.0C Address Selection (page 55), must be kept either low or high during the entire procedure. If the RST line is kept low during the procedure, the accessory must set it high not earlier than $10ms$ after the V_{CC} line goes high but before the first data transmission occurs.
- The first data transmission may start not earlier than $10ms$ after the V_{CC} line goes high. If the RST line has been kept low and the accessory might need to perform a warm reset of the CP chip later, the accessory must set the RST line high not earlier that $1ms$ after the start of the first data transmission ($t_{H.RST}$ interval in Figure 3-3 (page 58)). If the option of a warm reset later is not needed, the accessory may tie RST directly to either V_{CC} or GND.

Figure 3-3 (page 58) diagrams the timing of the I2C interface when it is started up by turning power on.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 3-3 Coprocessor I2C power on timing



In this diagram, $t_{STARTUP} \geq 10ms$ and $t_{H.RST} \geq 1ms$. The rise time of t_R and the fall time of t_F are both $< 1\mu s$ from 10% to 90% of the signal amplitude, and $t_{POWER\text{-}UP} < 200\mu s$ from $V_{CC} = 0.4$ V until $V_{CC} = 90\%$ of the target supply voltage.

3.7.2 I2C Startup On Warm Reset

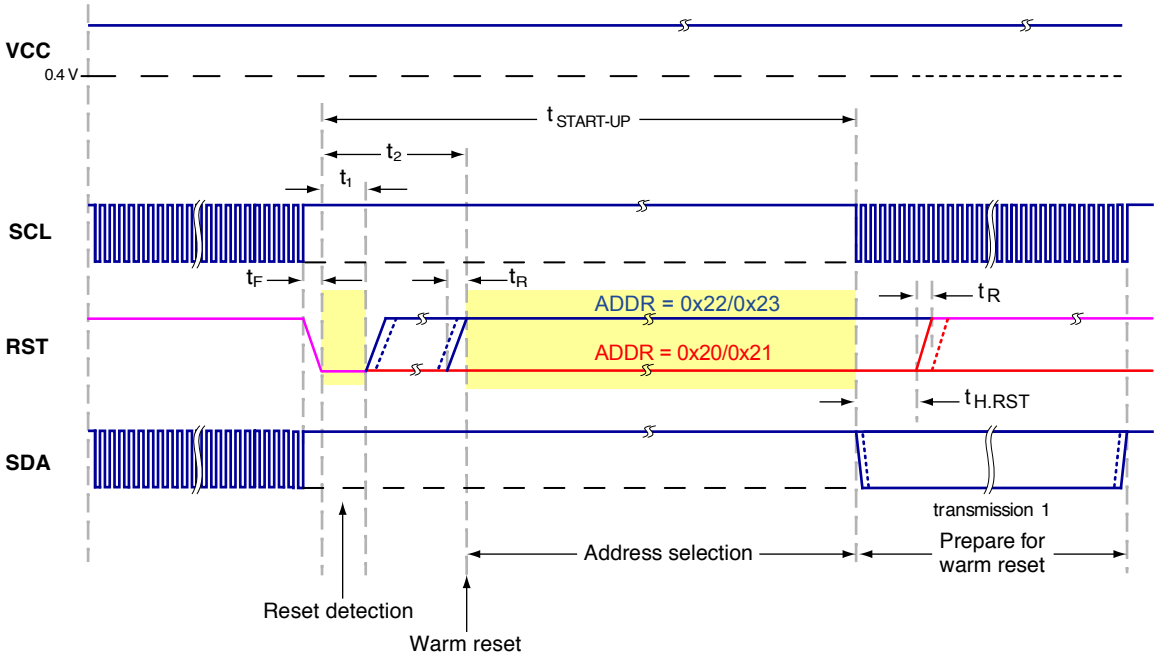
To reset the I2C interface through the RST line, after activating it through power-up as specified in I2C Startup On Power On (page 57), the accessory must perform the following procedure:

- The V_{CC} line must be supplied with power during the entire procedure.
- The terminal must halt I2C communication, and the SDA and SCL lines must be set high before the RST line is set low.
- The RST line must be set low and kept low for at least 10 μs. Not later than $1ms$ after the falling edge of the RST signal, the accessory must finish its I2C address selection by either keeping RST low or driving RST high. If the RST line is kept low during the warm reset, the accessory must set it high not earlier than $10ms$ after the V_{CC} line goes high but before the first data transmission occurs.
- The first data transmission may start not earlier than $10ms$ after the falling edge of the RST signal. If the RST line has been kept low during the warm reset, the accessory must set it high not earlier that $1ms$ after the start of the first data transmission ($t_{H.RST}$ interval in Figure 3-4 (page 59)).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 3-4 (page 59) diagrams the timing of the I2C interface when it is reset by software toggling the RST line without a power off/on cycle.

Figure 3-4 Coprocessor I2C warm reset timing



In this diagram, $t_{STARTUP} \geq 10$ ms, $t_{H.RST} \geq 1$ ms, $t_1 > 10\ \mu s$, and $t_2 = 3ms$. The rise time of t_R and the fall time of t_F are both $< 1\ \mu s$ from 10% to 90% of the signal amplitude.

3.7.3 I2C Communications Process

When the CP is addressed using I2C, it acts as a standard 7-bit I2C slave. The I2C slave address is configured upon reset and is based on the RST input. The I2C effective slave address for writing is shown in Figure 3-5 (page 59) and the corresponding read address in Figure 3-6 (page 59).

Figure 3-5 Coprocessor I2C slave write address

A6	A5	A4	A3	A2	A1	A0	R/nW
0	0	1	0	0	0	RST	0

Figure 3-6 Coprocessor I2C slave read address

A6	A5	A4	A3	A2	A1	A0	R/nW
0	0	1	0	0	0	RST	1

Copyright © 2014 Apple Inc. All Rights Reserved.

In I2C mode, the CP has both a write address and a read address, as is typical for an I2C device. The I2C write address of the CP consists of the seven bits [A6:A0] followed by 0 for the R/nW bit. The I2C read address of the CP consists of the seven bits [A6:A0] followed by 1 for the R/nW bit. If the RST input is connected to ground, the write and read addresses of the CP are 0x20 and 0x21 respectively; if it is pulled high, the write and read addresses of the CP are 0x22 and 0x23.

3.7.4 I2C Sleep Mode

The 2.0C CP conserves power by entering a Sleep mode. It enters this mode automatically and cannot be forced to it externally. When in Sleep mode, the CP automatically wakes in response to any I2C communications sent to its address.

3.8 Coprocessor 2.0C Registers

Registers within the Apple Authentication Coprocessor 2.0C (CP) are accessed via I2C transport, as described in I2C Communications Process (page 59). This section specifies the CP's register addressing details and telegram formats.

3.8.1 Register Addresses

Registers and their addresses in the CP are listed in Table 3-3 (page 60). Each register is discussed in the sections that follow.

> **Note:** Registers in the same block with consecutive addresses may be read from sequentially in increasing numerical order, except as noted in the last column of Table 3-3 (page 60). Registers must not be written to sequentially except as noted. Multibyte numeric values are stored in big-endian order; for example, the first byte in a two-byte register is the MSB of the stored value and the second byte is its LSB.

Table 3-3 Coprocessor register map

Address	Block	Name	Length (bytes)	Reset Value	Access
0x00	0	Device Version	1	0x05	Read-only
0x01	0	Firmware Version	1	0x01	Read-only

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Address	Block	Name	Length (bytes)	Reset Value	Access
0x02	0	Authentication Protocol Major Version	1	0x02	Read-only
0x03	0	Authentication Protocol Minor Version	1	0x00	Read-only
0x04	0	Device ID	4	0x00000200	Read-only
0x05	0	Error Code	1	0x00	Read-only
0x10	1	Authentication Control and Status	1	0x00	Read/write
0x11	1	Challenge Response Data Length	2	128	Read/write
0x12	1	Challenge Response Data	128	Undefined	Read/write
0x20	2	Challenge Data Length	2	20	Read/write
0x21	2	Challenge Data	128	Undefined	Read/write
0x30	3	Accessory Certificate Data Length	2	≤ 1280	Read-only
0x31	3	Accessory Certificate Data (Part 1)	128	Certificate	Read-only
0x32	3	Accessory Certificate Data (Part 2)	128	Certificate	Read-only
0x33	3	Accessory Certificate Data (Part 3)	128	Certificate	Read-only
0x34	3	Accessory Certificate Data (Part 4)	128	Certificate	Read-only
0x35	3	Accessory Certificate Data (Part 5)	128	Certificate	Read-only
0x36	3	Accessory Certificate Data (Part 6)	128	Certificate	Read-only
0x37	3	Accessory Certificate Data (Part 7)	128	Certificate	Read-only

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Address	Block	Name	Length (bytes)	Reset Value	Access
0x38	3	Accessory Certificate Data (Part 8)	128	Certificate	Read-only
0x39	3	Accessory Certificate Data (Part 9)	128	Certificate	Read-only
0x3A	3	Accessory Certificate Data (Part 10)	128	Certificate	Read-only
0x40	4	Self-Test Control and Status	1	0x00	Read/write
0x41-0x4C	4	Reserved			
0x4D	1	System Event Counter (SEC)	1	Undefined	Read-only
0x50	5	Apple Device Certificate Data Length	2	0x0000	Read-only
0x51	5	Apple Device Certificate Data (Part 1)	128	Undefined	Read/write
0x52	5	Apple Device Certificate Data (Part 2)	128	Undefined	Read/write
0x53	5	Apple Device Certificate Data (Part 3)	128	Undefined	Read/write
0x54	5	Apple Device Certificate Data (Part 4)	128	Undefined	Read/write
0x55	5	Apple Device Certificate Data (Part 5)	128	Undefined	Read/write
0x56	5	Apple Device Certificate Data (Part 6)	128	Undefined	Read/write
0x57	5	Apple Device Certificate Data (Part 7)	128	Undefined	Read/write
0x58	5	Apple Device Certificate Data (Part 8)	128	Undefined	Read/write

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Note: Normally, reading register 0x05 will clear it. However, register 0x05 can be read sequentially only as part of a sequence that begins with a register in the range 0x00-0x04, in which case the read operation does not clear it.

Note: Registers 0x11 and 0x20 may each be written sequentially with registers 0x12 and 0x21, respectively.

3.8.2 Register Descriptions

This section describes the ways that the CP registers listed in Table 3-3 (page 60) are used.

3.8.2.1 Device Version

The Device Version read-only register contains the version number of the coprocessor device. The current Authentication 2.0C coprocessor is designated as device version 0x05.

3.8.2.2 Firmware Version

The Firmware Version read-only register contains the version number of the coprocessor firmware. Firmware version numbers advance by whole integers.

3.8.2.3 Authentication Protocol Major and Minor Versions

The Authentication Protocol Major Version and Authentication Protocol Minor Version read-only registers provide the version number of the authentication protocol that the CP supports. This information is accessed by the iAP command `RetDevAuthenticationInfo` during accessory authentication.

3.8.2.4 Device ID

The Device ID read-only register is not used by accessories that implement iAP2.

3.8.2.5 Error Code

The Error Code read-only register stores the most recent communication or authentication process error code generated since the register was last cleared. The error code register is cleared after it is read. The possible error codes are listed in Table 3-4 (page 64).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

If a single communication operation happens to produce multiple errors (for example, by writing an invalid challenge response length during a multiregister write that also attempts to continue past the end of the corresponding block) then only the highest-numbered error code is stored.

Table 3-4 Coprocessor error codes

Error Code	Description
0x00	No error
0x01	Invalid register for read
0x02	Invalid register for write
0x03	Invalid challenge response length
0x04	Invalid challenge length
0x05	Invalid certificate length
0x06	Internal process error during challenge response generation
0x07	Internal process error during challenge generation
0x08	Internal process error during challenge response verification
0x09	Internal process error during certificate validation
0x0A	Invalid process control
0x0B	Process control out of sequence
0x0C-0xFF	Reserved

3.8.2.6 Authentication Control and Status

The Authentication Control and Status read/write register provides control and status information for the CP's authentication processes.

When read from, the Authentication Control and Status register provides the status of the most recently requested CP process, as shown in Figure 3-7 (page 64), Table 3-5 (page 65) and Table 3-6 (page 65).

Figure 3-7 Coprocessor Authentication Control and Status register, read-only bits

Table 3-5 Coprocessor Authentication ERR_SET values

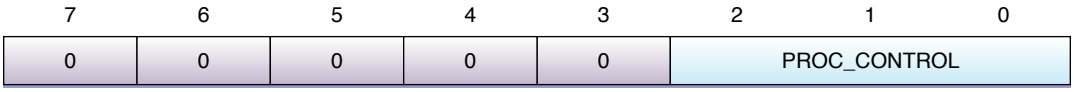
ERR_SET Value	Description
0	The Error Code register does not contain a code generated by the most recent command execution. However, it may still contain the most recent error code (greater than 0x00) generated by an earlier command.
1	The Error Code register contains the most recent process or communication error. Both this bit and the Error Code register contents are cleared after the Error Code register is next read. This bit is also cleared after every successful command execution.

Table 3-6 Coprocessor Authentication PROC_RESULTS values

PROC_RESULTS Value	Description
0	Most recent process did not produce valid results.
1	Accessory challenge response successfully generated.
2	Challenge successfully generated.
3	Apple device challenge response successfully verified.
4	Apple device certificate successfully validated.
5-7	Reserved.

When written to, the Authentication Control and Status register controls the start of CP processes, as shown in Figure 3-8 (page 65) and Table 3-7 (page 66).

Figure 3-8 Coprocessor Authentication Control and Status register, write-only bits



Note: Attempts to write to other bits in the Coprocessor Authentication Control and Status register are ignored.

Table 3-7 Coprocessor Authentication PROC_CONTROL values

PROC_CONTROL Value	Description	Notes
0	No operation	This control does nothing and always reports success (ERR_SET = 0; PROC_RESULTS = 0).
1	Start new challenge response-generation process	
2	Start new challenge-generation process	The length of the challenge to be generated is defined by the Challenge Data Length register and ranges from 1 to 128 bytes.
3	Start new challenge response-verification process	
4	Start new certificate-validation process	Do not attempt to read the accessory certificate after writing the Apple device certificate but before validating it by this control.
5	No operation	This control does nothing and always reports success (ERR_SET = 0; PROC_RESULTS = 0).
6-7	Reserved.	

3.8.2.7 Challenge Response Data Length

The Challenge Response Data Length read/write register holds the length in bytes of the results of the most recent challenge response-generation process (if the Apple device is authenticating an accessory) or challenge response-verification process (if the accessory is authenticating the Apple device).

Before a challenge response-generation process begins, this register should contain 0x80, the maximum allowable challenge response length. After completion of the challenge response-generation process, the CP updates this register to contain the actual length of the generated challenge response. This updated value should be read in order to determine how much of the Challenge Response Data register contains valid challenge response bytes.

Before a challenge response-verification process begins, this register should hold the actual length of the challenge response being verified.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

3.8.2.8 Challenge Response Data

In the case of a challenge response generation process, the Challenge Response Data register holds the newly-generated data. In the case of a challenge response verification process, it holds the challenge response to be verified.

3.8.2.9 Challenge Data Length

The Challenge Data Length read/write register holds the length, in bytes, of the current challenge. This challenge may either be written into the CP, during Apple device authentication of an accessory, or generated by the CP during accessory authentication of an Apple device.

Before starting a challenge response-generation process on the current challenge during Apple device authentication of an accessory, this register must contain the length of the challenge.

Before starting a new challenge-generation process during accessory authentication of an Apple device, this register should contain the requested challenge length. The length must be in the range of 1 to 128 bytes, thus writing any other value will cause an error.

The required length of an authentication challenge is 20 bytes.

3.8.2.10 Challenge Data

The Challenge Data register holds the current challenge data. This data is either written into the CP or generated by the CP depending on the specific operation. The number of bytes used or generated is determined by the value of the Challenge Length Data register.

3.8.2.11 Accessory Certificate Data Length

The Accessory Certificate Data Length read-only register holds the length of the X.509 certificate that the Apple device uses to authenticate an accessory. The length of a certificate varies but is always less than or equal to 1280 bytes. This length limit may not hold for future versions of the authentication protocol.

3.8.2.12 Accessory Certificate Data

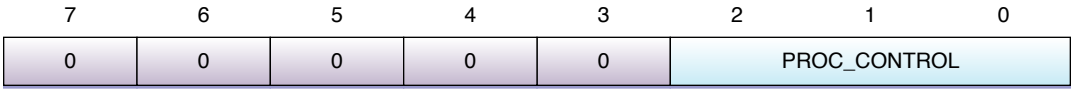
The Accessory Certificate Data read-only register holds the PKCS#7-wrapped X.509 certificate that the Apple device uses to authenticate an accessory. The Accessory Certificate may be read from the coprocessor in 128-byte pages starting at any Accessory Certificate Data Page address, or it may be read in a continuous stream starting at Page 1. Since the length of the Accessory Certificate varies, fewer than all of the pages may be used. The Accessory Certificate Data Length value can be read to determine which Accessory Certificate Data Pages contain the certificate data.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

3.8.2.13 Self-Test Control and Status

The Self-Test Control and Status read/write register provides access to the built-in self-test functions of the coprocessor. When it is set to a value of 1, the Self-Test Control and Status register initiates a self-test process, as shown in Figure 3-9 (page 68) and Table 3-8 (page 68).

Figure 3-9 Coprocessor Self-Test Control and Status register, write-only bits



Note: Attempts to write other bits are ignored.

Table 3-8 Coprocessor Self-Test PROC_CONTROL values

PROC_CONTROL Value	Description
0	None
1	Run X.509 certificate and private key tests
2-7	Reserved

When read from, bits 7-4 of the Self-Test Control and Status register report the results of the X.509 certificate and private key tests, as shown in Figure 3-10 (page 68) and Table 3-9 (page 68). The CP detects a read cycle and resets the Control and Status register to 0x00 after it; hence bits 7-4 must all be retrieved in one operation.

Figure 3-10 Coprocessor Self-Test Control and Status register, read-only bits

Table 3-9 Coprocessor Self-Test Results bits

Self-Test Results Bit	Test	Meaning If 0	Meaning If 1
7	X.509 Certificate	Certificate not found	Certificate found in memory
6	Private key	Private key not found	Private key found in memory
5-4	Reserved		

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Note: The X.509 and private key tests only verify that these elements are present in Flash memory; no authentication is performed.

3.8.2.14 System Event Counter

The System Event Counter (SEC) is a non-volatile register that holds the current value of the CP's event counter. The event counter automatically decrements one count per second while the CP is powered, stopping at 0. If the accessory controls power to the CP, it must wait until the SEC has decremented to 0 before removing power.

3.8.2.15 Apple Device Certificate Data Length

The Apple Device Certificate Data Length register holds the length of the X.509 certificate supplied by the attached Apple device. An accessory uses this certificate to authenticate an Apple device in both the certificate validation and challenge response verification processes. The length of an Apple device certificate varies but is always less than or equal to 1024 bytes. This length limit may not hold for future versions of the authentication protocol.

Writing a value in a range greater than 0 and less than or equal to 1024 will cause the CP to validate the data contained in the iPod Certificate Data registers. If the CP invalidates the iPod Certificate Data, it sets this register to 0.

3.8.2.16 Apple Device Certificate Data

The Apple Device Certificate Data register holds the X.509 Certificate that an accessory uses to authenticate an Apple device in both the certificate validation and challenge response verification processes. The Apple Device Certificate may be written to the coprocessor in 128-byte pages starting at any Apple Device Certificate Data Page address, but it may not be written in a multipage stream. Since the length of the Apple Device Certificate varies, not all of the pages need to be used. The Apple Device Certificate Data Length value determines which Apple Device Certificate Data Pages contain valid certificate data.

3.9 Coprocessor 2.0C I2C Protocol

The Apple Authentication Coprocessor (CP) supports the I2C communication protocol, acting as an I2C slave. Its SCL signal is the I2C clock line and is driven by the accessory. Its SDA signal is the I2C data line and is driven by whichever device is currently sending data.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Unlike the Apple Authentication Coprocessor version 2.0B, CP 2.0C does not perform clock synchronization by stretching SCL. It may, however, not-acknowledge (NACK) a requested register operation if busy, so the I2C master should expect retry operations as a normal part of CP 2.0C use.

The maximum supported I2C clock rate is 400 kHz.

3.9.1 Slave Selection and Reset

During reset, the RST signal must specify the CP's I2C slave address and must be held stable for at least 10*ms* after power-up or reset, as described in I2C Communications Process (page 59). As an I2C slave, the CP is then selected in-band via its I2C address. The least significant bit of the I2C slave address controls whether a write or a read operation is to be performed, as described in Coprocessor 2.0C Address Selection (page 55).

3.9.2 Coprocessor Busy

When the CP is busy processing it is unable to handle incoming communication attempts. If the coprocessor does not ACK its slave address during an attempted I2C communication, then the coprocessor is busy. The accessory must repeatedly attempt communication until the coprocessor sends an ACK after receiving its slave address.

3.9.3 Writing to the Coprocessor

To write data to the coprocessor, follow these steps:

1. Send the I2C start sequence.
2. Send the I2C write address of the CP.
3. Check for an ACK from the slave; if a NACK is received, wait 500 µ*s* and then loop back to Step 1.
4. Send the register address at which to begin writing.
5. Send the data bytes.
6. Send the I2C stop sequence.

3.9.4 Reading from the Coprocessor

To read data from the coprocessor, follow these steps:

1. Send the I2C start sequence.
2. Send the I2C write address of the CP.
3. Check for an ACK from the slave; if a NACK is received, wait 500 µ*s* and then loop back to Step 1.
4. Send the register address at which to begin reading.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

5. Optional: send the I2C stop sequence.
6. Send the I2C start sequence.
7. Send the I2C read address of the CP.
8. Check for an ACK from the slave; if a NACK is received, wait 500 µs and then loop back to Step 6.
9. Read the data bytes.
10. Send the I2C stop sequence.

Any additional reads after an I2C read stop sequence continue with the byte following the previous byte read until an invalid register address or an end of block is reached, at which point the slave returns 0xFF in response to all further reads.

3.10 Coprocessor 2.0C Device Characteristics

This section provides technical details and tolerances for the Apple Authentication Coprocessor 2.0C (CP) chip.

3.10.1 Physical Configuration

Figure 3-11 (page 71) shows the CP package's layout, pin locations, and dimensional tolerances.

Figure 3-11 Coprocessor package

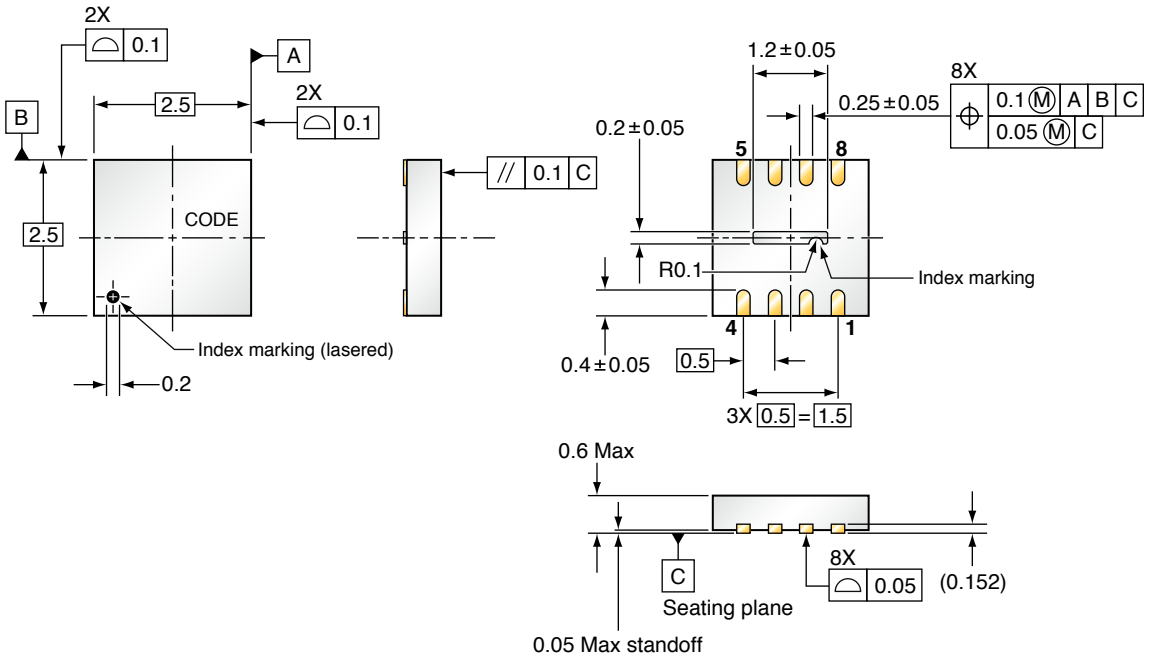


Figure 3-12 (page 72), Figure 3-13 (page 73), Figure 3-14 (page 74), Figure 3-15 (page 75), and Figure 3-16 (page 76) describe how the CP is delivered to accessory manufacturing facilities.

Figure 3-12 Coprocessor Packing Carrier Tape

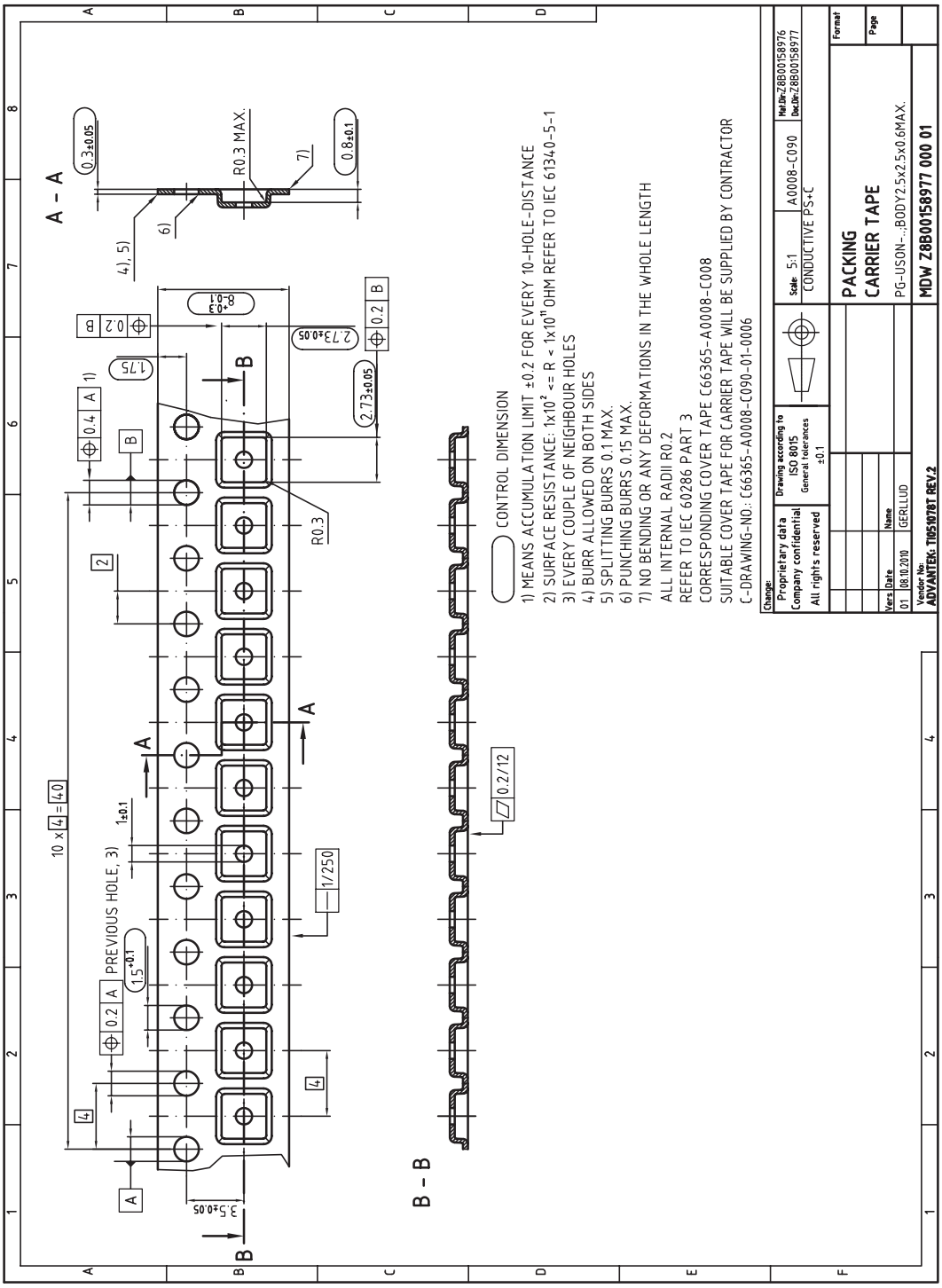


Figure 3-13 Coprocessor Packing Reel (1,000 unit)

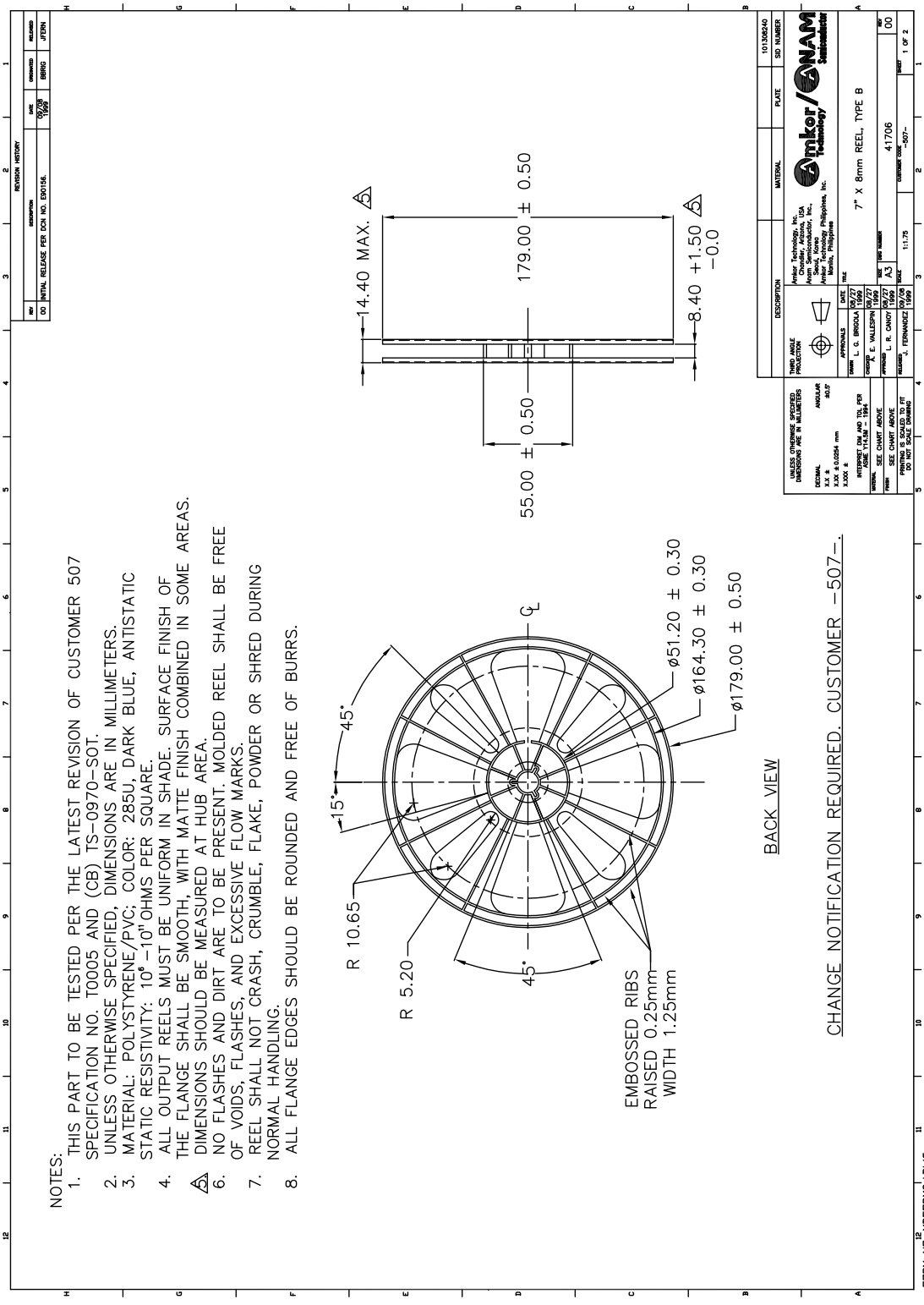


Figure 3-14 Coprocessor Packing Reel (1,000 unit) details

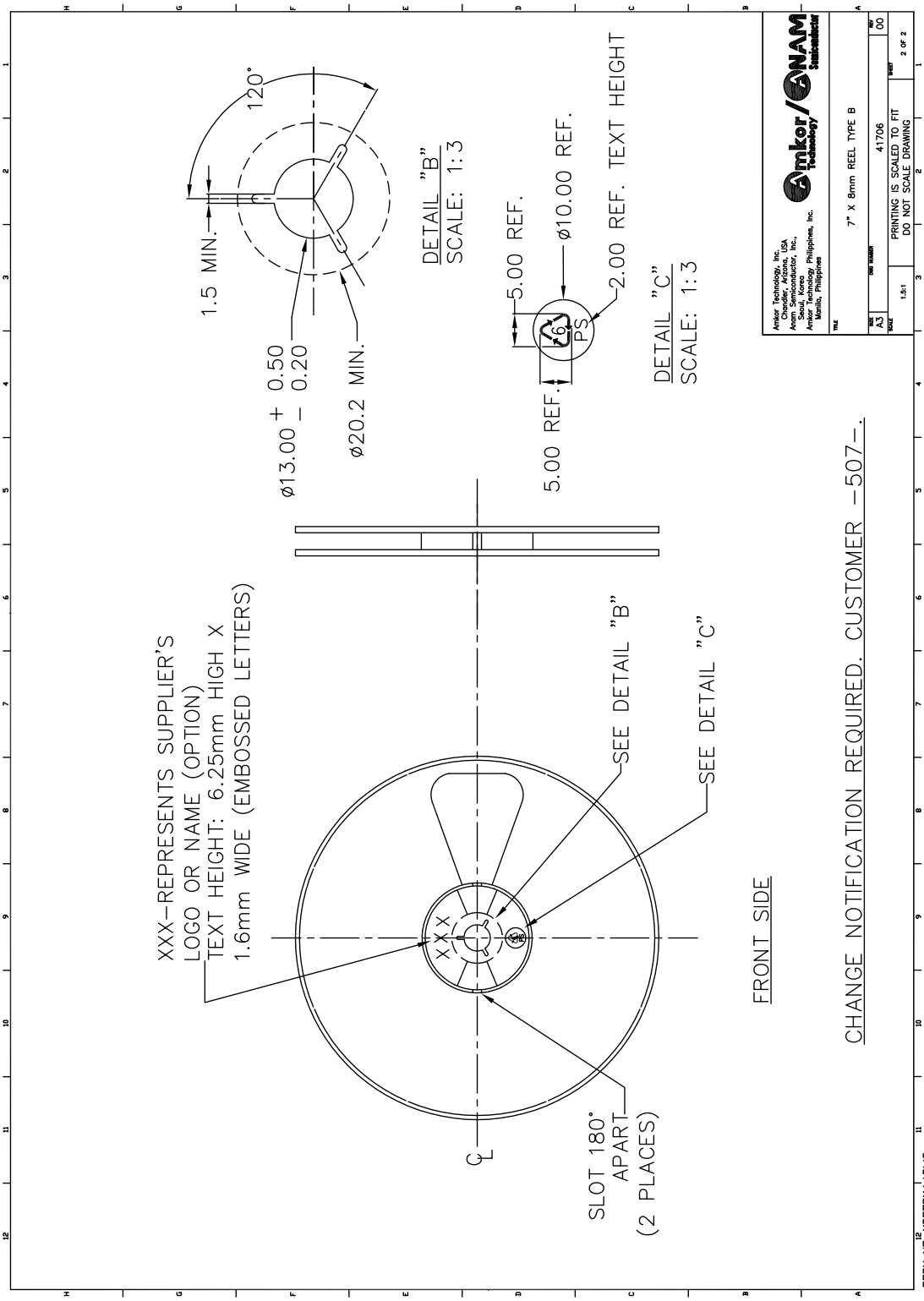


Figure 3-15 Coprocessor Packing Reel (10,000 unit)

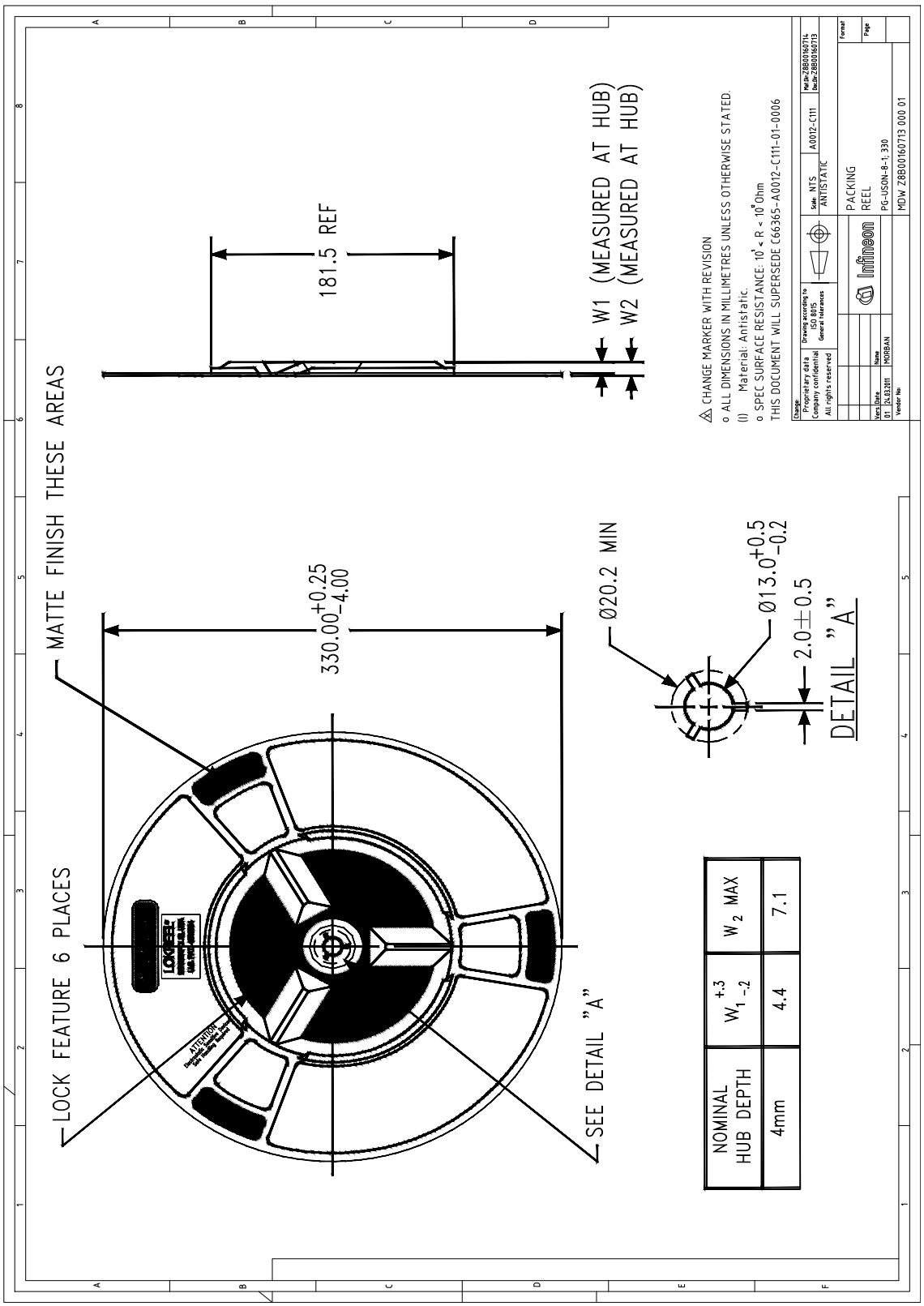
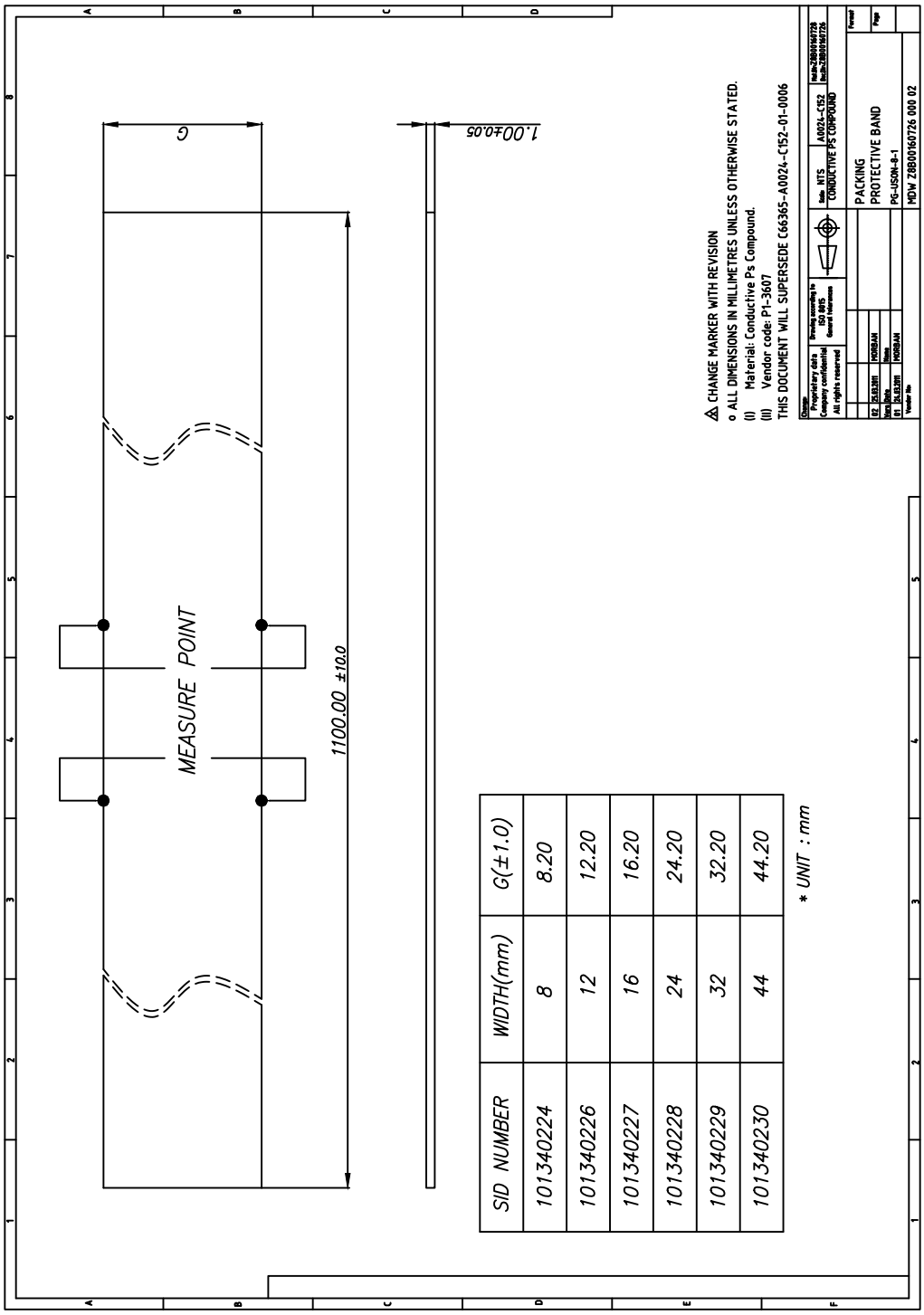


Figure 3-16 Coprocessor Packing Protective Band



SID NUMBER	WIDTH(mm)	G(±1.0)
101340224	8	8.20
101340226	12	12.20
101340227	16	16.20
101340228	24	24.20
101340229	32	32.20
101340230	44	44.20

Copyright © 2014 Apple Inc. All Rights Reserved.

3.10.2 Maximum Environmental Conditions

Table 3-10 (page 77) lists the CP's absolute maximum electrical and free-air temperature ranges. Stresses to the CP chip beyond the ranges listed in Table 3-10 (page 77) may cause permanent damage. Exposure to either end of any range for extended periods may affect device reliability.

Table 3-10 Coprocessor maximum electrical and temperature ranges

Condition	Maximum Range
Voltage applied at V_{CC} relative to V_{SS}	-0.3 V to +7.0 V
Voltage applied to any pin	-0.3 V to V_{CC} + 0.3 V
Storage temperature	-40 °C to +125 °C

3.10.3 Recommended Operating Conditions

The CP is available only in a standard temperature range configuration. Internal sensors may force it to its reset state if any of the conditions listed in Table 3-11 (page 77) are exceeded. Attempting to operate the CP in this state is not recommended and may lead to device failure or unreliability.

Table 3-11 Coprocessor maximum electrical and temperature ranges

Condition	Recommended Range
Operating free-air temperature	-25 to +85 °C
Supply voltage during program execution	1.62 to 5.5 V

3.10.4 I2C Interface Characteristics

Table 3-12 (page 77) specifies the limits of the I2C interface between the CP and other components.

Table 3-12 Coprocessor I2C interface characteristics

Parameter	Required Range
SCL clock frequency (f_{SCL})	10 to 400 kHz
External bus capacitance to ground (C_b)	Maximum 100 pF

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

3.10.5 DC Electrical Characteristics

Table 3-13 (page 78), Table 3-14 (page 78) and Table 3-15 (page 78) show the DC electrical characteristics of the CP chip over its recommended voltage and temperature ranges. Unless otherwise specified in these tables, V_{CC} = 1.62 to 5.5 V and T_A = -25 °C to +85 °C.

Table 3-13 Coprocessor supply current into V_{CC}, excluding external current

Parameter	Test Conditions	Minimum	Typical	Maximum	Unit
$I_{(AM)}$ Active mode (authentication process running)			6	7.5	mA
$I_{(sleep)}$ Sleep mode	T_A =25 °C		35	80	µA

Table 3-14 Coprocessor inputs

Symbol	Parameter	Minimum	Typical	Maximum	Unit
V_{IH}	High-level input voltage	$V_{CC} \times 0.7$		$V_{CC} + 0.3$	V
V_{IL}	Low-level input voltage	-0.3		$V_{CC} \times 0.2$	V
I_i	Input current	-10	1	10	µA

Table 3-15 Coprocessor outputs

Symbol	Parameter	Test Conditions	Minimum	Maximum	Unit
V_{OL}	Low-level output voltage	$I_{OL(max)}$ = -1 mA	V_{SS}	V_{SS} + 0.4	V

3.10.6 Timing Characteristics

When power is turned on to the CP, V_{CC} must reach 90% of the CP's target supply voltage within 200 µs after it exceeds 400 mV.

Figure 3-17 (page 79) illustrates the CP's typical I/O port input signal timing and voltage limits. Table 3-16 (page 79) lists the parameter values in Figure 3-17 (page 79).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 3-17 Coprocessor typical I/O port input waveform

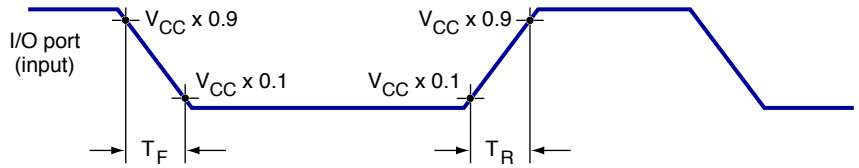


Table 3-16 Coprocessor Values for typical I/O port input waveform

Symbol	Description	Maximum Value
T_F	Fall time	1.0 µs
T_R	Rise time	1.0 µs

4. Apple Lightning Audio Module

Accessories may make use of the Apple Lightning Audio Module (LAM) to create the following types of accessories that plug into the Lightning connector on Apple devices running iOS 7.1 or later:

- headphones (see Headphones (page 378))
- accessories with an integrated 3.5 mm headphone jack, but not integrated speakers
- non-audio accessories

Accessories using the LAM module may:

- Draw power from the Apple device, even when asleep, and eliminate the need for an internal battery
- Receive lossless stereo 48 kHz digital audio output from the Apple device
- Send mono 48 kHz digital audio input to the Apple device (see note)
- Provide power to the Apple device from an internal battery and/or a pass-through external power source
- Implement Apple Headphone Remote Volume Up/Volume Down/Center buttons
- Implement additional buttons that work with iTunes Radio and other iOS media playback features
- Declare a preferred iOS app that works with the accessory
- Communicate with the preferred iOS app via an External Accessory Protocol
- Request that the preferred iOS app be launched on the Apple device

> **Note:** iOS 7.1 does not support the audio input feature. A future release of iOS will add this support. Accessory design requirements related to audio input may change in future specification revisions.

The Lightning Audio Module supports two accessory integration approaches:

- Standard LAM Accessories may be developed with a minimum of additional components when paired with a DAC supported by the Lightning Audio Module.
- Advanced LAM Accessories with digital audio processing features, such as active noise cancellation, may be developed when paired with a DSP and DAC.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 4-1 Standard LAM Accessory without Power

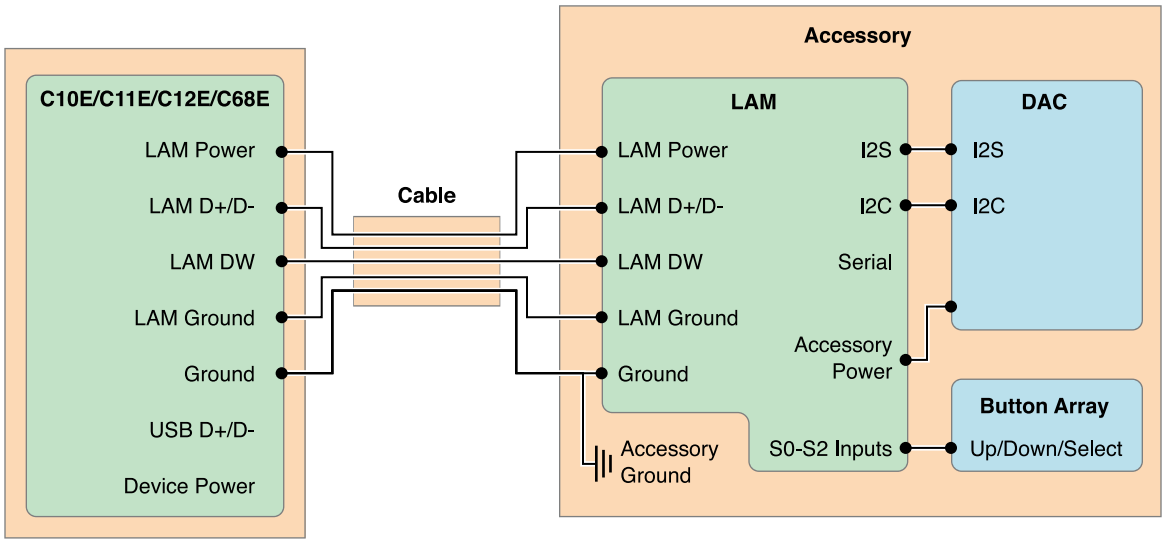
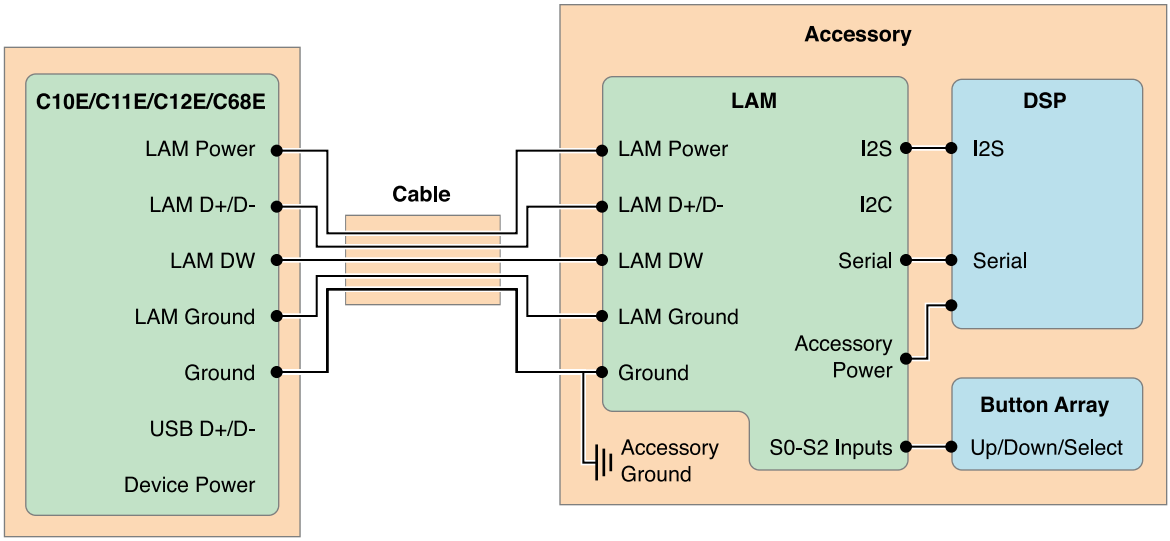


Figure 4-2 Advanced LAM Accessory without Power



Accessory designs that implement an integrated 3.5 mm headphone jack but do not include integrated speakers may also use the Lightning Audio Module to pass analog audio to or from the Apple device. These accessories must comply with the following requirements:

- They must comply with all requirements that apply to Advanced LAM Accessories.
- They may physically block or occlude the Apple device's headphone jack.
- They must not physically block or occlude the Apple device's built-in microphones or speakers.
- They must set the Audio Output Terminal Type and Audio Input Terminal Type to Headphone Jack. See accAudioTerminalInformation (0x0B) (page 102).

Copyright © 2014 Apple Inc. All Rights Reserved.

- They must use the accAudioTerminalStateInformation (0x23) (page 108) packet to indicate when headphones/microphone are plugged into or unplugged from the accessory headphone jack.

Accessory designs that do not integrate audio features may also use the Lightning Audio Module. These accessories must comply with the following requirements:

- They must comply with all requirements that apply to Advanced LAM Accessories.
- They must not physically block or occlude the Apple device's built-in microphones, speakers, or headphone jack.
- They must set the Audio Output Terminal Type and Audio Input Terminal Type to None. See accAudioTerminalInformation (0x0B) (page 102).
- They must set the Audio Output Terminal State and Audio Input Terminal State to Disabled. See accAudioTerminalStateInformation (0x23) (page 108).

4.1 Overview

Key accessory-facing interfaces of the Lightning Audio Module are:

- I2S audio (Host or Slave)
- Serial
- I2C (Host)
- 3 digital inputs (S0, S1, and S2 buttons)
- Accessory Power

4.2 Firmware Updates

Advanced LAM Accessories must be capable of field-deployable firmware updates from Apple devices running iOS and/or Macintosh computers running OS X.

4.3 Authentication

Accessories that integrate the Lightning Audio Module do not need to integrate an Apple Authentication Coprocessor.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

4.4 Lightning (C10E) or Lightning (C68E) Connector

Accessories that integrate the Lightning Audio Module must attach to the Apple device using a cable that integrates a Lightning (C10E) or Lightning (C68E) connector (see Apple Lightning Connector (page 114)).

4.5 Mechanical

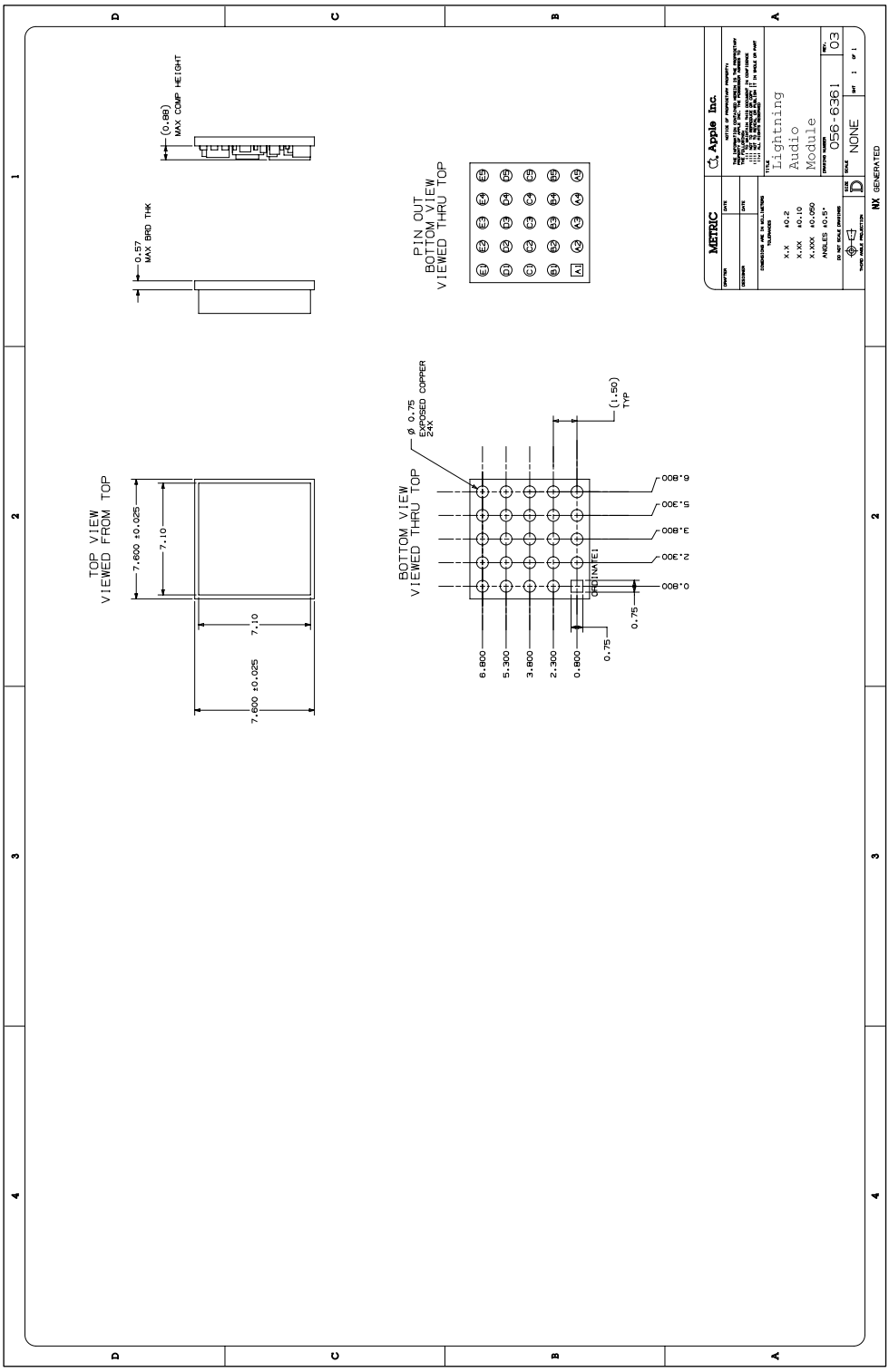
The Lightning Audio Module has the following mechanical characteristics:

- 7.6 mm x 7.6 mm LGA package on tape and reel
- Not encapsulated in order to maximize compatibility with pick and place operations
- Pickup is offset to the CSP surface
- A1 pad has a fiducial mark that may be used as a reference
- FR4 PCB material
- SAC305 solder is used. Solder reflow operations must take this into consideration
- 0-70 degrees C working temperature range

Note: Encapsulation of the Lightning Audio Module is recommended to pass salt spray environmental testing.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

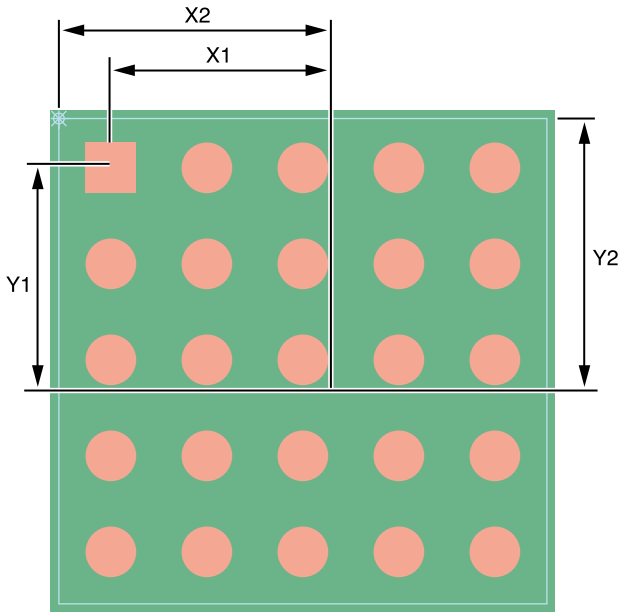
Figure 4-3 Lightning Audio Module Mechanical Package



Pickup offsets for the Lightning Audio Module (see Figure 4-4 (page 85)) are:

- (X1) 3.4635075 mm
- (Y1) 3.513225 mm
- (X2) 4.258955 mm
- (Y2) 4.30867 mm

Figure 4-4 Lightning Audio Module pickup offsets (bottom view)



4.6 Pad Layout and Assignments

The following diagrams detail the layout, names, and description of the Lightning Audio Module LGA pads. All Reserved pads must be soldered but left Not Connected (NC).

Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 4-5 Lightning Audio Module pad layout (top view)

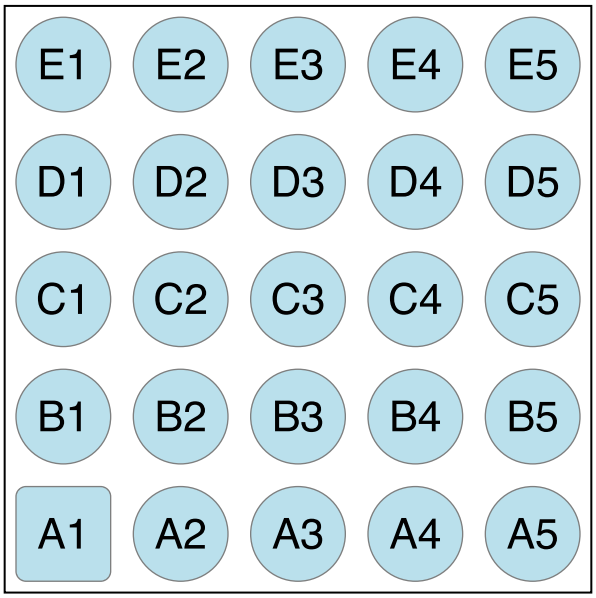


Table 4-1 Lightning Audio Module pad assignments

Pad	Name	Assignment
A1	Ground	see Cable (page 88)
A2	Reserved	
A3	S2	see S0-S2 Headphone Remote Inputs (page 95)
A4	Reserved	
A5	LAM Power	see Cable (page 88)
B1	Accessory Power	see Accessory Power (page 90)
B2	I2S MCLK	see I2S (page 93)
B3	S1	see S0-S2 Headphone Remote Inputs (page 95)
B4	I/O VREF	see Input/Output Electrical Characteristics (page 87)
B5	LAM D+	see Cable (page 88)
C1	Serial TX	see Serial (page 96)
C2	I2S SCLK	see I2S (page 93)
C3	S0	see S0-S2 Headphone Remote Inputs (page 95)

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Pad	Name	Assignment
C4	Reserved	
C5	LAM D-	see Cable (page 88)
D1	Serial RX	see Serial (page 96)
D2	I2S LRCLK	see I2S (page 93)
D3	I2C SCL	see I2C (page 95)
D4	Reserved	
D5	LAM Ground	see Cable (page 88)
E1	I2S SDOUT	see I2S (page 93)
E2	I2S SDIN	see I2S (page 93)
E3	I2C SDA	see I2C (page 95)
E4	LAM DW	see Cable (page 88)
E5	Ground	see Cable (page 88)

4.7 Input/Output Electrical Characteristics

The following electrical characteristics apply to the Lightning Audio Module I2S pads:

Table 4-2 Lightning Audio Module I2S pad electrical characteristics

Characteristic	Min	Max
High-level input	1.26 V	2.1 V
Low-level input	-0.3 V	0.54 V
High-level output	1.6 V	
Low-level output		0.2 V

The following electrical characteristics apply to all other Lightning Audio Module input/output pads.

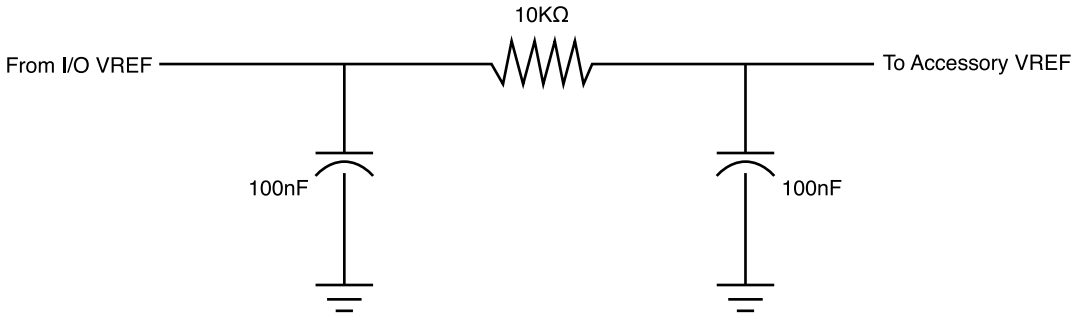
Copyright © 2014 Apple Inc. All Rights Reserved.

Table 4-3 Lightning Audio Module I/O pad (except I2S) electrical characteristics

Characteristic	Min	Max	Notes
High-level input	1.26 V	2.10 V	
Low-level input	-0.30 V	0.54 V	
High-level output	1.35 V		Max 1 mA, VDD = 1.8 V
Low-level output		0.45 V	Max 2 mA, VDD = 1.8 V

I/O VREF is a 1.8 V output and may be used as a voltage reference for the I/O level when the accessory's circuits operate with voltage levels different from the CMOS 1.8 V expected by the Lightning Audio Module. The accessory must not draw power from this pad. A RC filter as shown in Figure 4-6 (page 88) must be added before connecting it to the accessory's reference input.

Figure 4-6 I/O VREF RC circuit



The Serial RX input has the following additional characteristics:

Table 4-4 Lightning Audio Module Serial RX electrical characteristics

Characteristic	Typical	Worst Case
Capacitance	12 pF	15 pF
Input leakage	300 nA	1 uA

4.8 Cable

Accessory cables or interconnects that connect a Lightning (C10E) or Lightning (C68E) connector to a Lightning Audio Module must comply with the requirements in this section.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

The following signals must be connected from the Lightning Audio Module to the Lightning (C10E) or Lightning (C68E) connector using separate conductors.

- LAM Power
- LAM D+
- LAM D-
- LAM Ground
- LAM DW
- Ground

There is one exception. If all of the following are true:

- The accessory does not provide power to the Apple device (see Device Power (page 91)).
- Ground DCR ≤ 0.4 Ω.
- The maximum current through this path is 100 mA.

Then,

- LAM Ground and Ground conductors may be omitted from the cable or interconnect.
- LAM Ground and Ground pads must be connected to a cable shield drain wire.

LAM Ground must only be used as a non-current-carrying ground reference.

The following requirements also apply:

- LAM D+ and LAM D- must be a twisted pair and shielded using LAM Ground or Ground (if LAM Ground is not present).
- LAM D+ and LAM D- differential impedance nominally 90 Ω (85 Ω min, 95 Ω max).
- LAM Power DCR ≤ 1.0 Ω.

4.9 Power

4.9.1 Power States

The Lightning Audio Module will mirror Apple device power states with the following exceptions:

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- If configured to do so, the Lightning Audio Module will continue to provide power to the Lightning accessory even when the Lightning Audio Module and Apple device are sleeping.
- Direct user action on button S0 will wake the Lightning Audio Module and Apple device from a sleep state.
- Incoming packets from an Advanced LAM Accessory via the serial protocol (see Serial Protocol (page 97)) will wake the Lightning Audio Module, but not the Apple device, from a sleep state.
- accHIDReport serial protocol packets from an Advanced LAM Accessory (see accHIDReport (0x21) (page 107)) will wake the Lightning Audio Module and Apple device from a sleep state.

Note: Lightning accessories must not force the Apple device to wake from a sleep state autonomously or prevent it from entering a sleep state. This is grounds for failure to pass self certification.

4.9.2 Module Power

The Lightning Audio Module is powered from the Lightning (C10E) or Lightning (C68E) connector. Lightning accessories do not need to provide power to the Lightning Audio Module.

4.9.3 Accessory Power

Accessories may draw power from the Apple device via the Lightning Audio Module's Accessory Power pad. Additionally, they may configure the Lightning Audio Module to provide constant Accessory Power even when the Apple device is in a sleep state. This may enable them to implement advanced features such as Active Noise Cancellation (ANC) without incorporating an internal power source. The Accessory Power pad must not be used to charge an accessory's internal battery.

Note: Accessories must not draw power directly from the LAM Power pad on the Lightning (C10E) or Lightning (C68E) connector; all power must be drawn from the Accessory Power pin on the Lightning Audio Module.

Apple strongly recommends keeping the power draw as low as possible to maximize the lifetime of the Apple device's internal battery. If the accessory does not perform any user-visible or user-audible function (such as ANC) when the Apple device is asleep, it must not configure the Lightning Audio Module to provide constant Accessory Power. Instead, the Lightning Audio Module will stop providing Accessory Power when audio streaming has stopped or the Apple device enters a sleep state. It will re-enable Accessory Power when audio streaming restarts.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Accessories that configure the Lightning Audio Module to provide constant accessory power should implement the following steps to minimize power draw:

- When the Audio Output Stream Status field in the lamInformation packet (see lamInformation (0x10) (page 104)) is 0, the accessory should power down amplifiers, DACs, and other audio processing circuits but leave its UART operational and ready to process incoming packets.
- The accessory should monitor the SDA pad as specified in I2S (page 93). If SDA is low for more than 500 ms, the accessory may enter a deep sleep state, and should exit that state when SDA is subsequently driven high.

The voltage supply range from the Accessory Power pad is 2.5 V-4.7 V (3.3 V typical).

- Average current draw from the Accessory Power pad must not exceed 50 mA.
- Peak current draw from the Accessory Power pad must not exceed 100 mA with one exception. When the Lightning Audio Module enables Accessory Power, the accessory may exceed the 100 mA current limit until 5 ms has elapsed or 100 uF of capacitance has been charged, whichever comes first.

I2S MCLK will stay active as long as the Lightning Audio Module provides Accessory Power.

> **Note:** To minimize the risk of audible artifacts, Lightning accessories must immediately mute all audio interfaces when they detect a loss of Accessory Power.

4.9.4 Device Power

Lightning accessories may optionally provide power to the Apple device via the Lightning (C10E) or Lightning (C68E)'s Device Power, USB D+, USB D-, and Ground pads from an internal or external USB power source. The power source must communicate how much power is available to the Apple device using D+/D- resistors.

Additionally, the power source's ground must be directly connected to the Ground pad on the Lightning (C10E) or Lightning (C68E) connector and isolated from LAM Ground as shown:

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 4-7 Standard LAM Accessory with Power

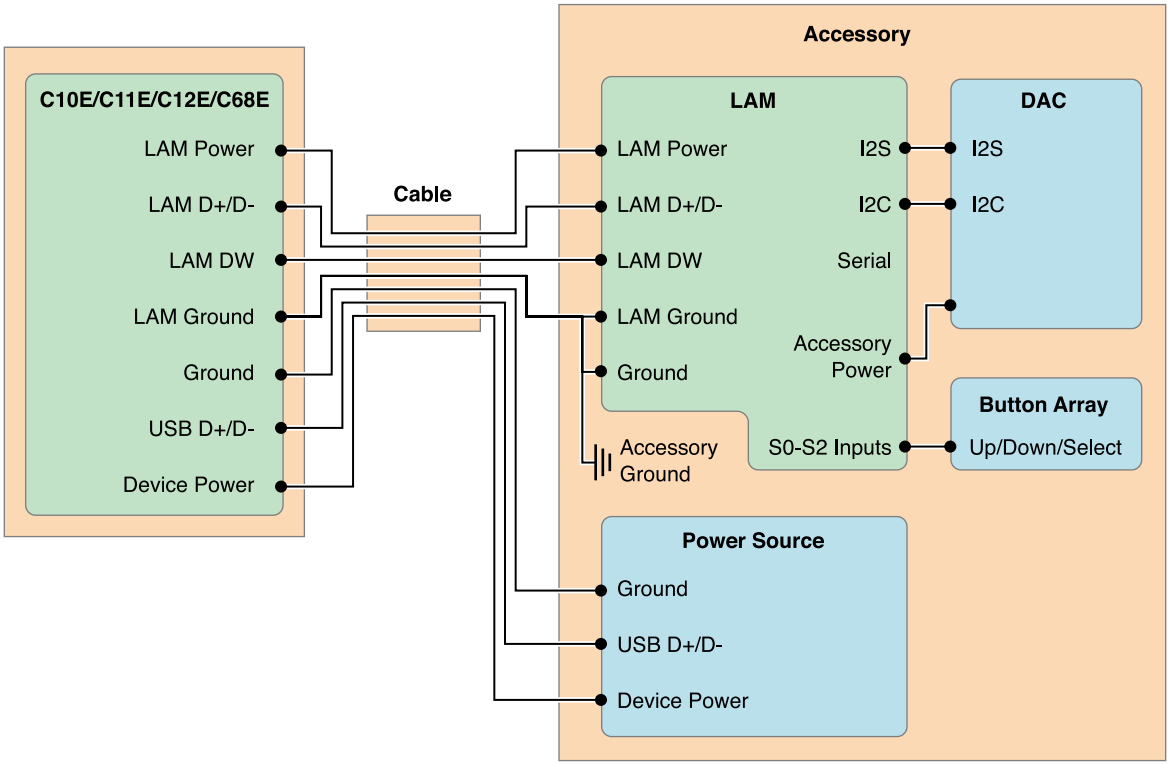
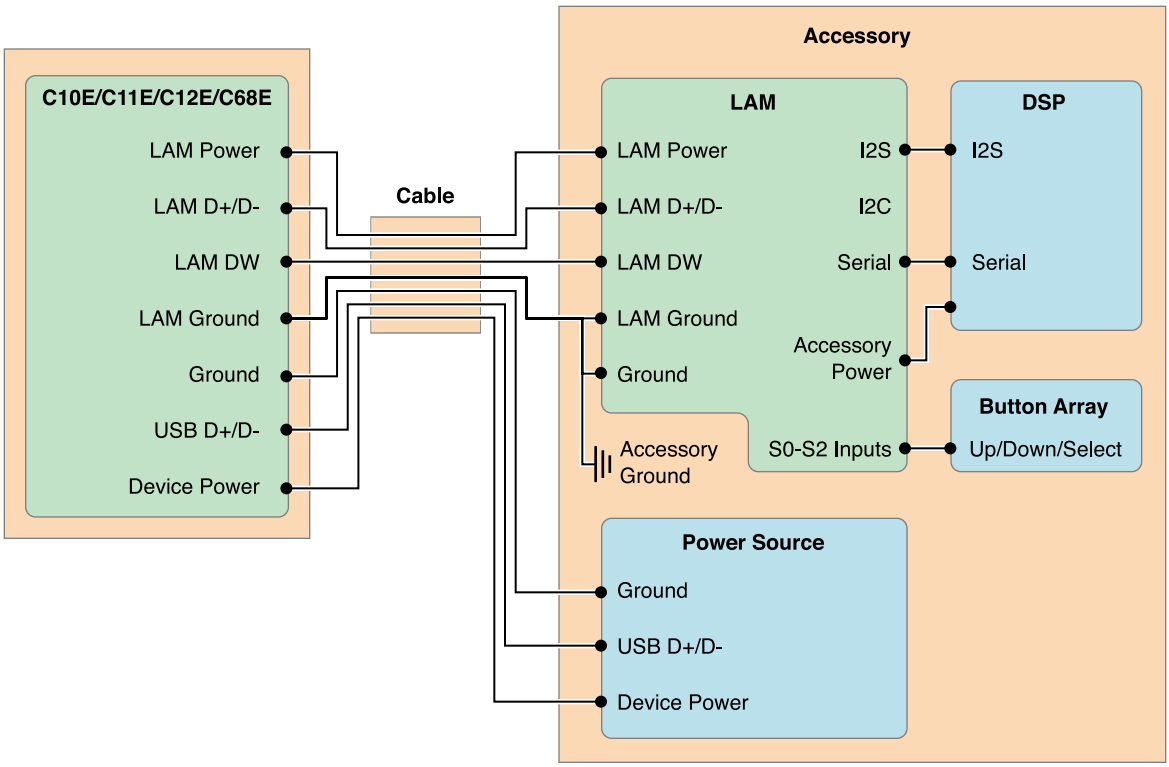


Figure 4-8 Advanced LAM Accessory with Power



Copyright © 2014 Apple Inc. All Rights Reserved.

If an Advanced Lightning accessory has an internal battery, it may draw some current from an external power source to recharge that battery while the Apple device is also charging, but must inform the Apple device of the current draw using accPassthroughPowerSourceInformation (0x24) (page 109). Additionally, the amount of current that may be drawn depends on the Apple device's power state as specified in lamInformation (0x10) (page 104).

4.10 Audio

4.10.1 DAC

Advanced LAM Accessories may use any DAC that works with their DSP.

Standard LAM Accessories must use a DAC from the following list:

- Wolfson WM8533

The Lightning Audio Module expects the WM8533 to be at I2C address 0x34, so the WM8533 address select pin must be pulled low.

4.10.2 I2S

Advanced LAM Accessories may consume audio from the Lightning Audio Module via I2S. The Lightning Audio Module can either be a I2S host or I2S slave. I2S slave mode is the default.

The I2S audio interface consists of five signals:

- *MCLK*: Master clock
- *SCLK*: Serial data shift clock
- *LRCLK*: Left/right clock
  - Identifies where each channel (left or right) is located within the data word
  - Toggles at the audio sampling rate
- *SDOUT*: Serial data output
- *SDIN*: Serial data input

When the I2S audio interface is operating in slave mode, SCLK, and LRCK are inputs. Conversely, SCLK and LRCLK are outputs when the I2S audio interface is operating in host mode. LRCLK is 48 kHz. MCLK is always an output regardless of slave or host mode and is 6.144 MHz.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

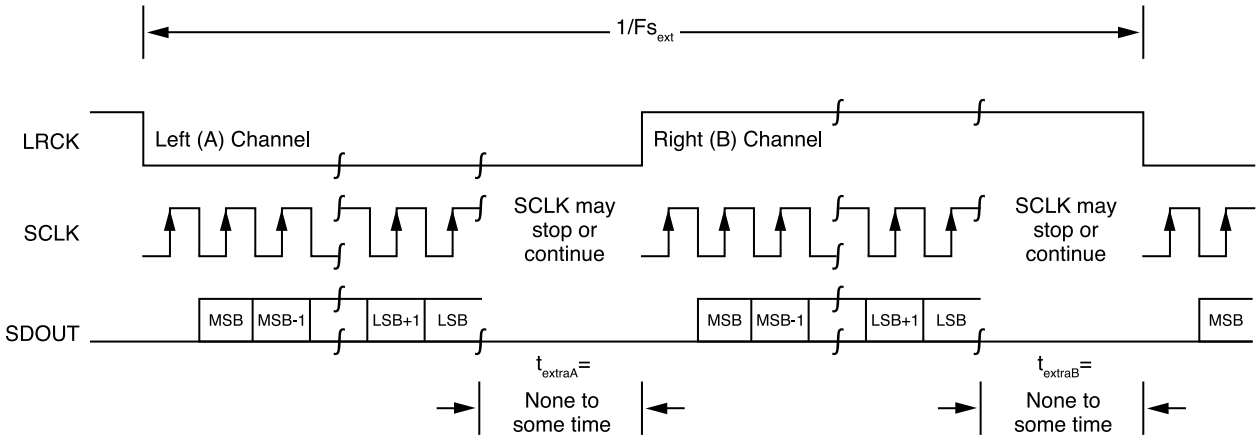
For Standard LAM accessories, the 128:1 ratio between MCLK and LRCLK is supported directly by WM8533. No PLL is required.

The I2S interface only supports a 48 kHz sample rate.

The I2S data stream has the following characteristics:

- Up to 24 bits/sample of stereo data may be transported.
- LRCK identifies the start of a new sample word and the active stereo channel (A or B).
- Data is clocked out of the SDOUT output using the falling edge of SCLK.
- Bit order is MSB to LSB.

Figure 4-9 Lightning Audio Module I2S Format



Advanced LAM Accessories may treat a low level on the I2C_SDA pad to indicate possible errors in the I2S data stream. Apple recommends that this signal be used to mute the audio DAC. The signal will remain low until the I2S data stream is deemed free of errors by the Lightning Audio Module. An external pull-up resistor must be attached to the I2C_SDA pad if it is used in this manner. A resistor value of 10 kΩ - 47 kΩ is recommended. If the I2C_SDA pad is not used in this manner, it must be connected as described in I2C (page 95).

If the accessory does not contain a microphone, it must connect SDIN to ground. If it does contain a microphone, the I2S audio input stream must comply with all of the following requirements:

- The accessory must natively sample audio at a rate of at least 48 kHz.
- Each sample must have at least 16 bits of data. 24 is recommended.
- The data on left and right channels must be identical.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

4.11 Control

Lightning accessories must take one of two possible approaches for implementing user controls.

4.11.1 S0-S2 Headphone Remote Inputs

Lightning accessories that make use of the S0-S2 headphone remote inputs must connect to the following Lightning Audio Module pads:

- *S0*: Select
- *S1*: Volume Down
- *S2*: Volume Up

The S0-S2 inputs default to a high impedance input "1". A button press corresponds to "0". All S0-S2 activity must occur as a result of direct user action.

The Lightning Audio Module provides internal pull-ups for S0, S1, and S2. No external pull-ups are required.

4.11.2 HID Headphone Remote

Advanced LAM Accessories that implement a HID Headphone Remote component (see HID Headphone Remote (page 441)) must register a HID descriptor with the Lightning Audio Module and send HID usage reports to Apple devices using the serial protocol (see accHIDComponentInformation (0x0A) (page 101) and accHIDReport (0x21) (page 107)). All HID usage reports must be generated as a result of direct user action.

When the Apple device is in a sleep state, Advanced LAM Accessories must only send a accHIDReport packet if the user takes direct action to press the Center button on the headphone remote. All other button presses must be ignored and accHIDReport packets must not be sent.

4.11.3 I2C

Standard LAM Accessories must use the I2C interface and not the Serial interface. They must integrate a supported DAC and allow the module to control the DAC directly. The LAM controls the DAC volume, so any headphone amplifier connected after the DAC should be set for constant gain.

To use the I2C interface, Standard LAM Accessories must connect 2.2 kΩ or 4.7 kΩ pull up resistors to I2C_SCL and I2C_SDA on the Lightning Audio Module.

Advanced LAM Accessories must not use the I2C interface for control purposes. Instead:

- The I2C_SCL pad must be connected to ground via a 1 kΩ-10 kΩ resistor.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- The I2C_SDA pad must be connected to ground via a 1 kΩ-10 kΩ resistor if it is not used as a warning of possible errors in the I2S audio stream as specified in I2S (page 93).

4.11.4 Serial

Advanced LAM Accessories must use the Serial interface and not the I2C interface.

Standard LAM Accessories may leave the serial interface unconnected as RX is internally pulled high.

The serial interface consists of two signals:

- *RX*: Receive
- *TX*: Transmit

The serial interface operates at either 57600 bps or 115200 bps. By default, the interface is configured to operate at 57600 bps. All Lightning Audio Module serial communications use 8 data bits, no parity bits, and one stop bit (8-N-1). The Lightning Audio Module does not support serial hardware flow controls (RTS/CTS and DTR/DSR).

The serial interface protocol offers the following features:

- Configuration of Accessory Identification information (only during manufacturing or once after firmware update)
  - Manufacturer, Model, Serial Number, and HW/FW Version
  - Preferred App Bundle Seed ID
  - External Accessory Protocol name
  - Accessory Power Mode
- Lightning Audio Module status updates
- External Accessory Protocol data transfer
- HID Media Remote Control user inputs
- Request Preferred App launch on Apple device
- Inform Apple device of changes in accessory configuration

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

4.12 Serial Protocol

4.12.1 Packet Format

The format for Lightning Audio Module serial interface packets is as follows:

Table 4-5 Lightning Audio Module serial protocol packet

Field	Bytes	Description
Wake	1	Optional. If present, must be `0xFF`
Start of Packet	1	Must be `0x55`
Length	1	Number of bytes in Type and Payload fields. Must be 129 or less.
Type	1	Packet type
Payload	0-128	Packet payload data, dependent on packet type
Checksum	1	Checksum byte

Accessories should send the Wake byte only if the accessory is self powered or has configured the Lightning Audio Module for constant Accessory Power. In that situation, the Lightning Audio Module will make use of the Wake byte to recover from a sleep state and correctly parse the rest of the packet.

The Checksum field is calculated by adding together the values of the Length, Type, and Payload bytes as signed 8-bit values (discarding any signed 8-bit overflow) and then negating that sum to create the signed 8-bit checksum byte. The sum of all bytes in the following packet fields must add up to 0x00:

- Length
- Type
- Payload
- Checksum

All packets with an invalid checksum are presumed invalid and must be ignored by both the Lightning Audio Module and the Lightning accessory.

4.12.2 Initialization Packets

These packets confirm mutual readiness to initiate communication between the Lightning Audio Module and the Lightning accessory.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

4.12.2.1 accReady (0x01)

No payload.

Before sending any other packets, the Lightning accessory must confirm the operating state of the Lightning Audio Module by sending this packet once a second until a lamReady (0x02) (page 98) packet is received in response.

4.12.2.2 lamReady (0x02)

No payload.

The Lightning Audio Module will send this packet when it has received a accReady (0x01) (page 98) packet or a configuration packet (see Configuration Packets (page 98)) from the Lightning accessory and is ready to start accepting other packets.

4.12.3 Configuration Packets

Configuration packets must only be sent during factory configuration or immediately after a firmware update.

Accessories must receive a lamReady packet from the Lightning Audio Module after every configuration packet before sending other packets.

By default, the Lightning Audio Module is configured to operate as a Standard LAM Accessory. To pass certification audit, the model name, serial number, etc. must be configured properly.

4.12.3.1 accConfigurationInformation (0x03)

Table 4-6 Lightning Audio Module accConfigurationInformation packet payload

Field	Bytes	Factory Default	Description
LAM Protocol Version	1	1	Must be set to 1
Accessory Power Mode	1	0	0 = Intermittent, 1 = Constant
I2S Role	1	0	0 = Slave, 1 = Host
I2C DAC	1	1	0 = None, 1 = WM8533
UART Baud Rate	1	0	0 = 57600 bps, 1 = 115200 bps
Hardware Major Version	1	0	
Hardware Minor Version	1	0	

Copyright © 2014 Apple Inc. All Rights Reserved.

Field	Bytes	Factory Default	Description
Hardware Revision	1	0	
Firmware Major Version	1	0	
Firmware Minor Version	1	0	
Firmware Revision	1	0	

Standard LAM Accessories must set the `I2C DAC` parameter to a non-zero value. Conversely, Advanced LAM Accessories must set the `I2C DAC` parameter to 0.

Standard LAM Accessories must set the following firmware version parameters:

- `Firmware Major Version` = 1
- `Firmware Minor Version` = 0
- `Firmware Revision` = 0

Changes to the UART Baud Rate will only take effect after the Lightning Audio Module has been power cycled. Accessories must receive a lamReadyPacket from the Lightning Audio Module before cycling power.

Actual data transfer rates will vary based on the packet size and configured baud rate. Generally, higher baud rates and larger packets are recommended for best results.

4.12.3.2 accNameInformation (0x04)

Table 4-7 Lightning Audio Module accNameInformation packet payload

Field	Bytes	Factory Default	Description
Name	1-n	null	UTF-8 string

The accessory must provide a value for the Name parameter that matches the accessory's markings and packaging.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

4.12.3.3 accManufacturerInformation (0x05)

Table 4-8 Lightning Audio Module accManufacturerInformation packet payload

Field	Bytes	Factory Default	Description
Manufacturer	1-n	null	UTF-8 string

The accessory must provide a value for the Manufacturer parameter that matches the accessory's markings and packaging.

4.12.3.4 accModelInformation (0x06)

Table 4-9 Lightning Audio Module accModelInformation packet payload

Field	Bytes	Factory Default	Description
Model	1-n	null	UTF-8 string

The accessory must provide a value for the Model parameter that matches the accessory's markings and packaging.

4.12.3.5 accSerialNumberInformation (0x07)

Table 4-10 Lightning Audio Module accSerialNumberInformation packet payload

Field	Bytes	Factory Default	Description
SerialNumber	1-n	null	UTF-8 string

The accessory must provide a value for the SerialNumber parameter that matches the accessory's markings and packaging.

> **Note:** The SerialNumber parameter must be unique for each accessory (i.e. serialized).

Copyright © 2014 Apple Inc. All Rights Reserved.

4.12.3.6 accPreferredAppBundleIdentifierInformation (0x08)

Table 4-11 Lightning Audio Module accPreferredAppBundleIdentifierInformation packet payload

Field	Bytes	Factory Default	Description
PreferredAppBundleIdentifier	1-n	null	UTF-8 string

If the accessory has a preferred app that communicates with the accessory using the External Accessory Protocol, it must set the PreferredAppBundleIdentifier parameter accordingly. Otherwise, the parameter must be set to a null string.

4.12.3.7 accExternalAccessoryProtocolNameInformation (0x09)

Table 4-12 Lightning Audio Module accExternalAccessoryProtocolNameInformation packet payload

Field	Bytes	Factory Default	Description
ExternalAccessoryProtocolName	1-n	null	UTF-8 string

If the accessory has a preferred app that communicates with the accessory using the External Accessory Protocol, it must set the ExternalAccessoryProtocolName parameter accordingly. Otherwise, the parameter must be set to a null string.

4.12.3.8 accHIDComponentInformation (0x0A)

Table 4-13 Lightning Audio Module accHIDComponentInformationInformation packet payload

Field	Bytes	Factory Default	Description
HID Component Function	1	0	Must be set to 8
HID Report Descriptor	2-n	null	

If the accessory intends to send accHIDReport packets (see accHIDReport (0x21) (page 107)), then it must set its HID component information using accHIDComponentInformation.

The HID Component Function parameter must be set to 8 during factory configuration.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

4.12.3.9 accAudioTerminalInformation (0x0B)

Table 4-14 Lightning Audio Module accAudioTerminalInformation packet payload

Field	Bytes	Description
Audio Output Terminal Type	1	See Table 4-15 (page 103)
Audio Output Terminal Latency	2	Number of audio samples as uint16 (Byte 0 = bits[15:8], Byte 1 = bits[7:0])
Audio Output Terminal Gain/Attenuation Step Size	1	See Table 4-17 (page 103)
Audio Output Terminal Gain Steps	1	Number of steps - 1 (0 = 1 step, 255 = 256 steps)
Audio Output Terminal Attenuation Steps	1	Number of steps - 1 (0 = 1 step, 255 = 256 steps)
Audio Input Terminal Type	1	See Table 4-16 (page 103)
Audio Input Terminal Latency	2	Number of audio samples as uint16 (Byte 0 = bits[15:8], Byte 1 = bits[7:0])
Audio Input Terminal Gain/Attenuation Step Size	1	See Table 4-17 (page 103)
Audio Input Terminal Gain Steps	1	Number of steps - 1 (0 = 1 step, 255 = 256 steps)
Audio Input Terminal Attenuation Steps	1	Number of steps - 1 (0 = 1 step, 255 = 256 steps)
Preferred Apple Device Audio Processing	1	See Table 4-18 (page 104)

The accAudioTerminalInformation packet is used to inform the Lightning Audio Module of the accessory's audio input/output terminal configuration.

All other parameters must be set to 0 if the Terminal Type parameter is None.

Accessories must accurately communicate their audio processing latency. Apple devices will use these parameters to help maintain audio/video sync across different terminals for a better user experience.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Accessories must accurately communicate their ability to set gain/attenuation in decibels (dB) using the Step Size and Steps parameters. Apple devices will use these parameters to normalize audio volume settings across different terminals for a better user experience. The total number of gain steps (both Gain and Attenuation) must not exceed 255.

At this time, the Preferred Apple Device Audio Processing parameter must be set to 0.

Table 4-15 Lightning Audio Module Audio Output Terminal Types

Value	Type	Notes
0	None	Must be used if accessory has no ability to render audio from Apple device
1	Headphone	Speaker(s) must be positioned near user's ears
2	Headphone Jack	Integrated 3.5 mm headphone jack
3-255	Reserved	

Table 4-16 Lightning Audio Module Audio Input Terminal Types

Value	Type	Notes
0	None	Must be used if accessory has no ability to record audio to Apple device
1	Headset Microphone	Microphone must be positioned near the user's mouth
2	Headphone Jack	Integrated 3.5 mm headphone jack
3-255	Reserved	

Table 4-17 Lightning Audio Module Audio Terminal Gain/Attenuation Step Size Values

Value	Step Size (dB)
0	0.5
1	1.0
2	1.5
3	2.0
4	2.5

Taiwu@quinergi.com.tw Uploaded by [ID]

Value	Step Size (dB)
5	3.0
6	3.5
7	4.0
8	4.5
9	5.0
10	5.5
11	6.0
12	6.5
13	7.0
14	7.5
15	8.0
16-255	Reserved

Table 4-18 Preferred Apple Device Audio Processing Options

Value	Type	Notes
0	None	Apple device should not perform any audio processing
1-255	Reserved	

4.12.4 Operation Packets

These packets may be sent or received anytime after protocol initialization has occurred.

4.12.4.1 lamInformation (0x10)

Table 4-19 Lightning Audio Module lamInformation packet payload

Field	Bytes	Description
External Accessory Protocol Session	1	0 = Session Closed, 1 = Session Opened

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Field	Bytes	Description
Device Power State	1	See Table 4-20 (page 105)
Device Battery Level	1	0-255, 0 = Empty, 255 = Full
Audio Output Stream Status	1	0 = Not Playing, 1 = Playing
Audio Output Stream Type	1	See Table 4-21 (page 106)
Audio Input Stream Status	1	0 = Not Recording, 1 = Recording
Audio Input Stream Type	1	See Table 4-21 (page 106)

The External Accessory Protocol Session parameter indicates whether an iOS app on the device has opened an EASession to the accessory. Accessories must not send accExternalAccessoryProtocolSessionData packets (see accExternalAccessoryProtocolSessionData (0x20) (page 107)) until a lamInformation packet has been received with this parameter set to 1.

Accessories that both provide power to the Apple device via the Lightning Audio Module and reserve some current to charge their own internal battery using accPassthroughPowerSourceInformation packets (see accPassthroughPowerSourceInformation (0x24) (page 109)) must monitor the Device Power State and Device Battery Level parameters to implement correct charging behavior. The Device Battery Level parameter is only valid when the Device Power State has a value of 0 or 1.

Table 4-20 Lightning Audio Module Device Power State Values

Value	Description
0-1	Device is drawing from internal battery
2-3	Device is connected to external power and not charging its internal battery
4	Device is connected to external power and charging its internal battery
5	Device is connected to external power and its internal battery is fully charged
6-255	Reserved

The Audio Input/Output Stream Status parameters may be used to optimize accessory power consumption. For instance, headphone amplifiers may be placed into a lower power state while the Output Stream Status is Not Playing, although they must remain able to render audio immediately if the status changes to Playing.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Accessories must use the Audio Input/Output Stream Type parameters to trigger content-specific audio processing modes as defined in Table 4-21 (page 106). Most, but not all, stream types are used by both input and output streams. If an accessory receives a stream type of Reserved, it must treat it as if the stream type was set to Default.

Table 4-21 Lightning Audio Module Audio Input/Output Stream Types and Processing Requirements

Value	Name	Input	Output	Accessory Processing Requirements
0	Default	X	X	No requirements, accessory may apply any processing
1	Measurement	X	X	Non-linear and time varying processing must be disabled
2-255	Reserved	X	X	Must be treated the same as Default

4.12.4.2 lamAudioTerminalInformation (0x11)

Table 4-22 Lightning Audio Module lamAudioTerminalInformation packet payload

Field	Bytes	Description
Audio Output Terminal Mute	1	0 = Not Muted, 1 = Muted
Audio Output Terminal Gain Steps	1	Signed byte expressing number of gain steps above zero (positive values) or attenuation steps below zero (negative values).
Audio Input Terminal Mute	1	0 = Not Muted, 1 = Muted
Audio Input Terminal Gain Steps	1	Signed byte expressing number of gain steps above zero (positive values) or attenuation steps below zero (negative values).

If an audio terminal is Muted, the accessory must mute its corresponding audio interface. The accessory may apply a ramp of its choosing while transitioning to a Muted state.

If an audio terminal is Not Muted, the accessory must un-mute the audio interface. The accessory may apply a ramp of a its choosing while transitioning to a Not Muted state.

Confidential Apple Inc. Copyright © 2014 Apple Inc. All Rights Reserved.

Gain values will be set based on information provided by the accessory in an accAudioTerminalInformation packet (see accAudioTerminalInformation (0x0B) (page 102)). Changes in the Gain of an audio terminal while Not Muted must be applied without any ramp by the accessory. The accessory also must not remap or scale the requested Gain value from the Apple device.

The Lightning Audio Module will send a lamAudioTerminalInformation packet to the accessory before starting to stream or record audio for the first time after accessory connect. The accessory must not independently remember and apply its own previous audio terminal volume settings.

4.12.4.3 lamExternalAccessoryProtocolSessionData (0x12)

Table 4-23 Lightning Audio Module lamExternalAccessoryProtocolSessionData packet payload

Field	Bytes	Description
Data	1-n	External Accessory Protocol data

4.12.4.4 lamHIDReport (0x13)

Table 4-24 Lightning Audio Module lamHIDReport packet payload

Field	Bytes	Description
HID Report	1-n	HID report

4.12.4.5 accExternalAccessoryProtocolSessionData (0x20)

Table 4-25 Lightning Audio Module accExternalAccessoryProtocolSessionData packet payload

Field	Bytes	Description
Data	1-n	External Accessory Protocol session data

4.12.4.6 accHIDReport (0x21)

Table 4-26 Lightning Audio Module accHIDReport packet payload

Field	Bytes	Description
HID Report	1-n	HID report

Paul Wu LTD
paul@uniorand.com.tw

4.12.4.7 accRequestAppLaunch (0x22)

No payload.

This packet will request that the Apple device launch the app specified by accPreferredAppBundleIdentifierInformation (see accPreferredAppBundleIdentifierInformation (0x08) (page 101)) on the accessory's behalf. The accessory must not assume that the app has launched, because the request may be denied by iOS or the user. A IamInformation (see IamInformation (0x10) (page 104)) packet must be received indicating that the app has successfully opened an External Accessory Protocol session before the accessory may presume that it is now communicating with the app.

Note: This packet must only be sent in response to direct user action, such as pushing a button on the accessory. Autonomous launch of apps is explicitly prohibited.

4.12.4.8 accAudioTerminalStateInformation (0x23)

Table 4-27 Lightning Audio Module accAudioTerminalStateInformation packet payload

Field	Bytes	Description
Audio Output Terminal State	1	0 = Disabled, 1 = Enabled
Audio Input Terminal State	1	0 = Disabled, 1 = Enabled

If the accessory must accurately report terminal state as Enabled or Disabled to the Lightning Audio Module when initially connected and whenever it changes. Apple devices will use this information to adjust audio routing for the appropriate user experience. The Lightning Audio Module will assume that all accessory audio terminals are Disabled until the accessory sends an accAudioTerminalStateInformation packet indicating otherwise. The accessory should report the Enabled state as early as possible to minimize audio routing time.

Note: Audio terminal state must be reported as Enabled if and only if the terminal is fully capable of rendering or recording audio in a manner that is perceptible to the end user. Failure to comply with this requirement may cause a poor user experience in the form of truncated audio output or input and is grounds for failure to pass self certification.

At this time, the Audio Input Terminal State must always be reported as Disabled. When a future iOS release adds support for audio input via the Lightning Audio Module, an accessory firmware update may be released (pending Apple's approval) that changes this behavior.

Confidential (Apple Internal Only)

4.12.4.9 accPassthroughPowerSourceInformation (0x24)

Table 4-28 Lightning Audio Module accPassthroughPowerSourceInformation packet payload

Field	Bytes	Description
Reserved Current For Accessory	2	Number of milliamps expressed as uint16 (Byte 0 = bits[15:8], Byte 1 = bits[7:0]

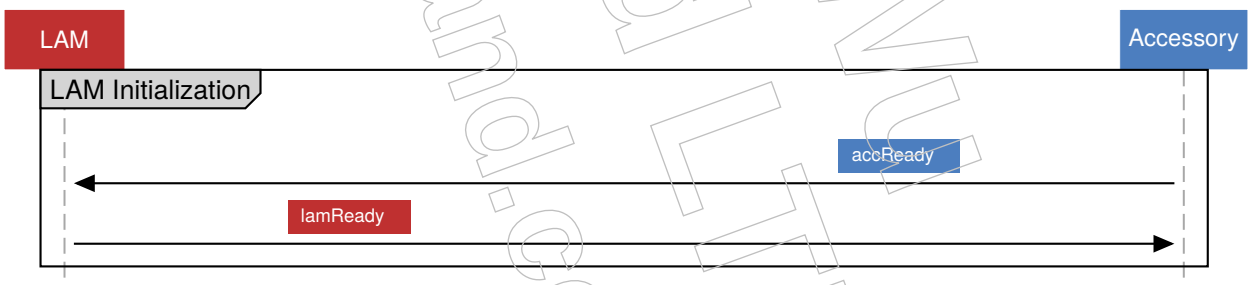
If the accessory has an internal battery, it must use this packet to communicate how much current it will draw from the pass-through power source to recharge the internal battery before passing the rest on to the device. The Apple device will adjust its current draw from the pass-through power source accordingly. This packet must be sent whenever the accessory's current draw from the pass-through power source changes.

The accessory must not reserve more than 500 mA of current until the Apple device's battery is fully charged as indicated by the Device Power State parameter in a lamInformation packet (see lamInformation (0x10) (page 104)).

4.12.5 Examples

4.12.5.1 Initialization

The following example demonstrates how to initialize communications with a Lightning Audio Module.

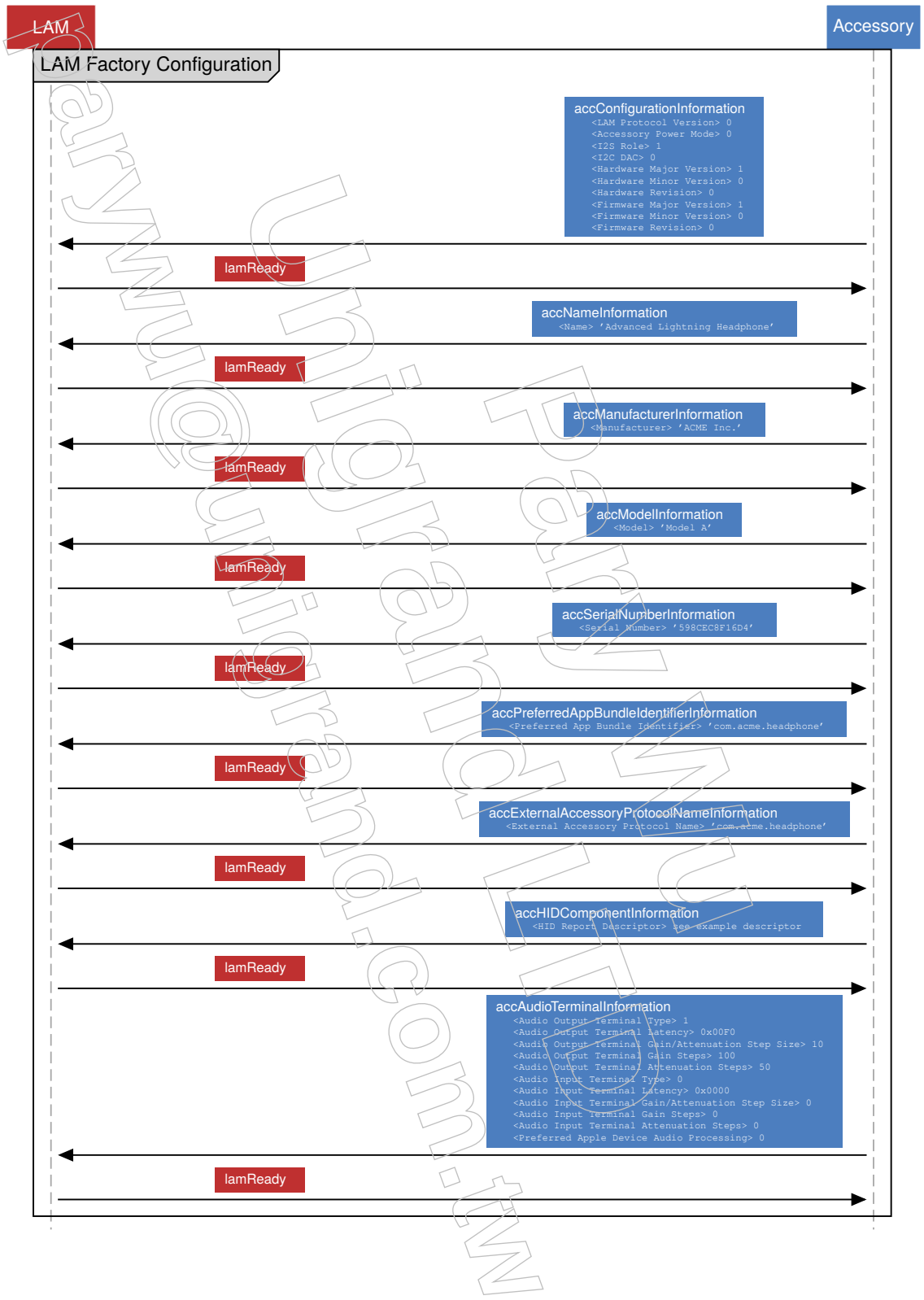


4.12.5.2 Advanced LAM Accessory Factory Configuration

The following example demonstrates how to configure the Lightning Audio Module for use in an Advanced Lightning accessory that:

- Performs active audio processing, but only when audio is streaming from the Apple device
- Communicates with an iOS app that configures EQ settings and downloads firmware updates
- Implements user button controls for Volume Up, Select, and Volume Down

Apple Confidential Property
Unigrand Ltd.
Unigrand.com.tw

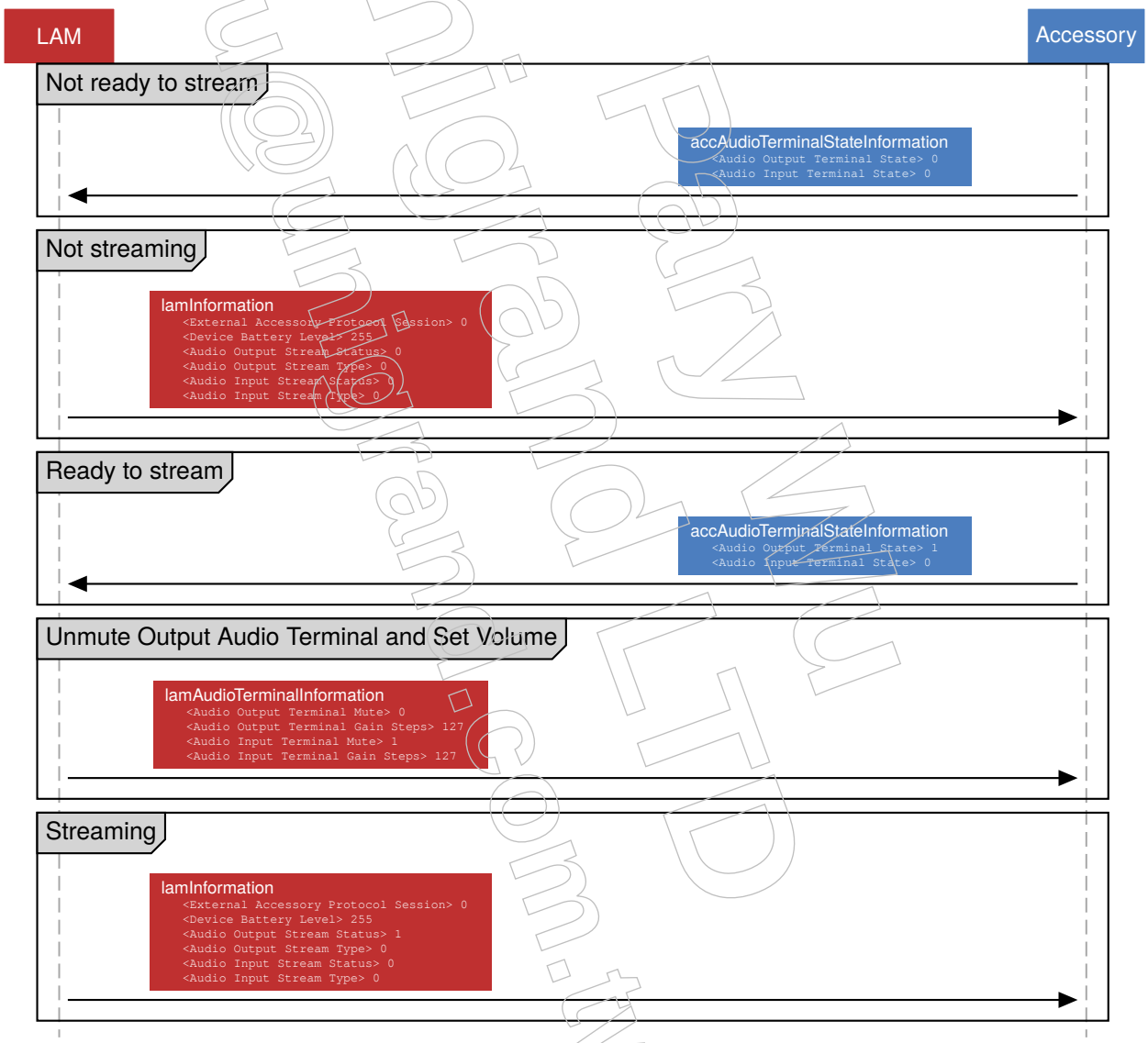


2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

4.12.5.3 Advanced LAM Accessory Typical Usage

The following example demonstrates typical traffic exchanged between the Lightning Audio Module and an Advanced Lightning accessory that:

- Communicates its initial audio terminal information (not ready to stream)
- Receives initial audio terminal information from the Lightning Audio Module (not streaming)
- Communicates that it is ready to accept audio output from the Lightning Audio Module
- Receives notification of incoming audio streaming data from the Lightning Audio Module



Paiyu Technology LTD
sales@paiyutech.com.tw

4.13 Test Procedures

4.13.1 Connectivity

While the accessory is attached to the Apple device, verify that:

1. The accessory is visible in the Settings app and identifies with the correct information strings.
2. The Apple device will enter a sleep state after a period of inactivity while the accessory is still attached.
3. Pressing the Center button on the headphone remote will wake the Apple device from a sleep state.
4. Pressing any other button on the headphone remote will not wake the Apple device from a sleep state.

4.13.2 Audio Quality

Verify that:

1. When the Apple device is not playing music, the accessory's output audio stream is devoid of pops/clicks/artifacts at all times, especially during connection or disconnection.
2. When the Apple device is playing music, the accessory's output audio stream is devoid of pops/clicks/artifacts at all times, especially during connection or disconnection.

4.13.3 Power

Verify that:

1. The accessory complies with all requirements regarding Accessory Power from the Lightning Audio Module (see Accessory Power (page 90)).
2. The accessory remains powered when the Apple device is in a sleep state if the Lightning Audio Module has been configured to provide constant power.

4.13.4 Charge/Sync

If the accessory supports pass-through charge/sync of the Apple device from an external USB host or power supply, verify that:

1. The Apple device charges when the accessory is connected to an external USB power supply, such as an Apple USB power adapter.
2. The Apple device is recognized and can sync/restore when the accessory is connected via USB to a host computer.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

3. If the accessory has an internal battery that can also recharge from an external USB power supply, the accessory's internal battery does not draw more than 500 mA until the Apple device's battery is fully charged.

4.13.5 Firmware Update

Verify that the accessory's firmware can be updated by an end user from an Apple product (running iOS or OS X).

Pary Wu
Unigrand LTD
pary.wu@unigrand.com.tw

5. Apple Lightning Connector

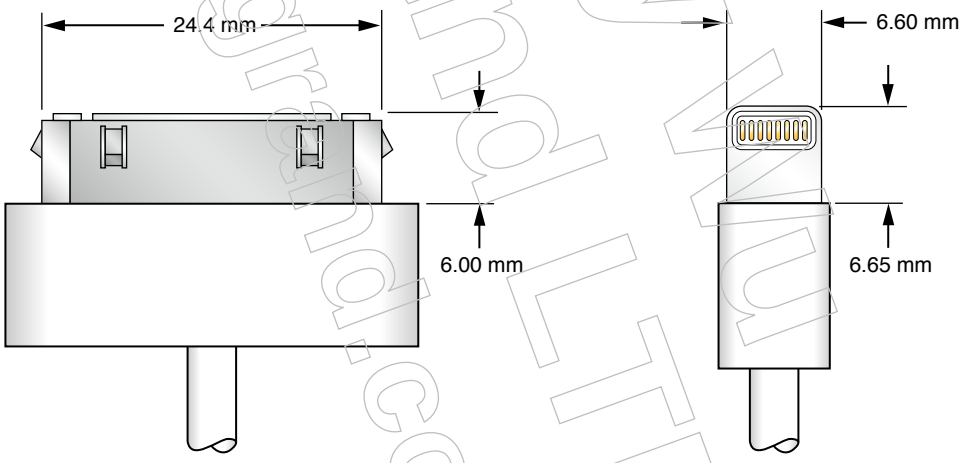
Accessories may incorporate a Lightning connector to establish mechanical and electrical connections with an Apple device and enable other accessory interface features.

This chapter covers the Lightning connector. See *MFi Accessory Hardware Specification R9* for details of the 30-pin.

5.1 Comparison with the 30-pin Connector

5.1.1 Dimensions

Figure 5-1 Comparison of 30-pin connector and Lightning connector dimensions



5.1.2 Active Component

Unlike the 30-pin connector, which is a passive component, the Lightning connector is an active component. Accessories that integrate a Lightning connector must treat it accordingly during both manufacture and field deployment.

Partaunigrand LTD
Partaunigrand.com.tw

5.1.3 Reversability

The Lightning connector engages with Apple devices in either orientation. Accessories do not need to manage this capability. The connector pad/pin configuration always remains constant from the accessory's point of view, because the connector automatically routes power and data to and from the accessory appropriately after determining its insertion orientation.

5.1.4 Accessory Identify

Accessories do not need to provide an Accessory Identify resistor to indicate choice of iAP transport. The Lightning connector contains all needed configuration information internally and handles interface setup with the Apple device.

5.1.5 Accessory Detect

Accessories do not need to manage the state of an Accessory Detect pin. The Lightning connector handles this task.

5.1.6 Apple Device Detect

There is no Apple Device Detect (pin 30 on the 30-pin connector) equivalent. See Apple Device Detection (page 48) for information on detecting the presence of an Apple device.

5.1.7 Analog Audio

There is no provision for analog audio output via the Lightning connector. See Digital Audio (page 366) for information on digital audio input and output.

5.1.8 Analog Video

There is no provision for analog video output.

5.1.9 DisplayPort Video

There is no provision for DisplayPort video output.

5.2 Connector Versions

There are 5 versions of the Lightning connector:

- Lightning (C10) and Lightning (C11) connectors have 9 pads on both sides of an exposed PCB.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- Lightning (C12) has 9 pins located at the end of the connector.
- Lightning (C48) has 4 pads on one side of an exposed PCB.
- Lightning (C68) has 9 pads on both sides of an exposed PCB.

All references to the Lightning connector in this chapter are applicable to all versions unless explicitly specified otherwise.

Accessory developers must take the following recommendations and requirements into consideration when deciding which version to use for a particular accessory design:

Table 5-1 Lightning connector modules

Module	Cable	Dock	Dongle	Form-Fitting
Lightning (C10)	Optional	Prohibited	Prohibited (see note)	Optional
Lightning (C11)	Prohibited	Recommended	Optional	Optional
Lightning (C12)	Prohibited	Prohibited	Optional	Recommended
Lightning (C48)	Optional	Optional	Optional	Optional
Lightning (C68)	Optional	Optional	Optional	Optional

Note: See Dongle Accessories (page 153) for one exception to the prohibition on use of Lightning (C10) in accessory dongle integrations.

Tony Wu
Unionchip LTD
tonywu@unionchip.com.tw

Figure 5-2 Lightning (C10)

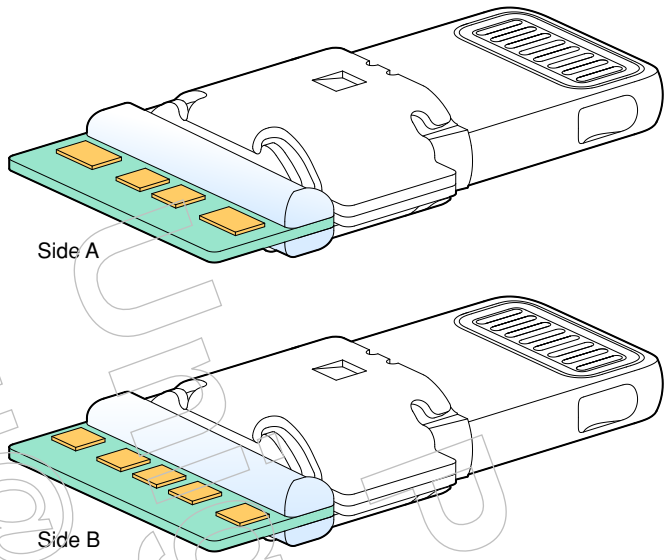
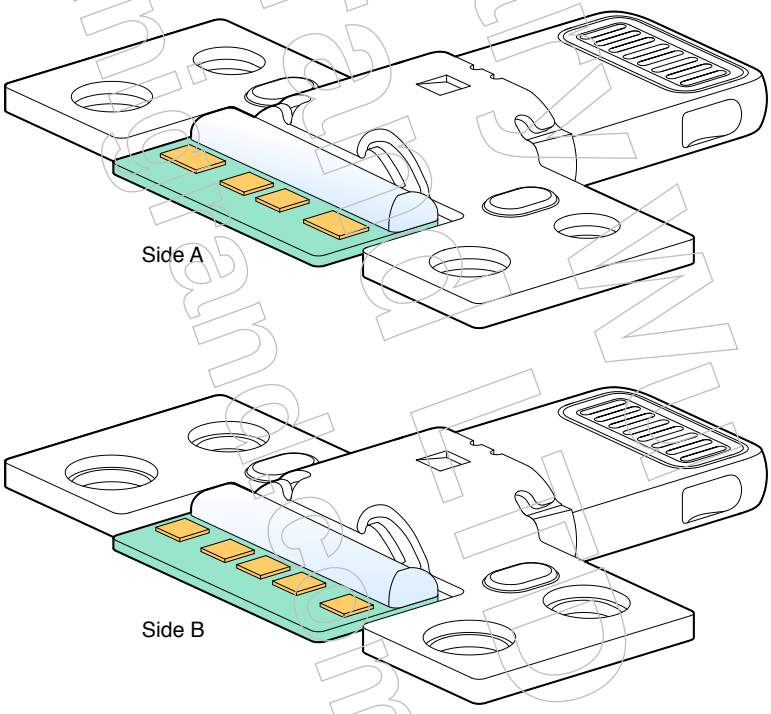


Figure 5-3 Lightning (C11)



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 5-4 Lightning (C12)

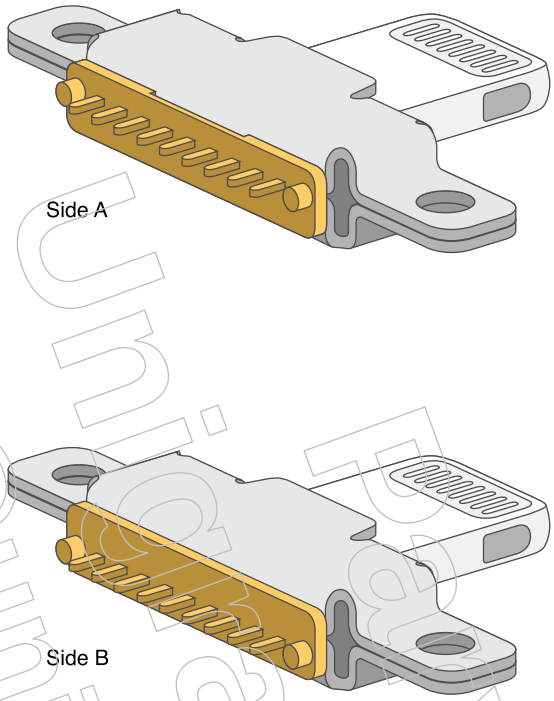
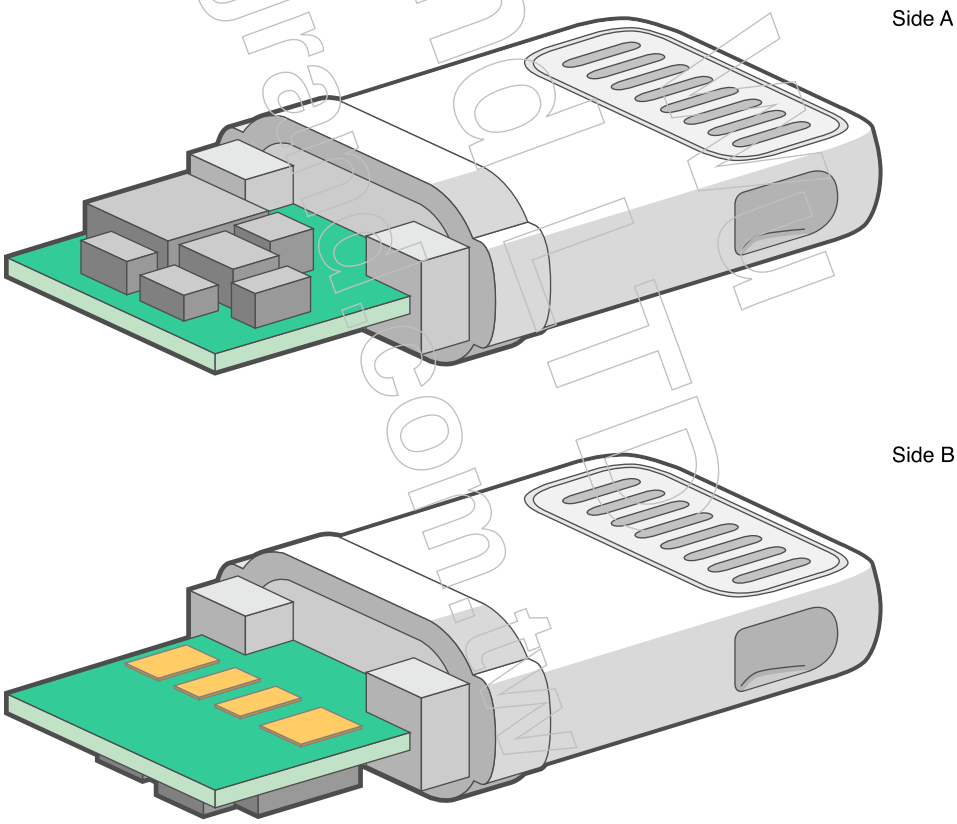
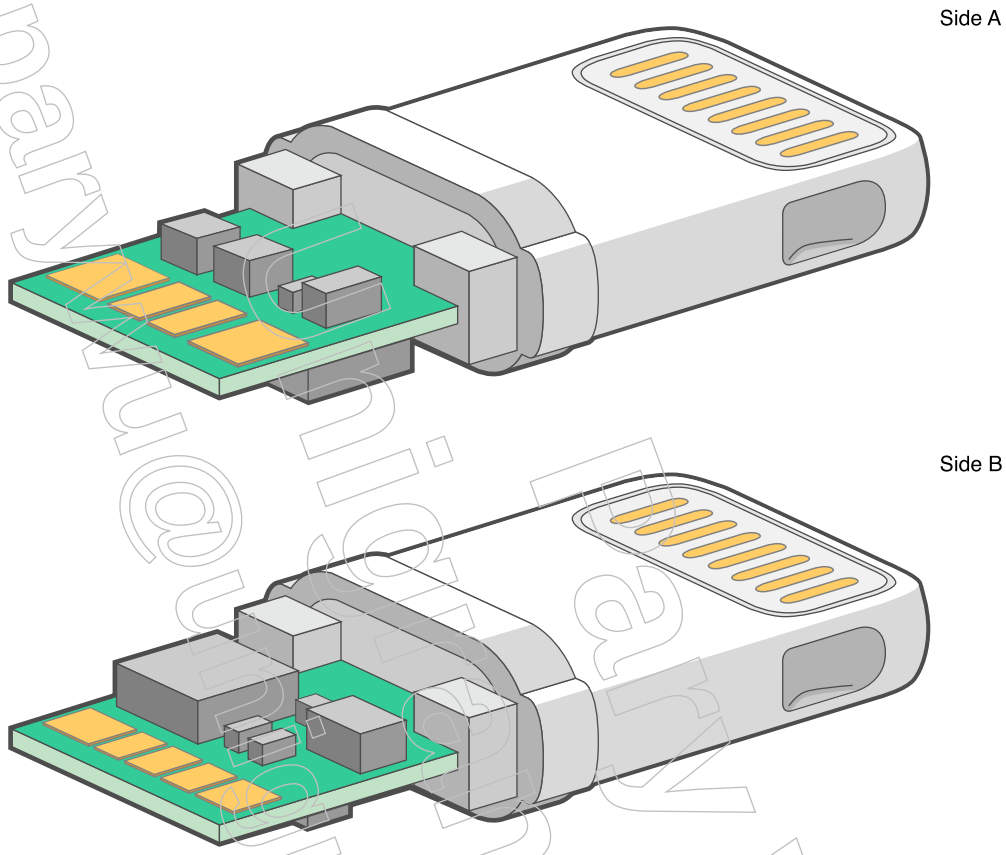


Figure 5-5 Lightning (C48)



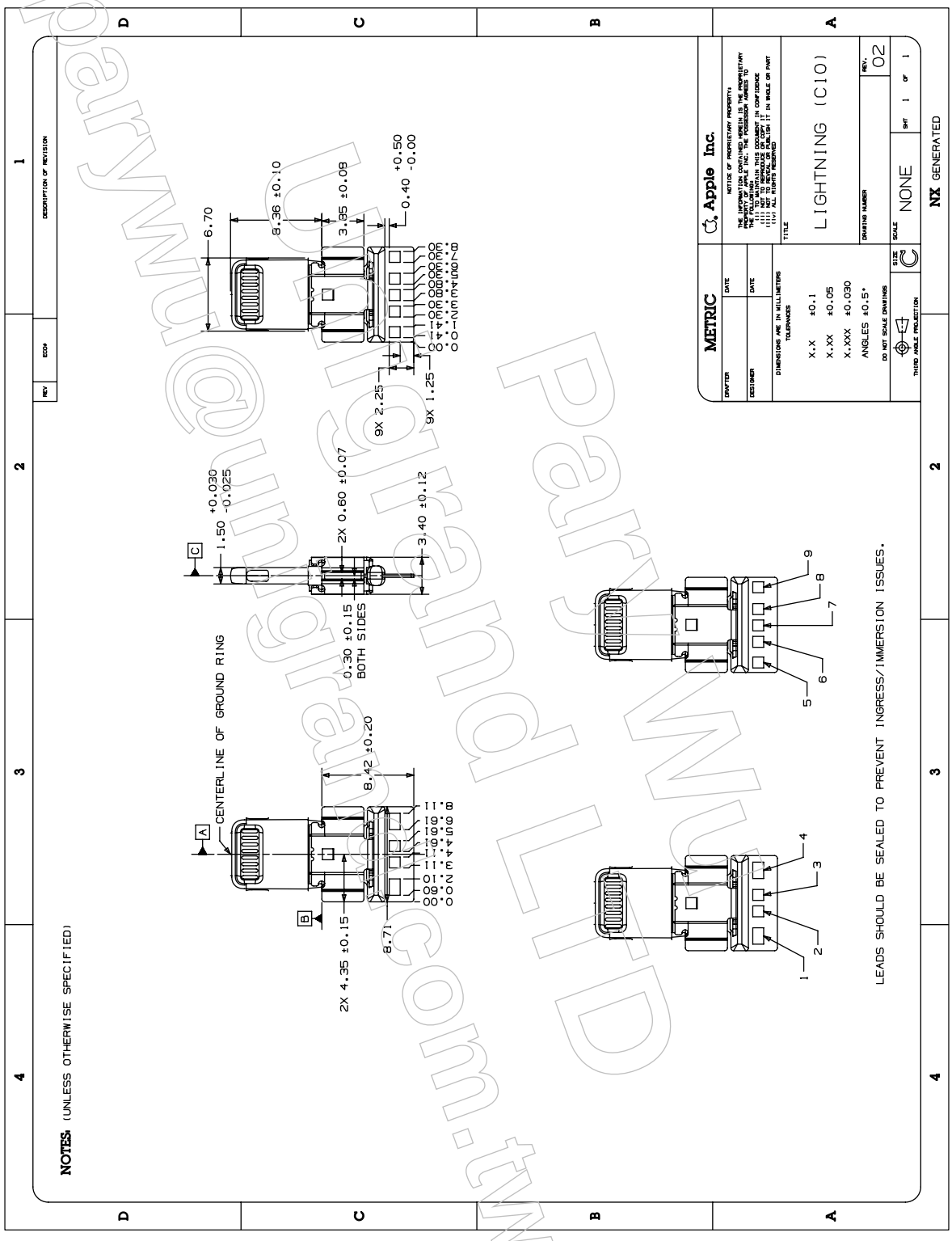
paywu@unigrandtech.com Unigrand Ltd Paywu

Figure 5-6 Lightning (C68)



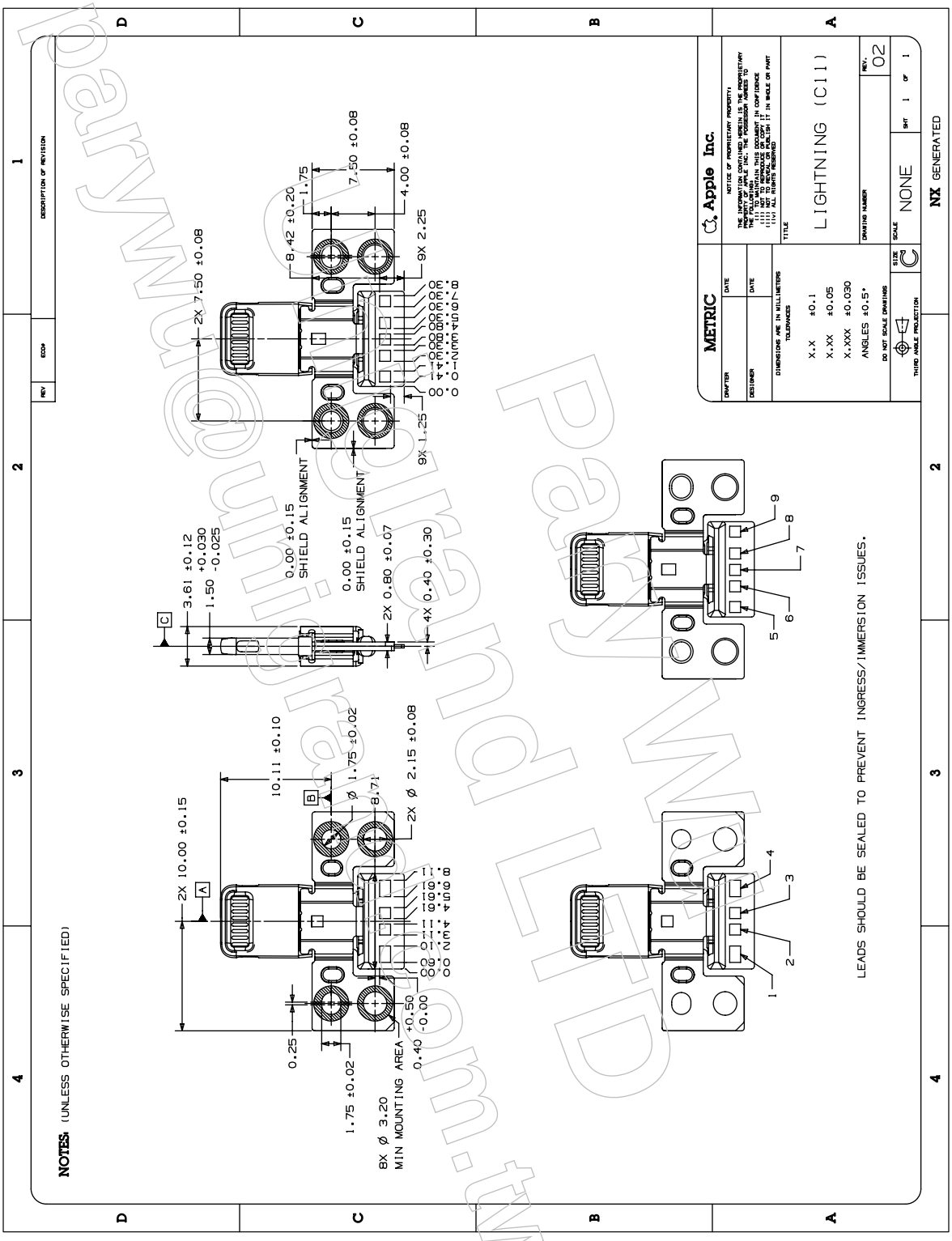
paryvu@outlook.com Pary Vu Pary Vu Brand LTD

Figure 5-7 Lightning (C10) dimensions



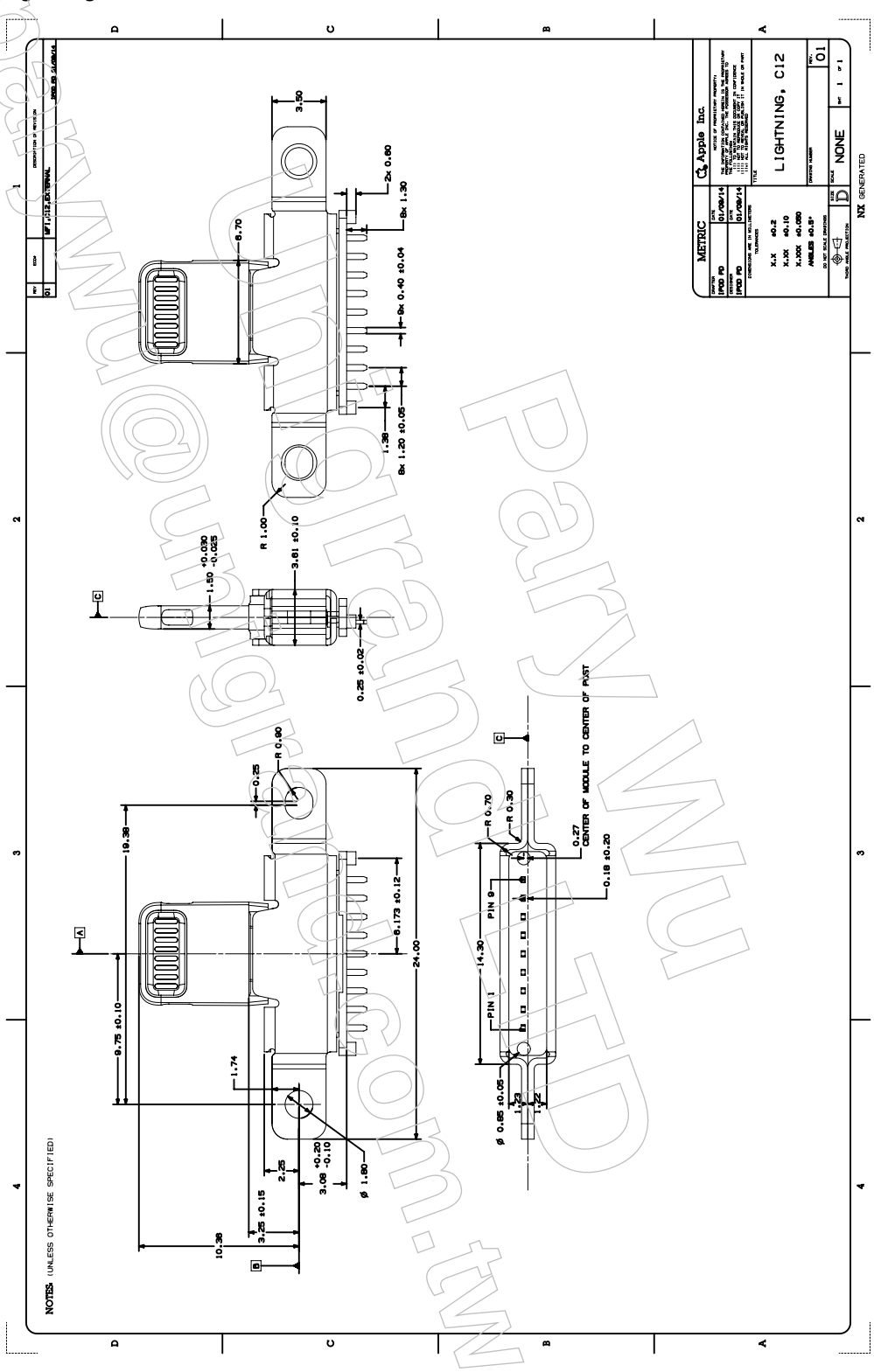
Party Wu Limited
Party@unlimited.com.tw

Figure 5-8 Lightning (C11) dimensions



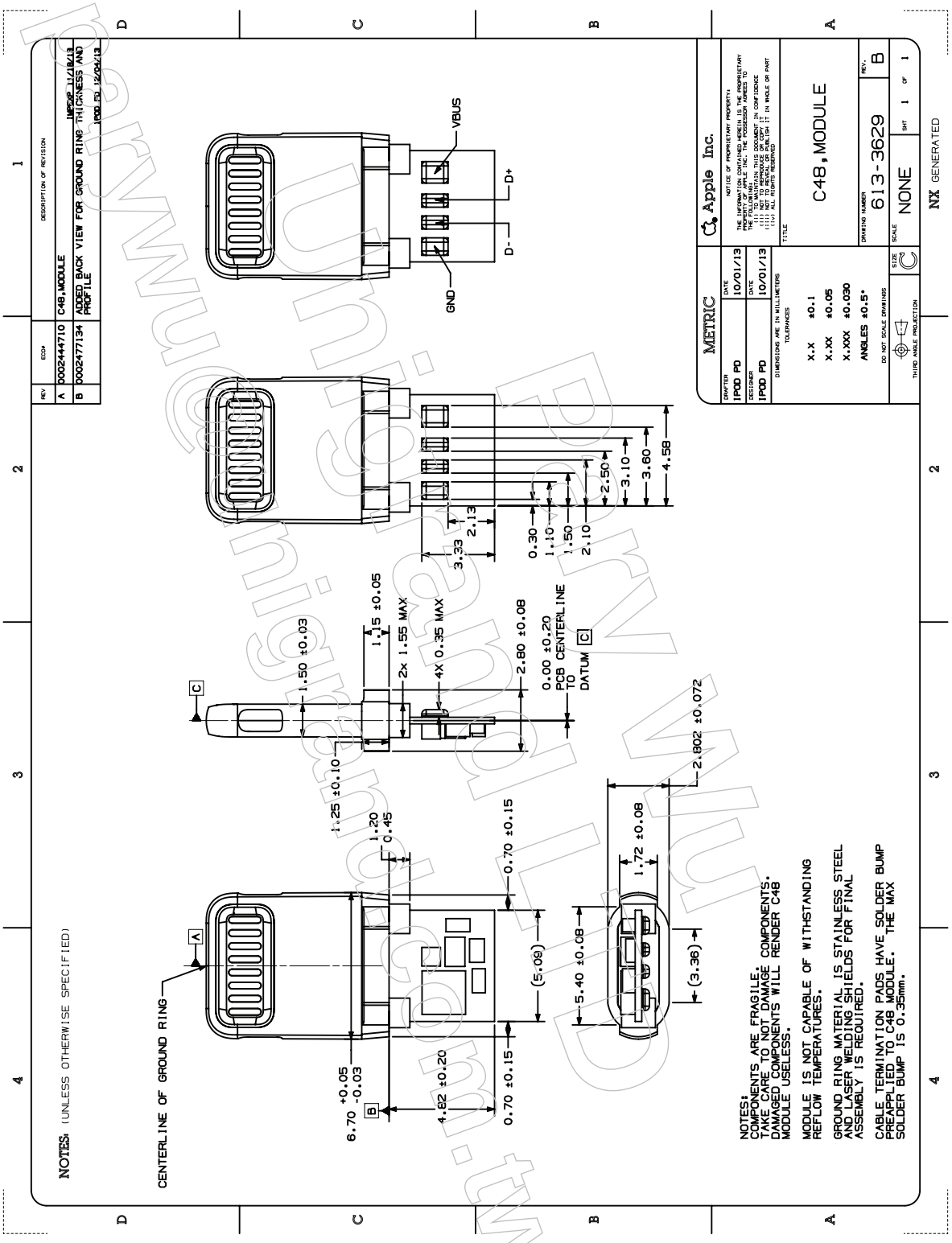
2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 5-9 Lightning (C12) dimensions



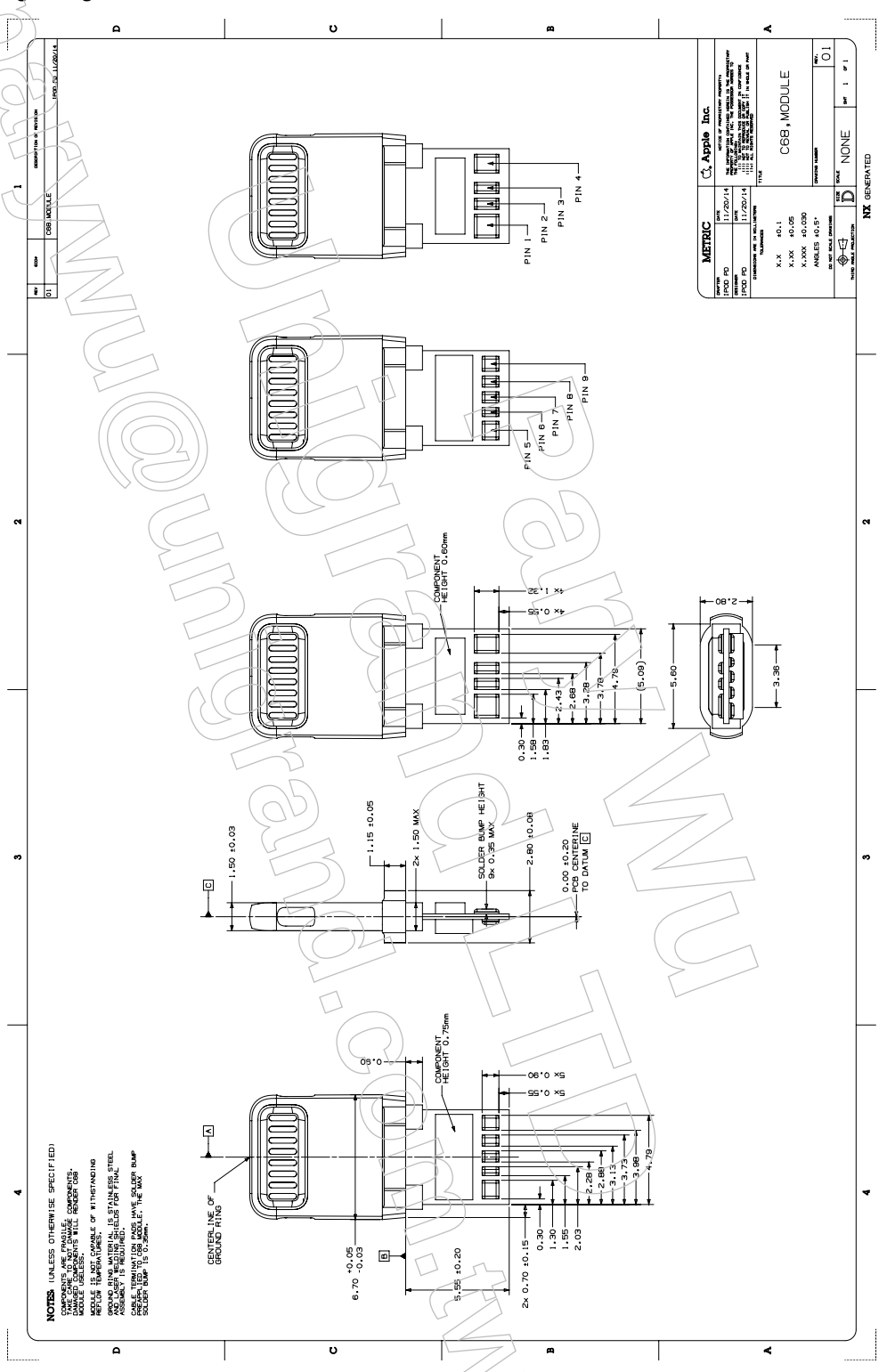
Party Wu
partywu@unionengineering.com.tw

Figure 5-10 Lightning (C48) dimensions



Unofficial Document

Figure 5-11 Lightning (C68) dimensions



Barwu@unigrand.com

5.3 Connector Pad/Pin Configuration

Aside from power and grounds, the pad/pin assignments are not fixed. Instead, the configuration of the other pads/pins is negotiated between the device and the accessory. Accessory developers must choose a configuration and order pre-configured Lightning connectors from Apple for use in final accessory assembly. There are three standard configurations (A/B/C), one for each of the available data transports (USB Host Mode, USB Device Mode, and Serial), and one configuration (E) that is specific to the Lightning Audio Module (see Apple Lightning Audio Module (page 80)).

> **Note:** All accessories that do not have the potential for data communication with the Apple device (Power Only or Power/Sync) accessories must use the B configuration. The D configuration (Power Only) has been deprecated and must not be used in new accessory designs.

More information on the data transports supported by the Lightning connector can be found in the following chapters:

- Serial (see Serial (page 494))
- USB Device Mode (see USB Device Mode (page 522))
- USB Host Mode (see USB Host Mode (page 527))

To refer to a specific connector version and configuration, the connector version is immediately followed by the configuration. For example, Lightning (C10B) refers to a Lightning (C10) connector with the B, or USB Device Mode, configuration.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 5-12 C10/C11 Connector Pads

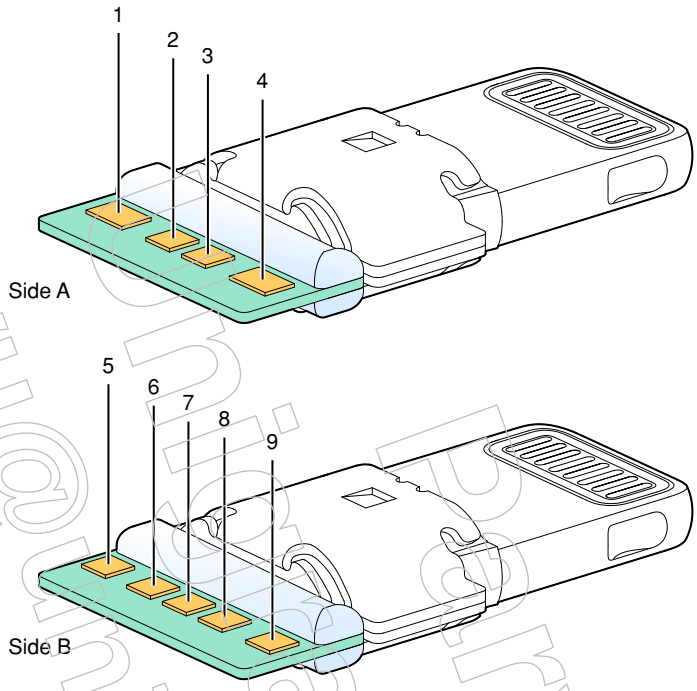
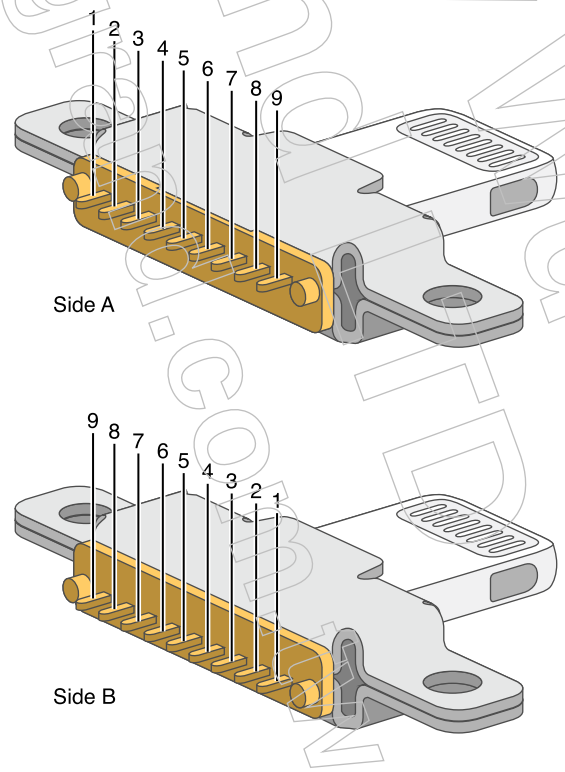
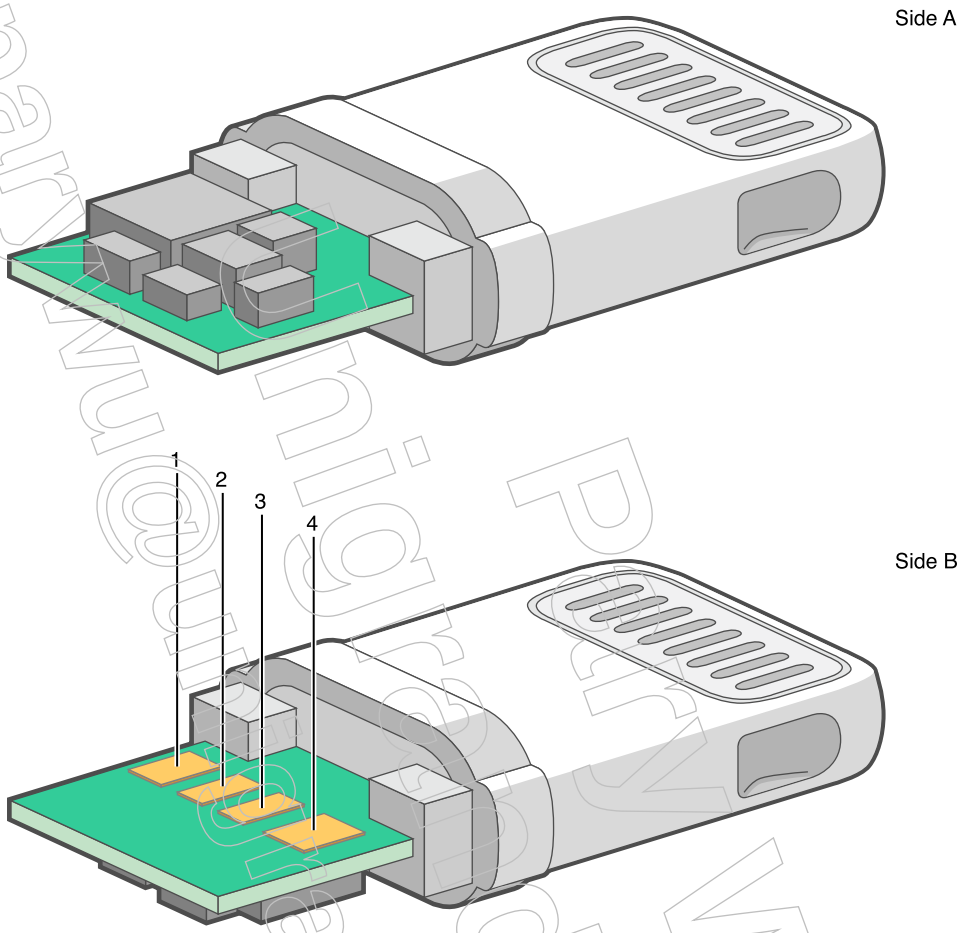


Figure 5-13 C12 Connector Pins



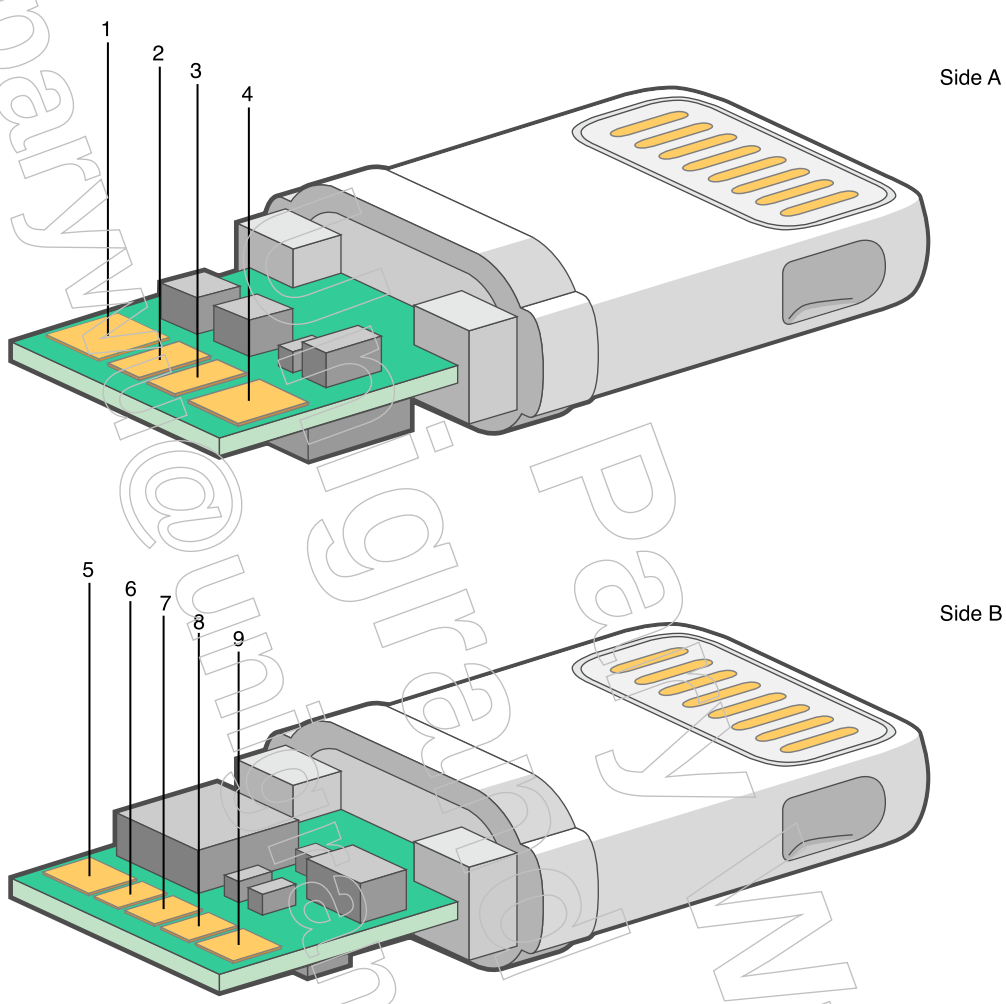
paryvu@uniqueiphone.com

Figure 5-14 C48 Connector Pads



parywu@uniband.com.tw Pary Wu Uniband LTD

Figure 5-15 C68 Connector Pads



Lightning (C10), Lightning (C11), Lightning (C12), and Lightning (C68) connector configurations that exchange data with the Apple device offer accessories the ability to provide power to, and draw power from, the Apple device via the Device Power and Accessory Power pads/pins respectively. Accessories that do not draw power from the Apple device may leave the Accessory Power pad/pin unconnected (NC). This may ease accessory assembly operations.

Lightning (C48) connectors do not allow accessories to draw power from the Apple device, but do allow them to provide power to the Apple device.

Accessories must use the B configuration for a connector (USB Device Mode) if there is no potential for data communication with the Apple device.

Accessories that incorporate a Lightning Audio Module (see Apple Lightning Audio Module (page 80)) must use a Lightning (C10E), Lightning (C11E), Lightning (C12E), or Lightning (C68E) connector. Conversely, accessories that do not incorporate a Lightning Audio Module must not use those connectors.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 5-2 Lightning (C10) connector configurations

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
1	Ground 1	Ground 1	Ground 1	Ground 1
2	USB D-	USB D-	Accessory Serial TX	USB D-
3	USB D+	USB D+	Device Serial TX	USB D+
4	Device Power	Device Power	Device Power	Device Power
5	Accessory Power	Accessory Power	Accessory Power	LAM Power
6	Not Connected	Not Connected	Not Connected	LAM DW
7	Not Connected	Not Connected	Not Connected	LAM D-
8	Not Connected	Not Connected	Not Connected	LAM D+
9	Ground 2	Ground 2	Ground 2	LAM Ground

Table 5-3 Lightning (C11) connector configurations

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
1	Ground 1	Ground 1	Ground 1	Ground 1
2	USB D-	USB D-	Accessory Serial TX	USB D-
3	USB D+	USB D+	Device Serial TX	USB D+
4	Device Power	Device Power	Device Power	Device Power
5	Accessory Power	Accessory Power	Accessory Power	LAM Power
6	Not Connected	Not Connected	Not Connected	LAM DW
7	Not Connected	Not Connected	Not Connected	LAM D-
8	Not Connected	Not Connected	Not Connected	LAM D+
9	Ground 2	Ground 2	Ground 2	LAM Ground

tai@unionlan.com.tw Unionlan Only For TW

Table 5-4 Lightning (C12) connector configurations

Pin	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
1	Not Connected	Not Connected	Not Connected	LAM DW
2	USB D+	USB D+	Device Serial TX	USB D+
3	Ground 1	Ground 1	Ground 1	Ground 1
4	USB D-	USB D-	Accessory Serial TX	USB D-
5	Accessory Power	Accessory Power	Accessory Power	Accessory Power
6	Not Connected	Not Connected	Not Connected	LAM D-
7	Ground 2	Ground 2	Ground 2	Ground 2
8	Not Connected	Not Connected	Not Connected	LAM D+
9	Device Power	Device Power	Device Power	Device Power

Table 5-5 Lightning (C48) connector configurations

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial
1	Ground	Ground	Ground
2	USB D-	USB D-	Accessory Serial TX
3	USB D+	USB D+	Device Serial TX
4	Device Power	Device Power	Device Power

Table 5-6 Lightning (C68) connector configurations

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
1	Ground 1	Ground 1	Ground 1	Ground 1
2	USB D-	USB D-	Accessory Serial TX	USB D-

Confidential Property of Foxconn Interconnect Technology Limited

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
3	USB D+	USB D+	Device Serial TX	USB D+
4	Device Power	Device Power	Device Power	Device Power
5	Accessory Power	Accessory Power	Accessory Power	LAM Power
6	Not Connected	Not Connected	Not Connected	LAM DW
7	Not Connected	Not Connected	Not Connected	LAM D-
8	Not Connected	Not Connected	Not Connected	LAM D+
9	Ground 2	Ground 2	Ground 2	LAM Ground

5.4 Connector Mechanical Requirements

All accessories that integrate the Lightning connector into their design must comply with the mechanical requirements in this section. Different requirements apply, depending on the nature of the integration (e.g. cable, dongle, dock) and the Apple devices with which the accessory is compatible. If an accessory integrates more than one Lightning connector in its design, it must comply with the relevant requirements for each integration. Similarly, if an accessory declares compatibility with more than one Apple device, it must comply with the union of all relevant requirements for each device.

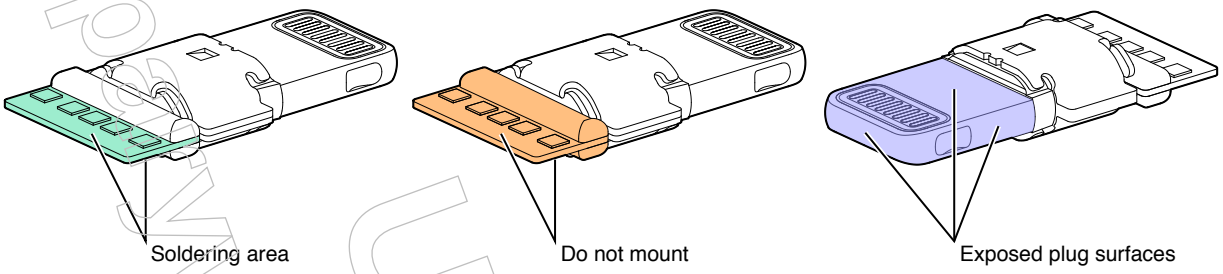
5.4.1 All Accessories

The critical mechanical areas for mounting and connecting a Lightning connector are shown in Figure 5-16 (page 132), Accessory designs must respect these areas when meeting the following requirements:

- The Lightning connector must not be subjected to any process, such as a reflow process, that heats it outside of the soldering area to a temperature above 80 degrees C for more than 30 minutes. Recommended processes include soldering, hot-bar, and over-molding.
- The accessory should not mount or hold on to the Lightning connector's PCB.
- Any process performed on the Lightning connector must not cause any buildup of material on its exposed plug surfaces.

Confidential Daniel Wu daniel_wu@unionchip.com.tw

Figure 5-16 Connector critical areas

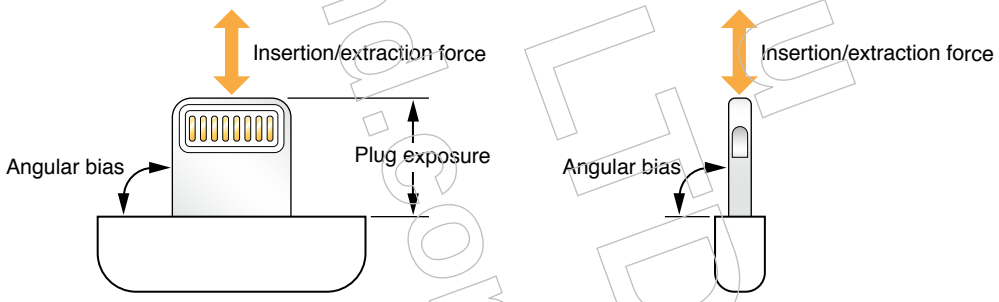


Additionally, all accessory integrations of a Lightning (C11) connector must make physical contact with the connector in the area labeled 'MIN MOUNTING AREA' as shown in Figure 5-8 (page 121).

Figure 5-17 (page 132) illustrates the following requirements related to plug engagement:

- At least 6.47 mm of the connector plug must be exposed. 6.65 mm +0.10 mm/-0.18 mm is recommended.
- When a force of 30 N is applied to the plug in the direction of device insertion, the exposed connector plug length must not decrease below 6.47 mm. The plug is designed to exert a force of 15 N during device insertion.
- The connector plug must not break away from the accessory when a force of 30 N or less is applied on the plug in the direction of device extraction.
- The connector must not be mounted in such a way that the connector plug is biased more than 2 degrees sideways.

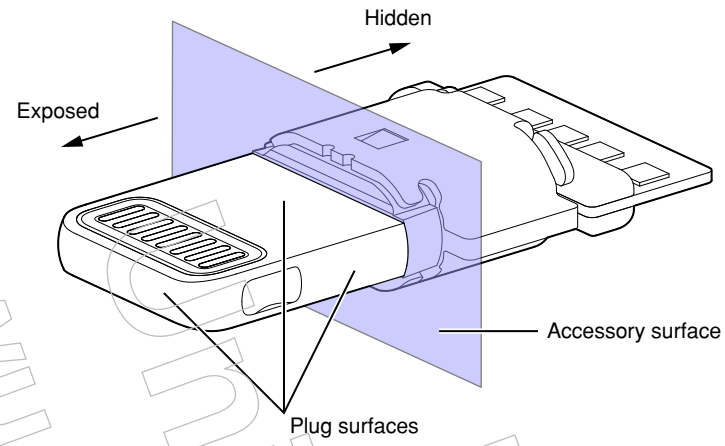
Figure 5-17 Connector plug engagement



Note: All Lightning connector faces besides the cosmetic plug faces should be hidden within the accessory such that the faces are not visible as shown in Figure 5-18 (page 133). Gaps between the accessory surface and the Lightning connector must not exceed 0.25 mm.

Apple Confidential
Uploaded by Wi
Manuals.com

Figure 5-18 Connector plug exposure



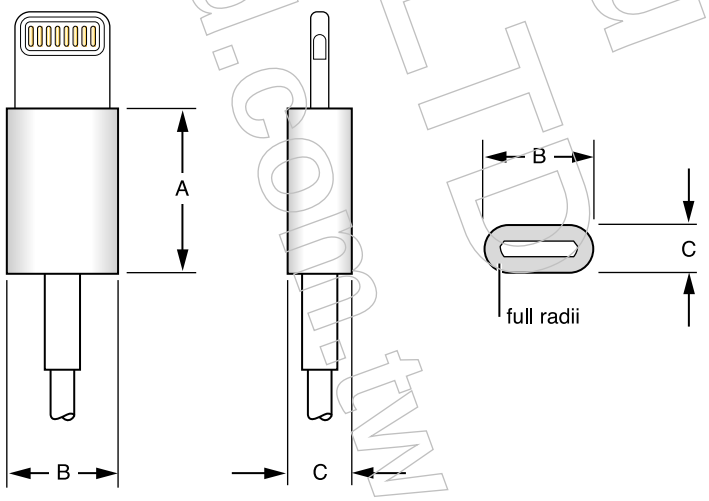
Accessories that use a Lightning (C48) or Lightning (C68) connector must shield the connector. See Shielding (page 137) and Shielding (page 149).

5.4.2 Cable Accessories

A cable is an accessory that integrates a Lightning connector in such a way that the rigid enclosure holding the connector contains only the components necessary to integrate the connector. All other components are held in a separate enclosure that is connected by wire or flex.

An enclosure refers to the rigid portion of an accessory that encloses the components of the accessory. If a cable incorporates a strain relief, the strain relief is not counted in the enclosure. An enclosure must not deform under normal loads.

Figure 5-19 Connector cable enclosure



Party Vu Limited Partnership Ltd
partyvu@outlook.com

To be compatible with cases, the Lightning connector enclosure dimensions must not exceed the following dimensions with full radii rounded edges as shown in Figure 5-19 (page 133):

- (A) 20 mm
- (B) 12.00 mm
- (C) 6.25 mm

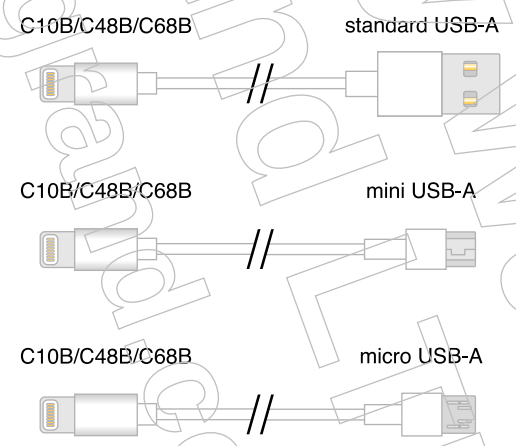
Note: Connector enclosure must contain rounded edges.

5.4.2.1 Cables with USB Connectors

All accessory cables that incorporate both a Lightning connector and a USB-A connector plug (standard, mini, or micro) must use a USB Device Mode Lightning connector configuration.

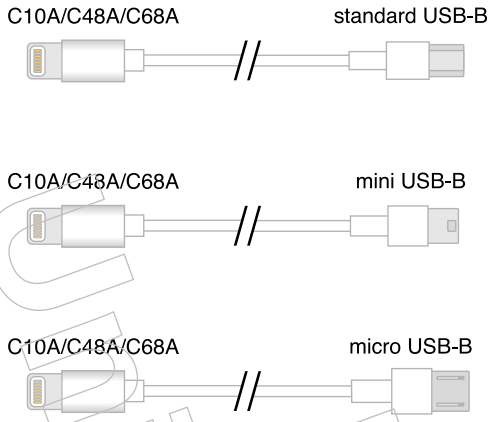
See Figure 5-20 (page 134) and Figure 5-21 (page 135) for all allowed USB to Lightning cable integrations.

Figure 5-20 Allowable USB-A Lightning Cable Integrations



Leawu Wu Unofficial Personal Use Only Unofficial Lab.com.tw

Figure 5-21 Allowable USB-B Lightning Cable Integrations



Accessory cables that incorporate a Lightning connector must not terminate in a USB receptacle.

5.4.2.2 Cables with Non-USB Connectors

Accessory cables that incorporate a Lightning connector must not terminate in a coaxial power connector. Examples of such connectors can be found in the IEC 60130-10, EIAJ, or DIN 45323 standards specifications.

Accessory cables that incorporate a Lightning connector must not terminate in a non-USB receptacle.

Figure 5-22 Example Non-USB Lightning Cable Integration

5.4.2.3 Cable Encapsulation

Additionally, exposed copper of cable/flex and exposed PCB copper should be encapsulated after soldering with electrically nonconductive and liquid sealing compound, for moisture protection. Some suggested compounds that can be used for this purpose are:

- Loctite 3703 UV Encapsulant
- Henkel UV9060 UV Encapsulant
- Cemedine SuperX 8008 Neo Encapsulant (RTV)
- Dymax 29990 UV Encapsulant

A jet dispense process is recommended for applying encapsulation. Some recommended vendors of encapsulation equipment are:

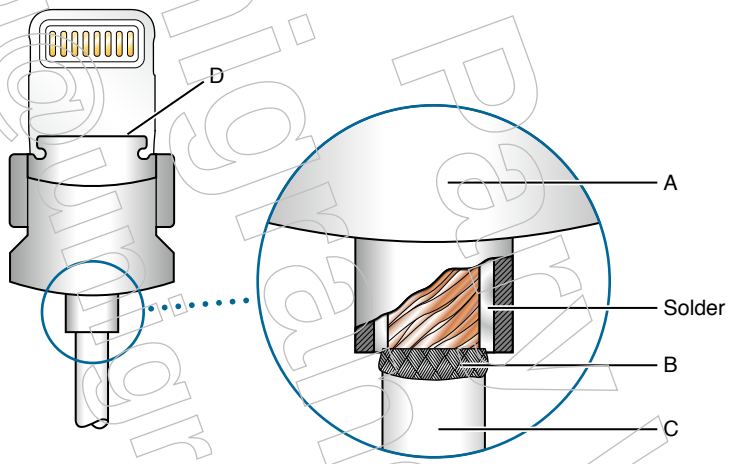
parywu@unionpad.com.tw Unionpad Wu LTD

- Nordson EFD/ASYMTEK (http://www.nordson.com)
- Musashi Engineering (http://www.musashi-engineering.co.jp)

5.4.2.4 Cable EMC Considerations

Figure 5-23 (page 136) illustrates a typical cable accessory integration of the Lightning (C10) connector. Accessory developers are responsible for all components other than Lightning (C10), including the crimp and cable terminations.

Figure 5-23 Cable EMC Diagram



The following recommendations related to EMC apply to all cable accessories, using Figure 5-23 (page 136) as a visual reference:

- The cable (C) should have both braid and foil shield. The braid should have a minimum of 95% coverage.
- Shield braid (B) should make 360 degree termination to the plug housing (A). Copper tape is recommended for this purpose. The termination should include mechanically secured crimp and soldering. 'Pigtail'-type connections should not be used.
- Connection from plug (D) to plug housing should make a 360 degree termination.
- Connector-to-connector resistance should be <100 mΩ maximum end to end.

5.4.2.5 Lightning (C48)/Lightning (C68) Cables

All cables that terminate in a Lightning (C48) or Lightning (C68) connector must additionally comply with the requirements in this section.

All cables that incorporate a Lightning (C48) or Lightning (C68) connector must pass the test procedures specified in Cable Accessories (page 159) and Lightning (C48)/Lightning (C68) Accessories (page 156).

Preliminary Apple Confidential Information

5.4.2.5.1 Cable Termination

Lightning (C48)/Lightning (C68) cable termination must be done using one of the following methods:

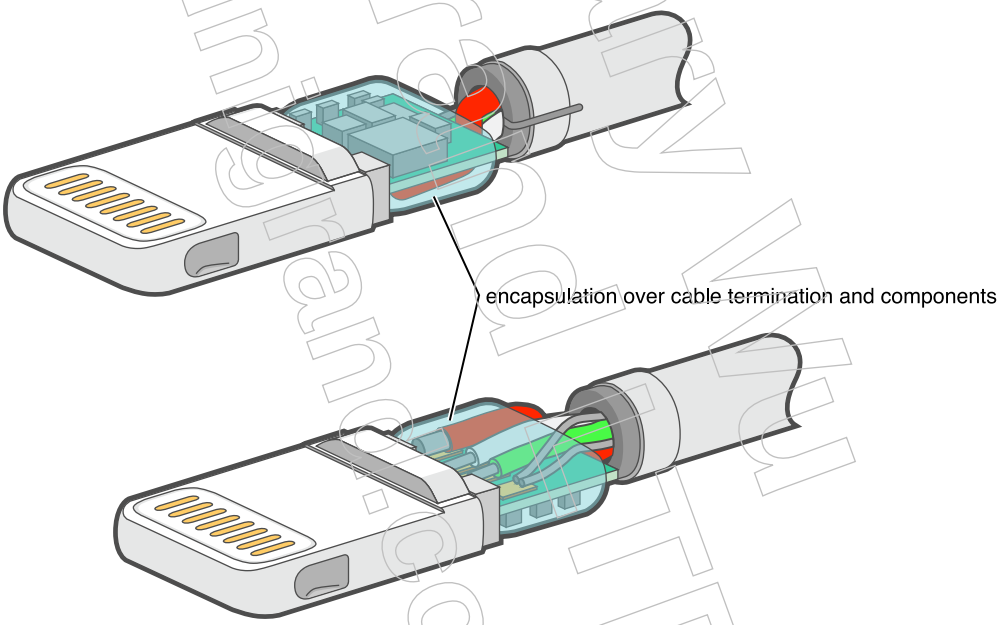
- Hand soldering
- Hot bar soldering
- Laser soldering

Ensure that none of the components of the connector are damaged during termination.

5.4.2.5.2 Encapsulation

Both sides of the Lightning (C48) or Lightning (C68) connector must be encapsulated (see Cable Encapsulation (page 135)) to protect it from liquid ingress and damage. UV cure encapsulation is recommended. Avoid processes that subject the connector to heat over 85° Celsius.

Figure 5-24 Lightning (C48) Cable Encapsulation Diagram

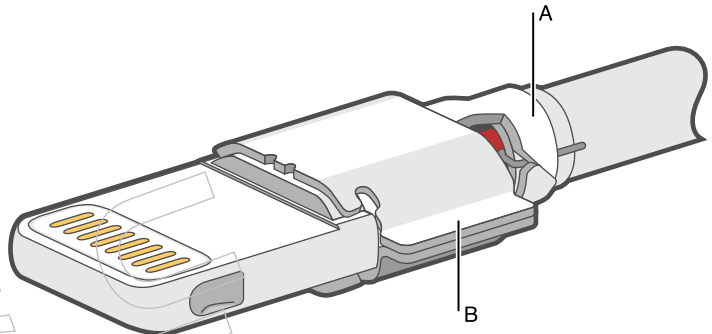


5.4.2.5.3 Shielding

All cables incorporating a Lightning (C48) or Lightning (C68) connector must incorporate a cable shields as shown in Figure 5-25 (page 138). Use of 0.40 mm thick SUS 304 1/2H material is recommended.

Apple Confidential
Unauthorized Copy Not Permitted

Figure 5-25 Lightning (C48) Cable Shielding Diagram



The shield must extend to crimp to the cable shield braid (A) as illustrated in Figure 5-25 (page 138). See Cable EMC Considerations (page 136) for additional information.

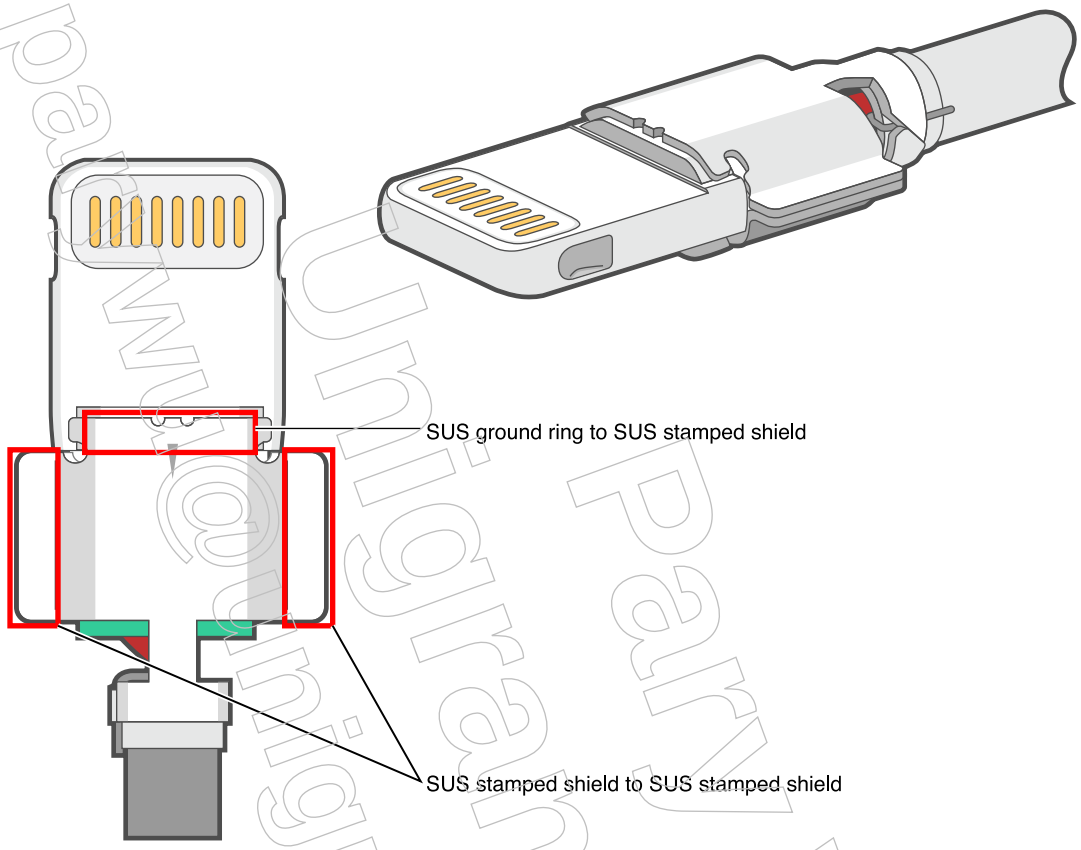
Additionally, flanges (B) are recommended for increased plug strength as illustrated in Figure 5-25 (page 138).

Shields must be free of burrs. Electropolishing is recommended as burrs can cut into wires and cause failures either in the factory or in the field.

The SUS stamped shield must be laser welded to the Lightning (C48) or Lightning (C68) ground ring as shown in Figure 5-26 (page 139). Recommended flange welding areas are also indicated. See Laser Welding (page 144).

partywu@inergrand.com.tw Inergrand LTD Party Wu

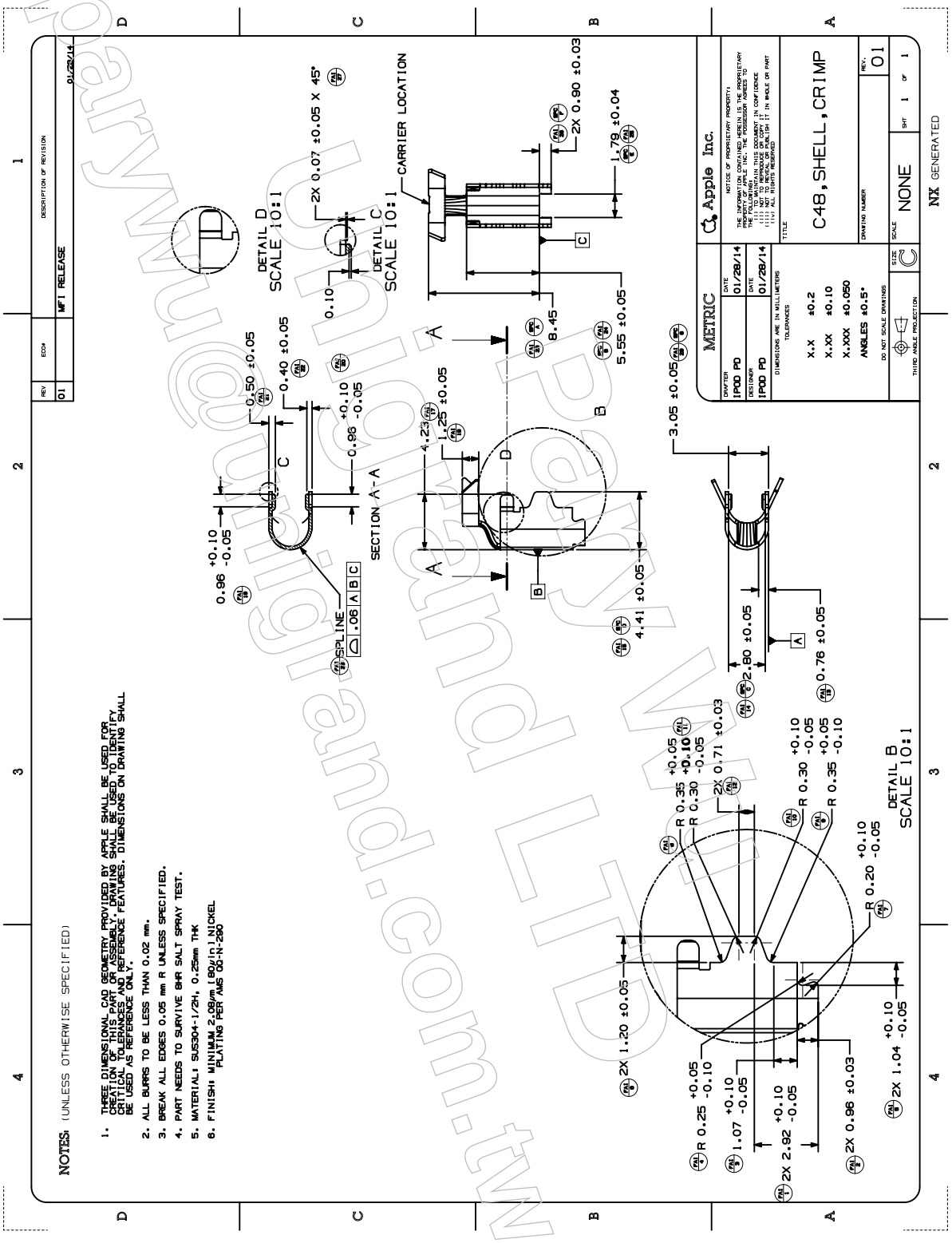
Figure 5-26 Lightning (C48) Cable Shield Laser Welding Diagram



Apple recommends the following two-part cable shield design for Lightning (C48).

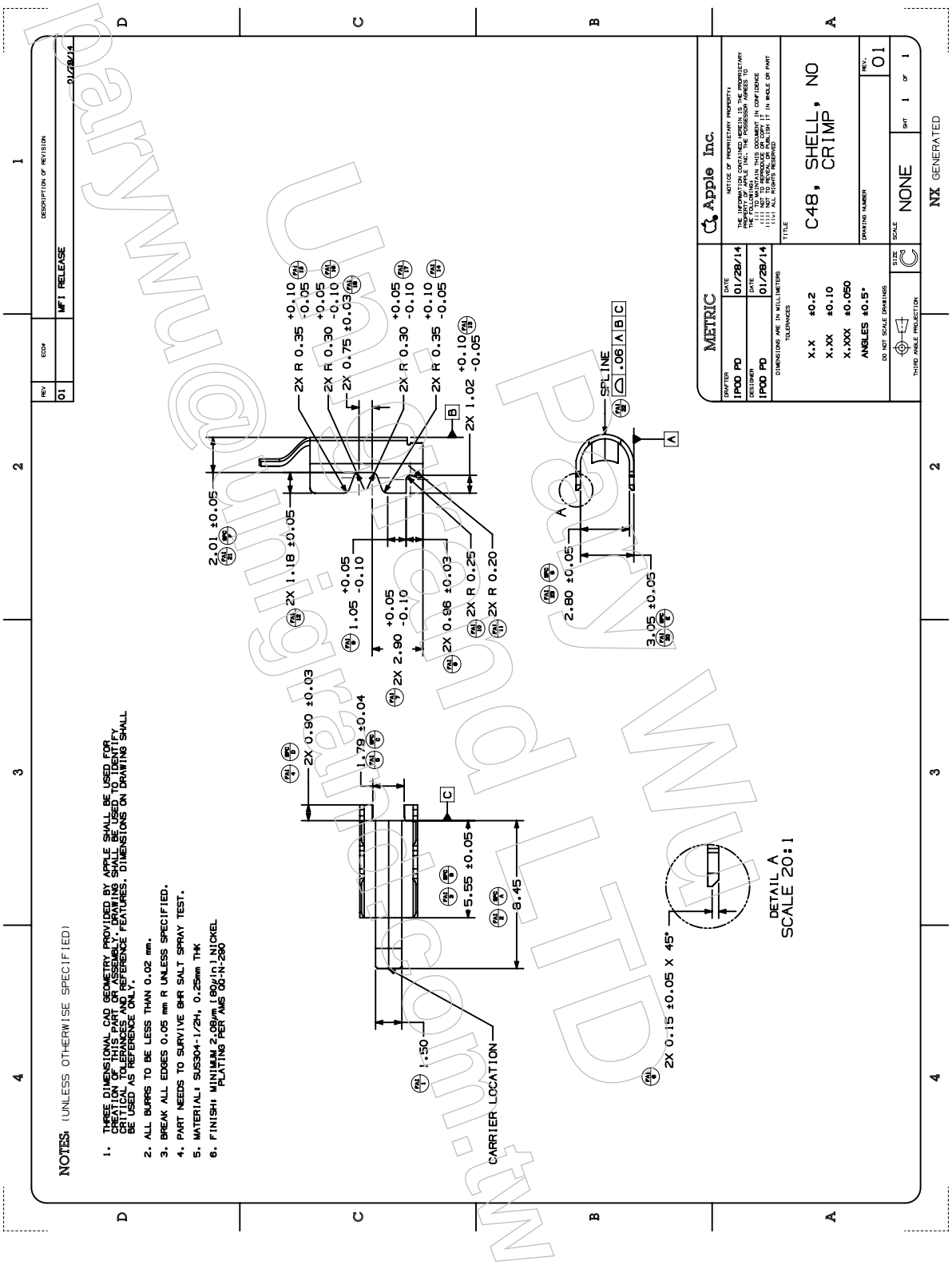
Unigrand LTD Parv Vui
parvui@unigrand.com.tw

Figure 5-27 Lightning (C48) Recommended Cable Shield, Crimp



Copyright Protected and Printed by VeriPart Ltd
http://www.veripart.com.tw

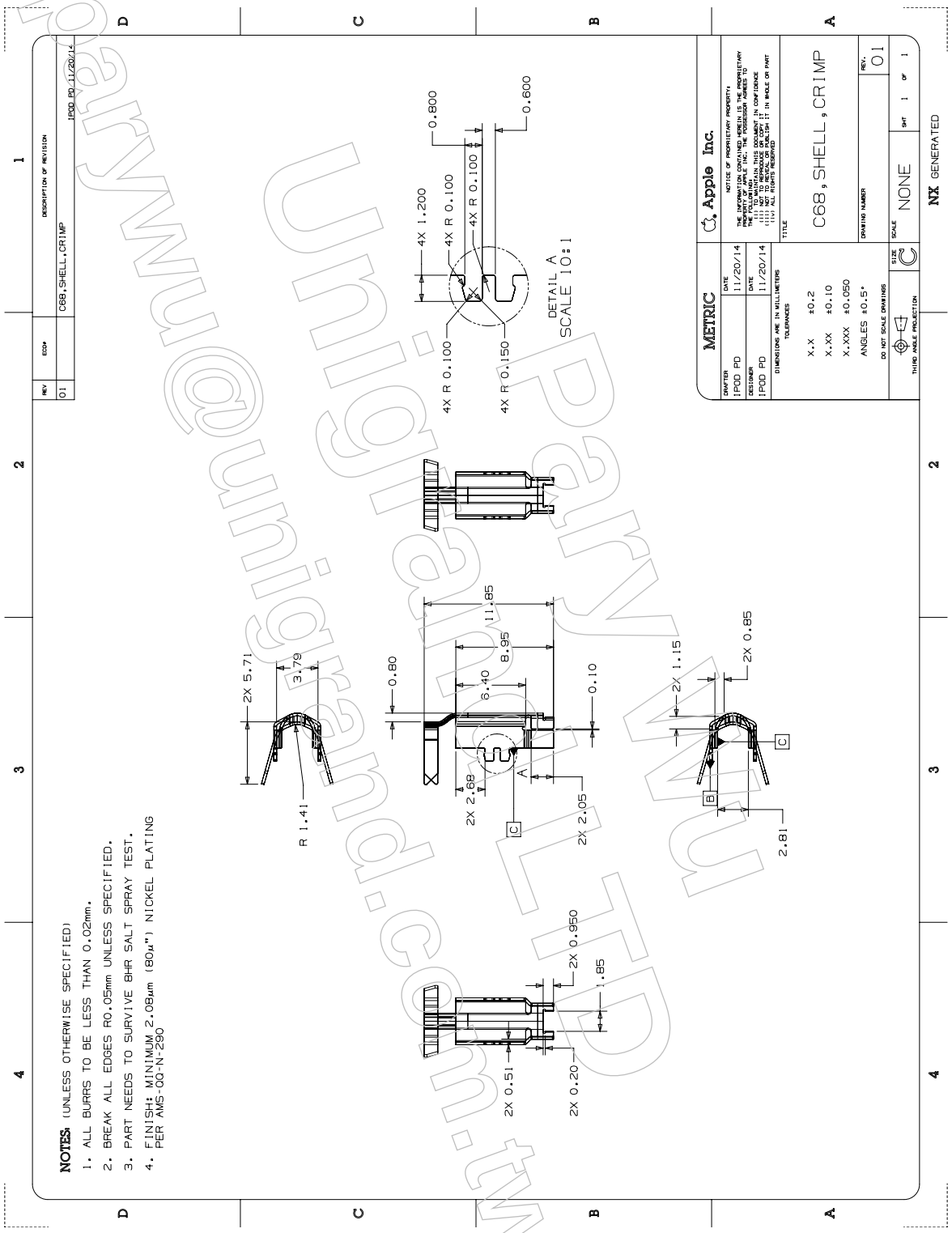
Figure 5-28 Lightning (C48) Recommended Cable Shield, No Crimp



Apple recommends the following two-part cable shield design for Lightning (C68).

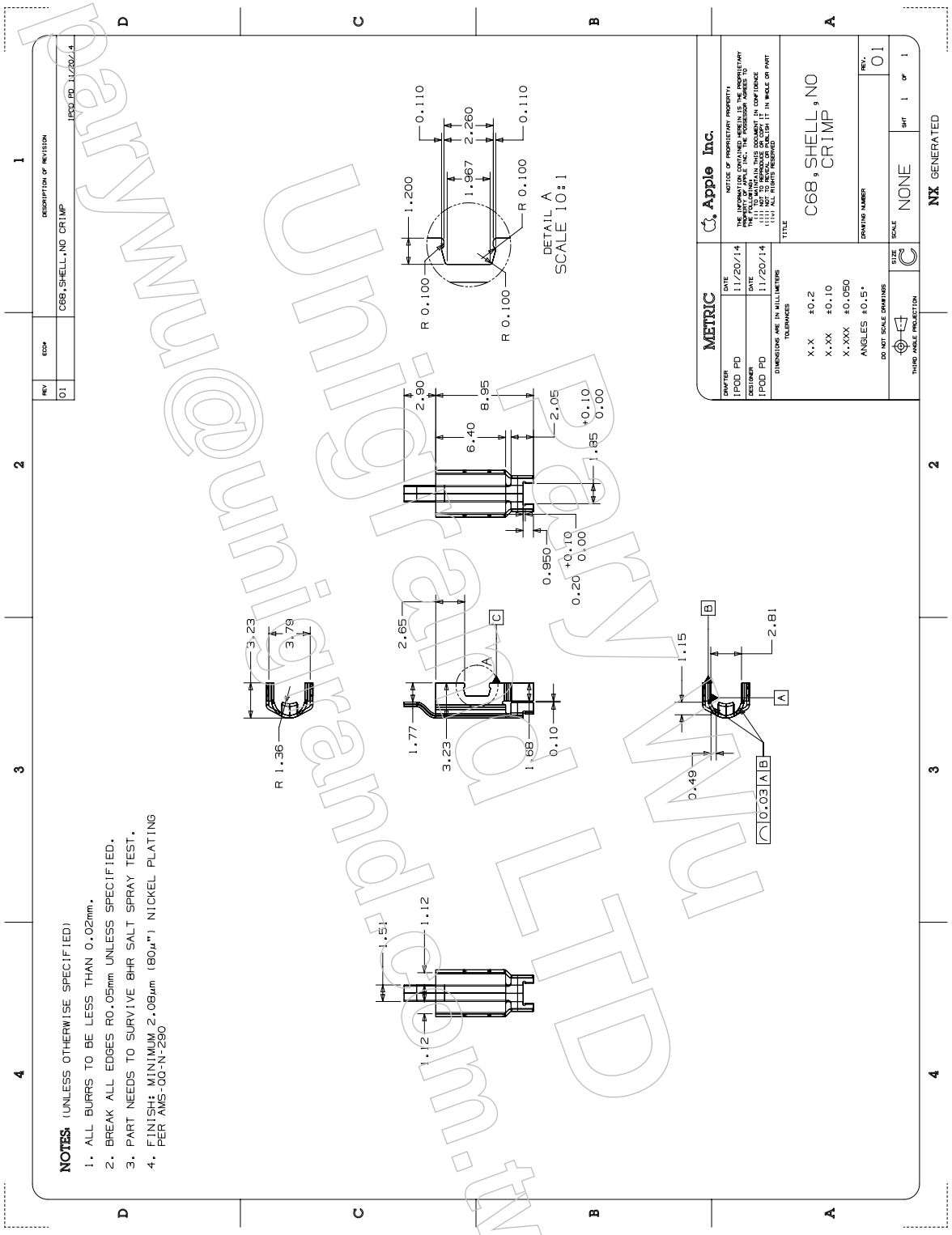
Paraguard Limited Partnership
Unauthorized Use Prohibited

Figure 5-29 Lightning (C68) Recommended Cable Shield, Crimp



Party Wu
Unigiant@unigiant.com.tw

Figure 5-30 Lightning (C68) Recommended Cable Shield, No Crimp



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

5.4.2.5.4 Laser Welding

A SUS stamped shield must be laser welded to the Lightning (C48) or Lightning (C68) ground ring. Apple suggests the following laser welding equipment vendors:

- Han's Laser (http://www.hanslaser.net)
- Lype Laser (http://www.lype-laser.com)

5.4.3 Dock Accessories

A dock is an accessory that is designed to hold the Apple device when it is connected to the Lightning connector. In a dock, the Apple device can be moved relative to the Lightning connector (unlike in a form-fitting accessory, see Form-Fitting Accessories (page 153)).

All dock accessories must use either:

- Lightning (C48) or Lightning (C68) with a shield (see Shielding (page 149))
- Lightning (C11)

One of two approaches must be used to secure the Lightning connector to the dock accessory:

- 4 M1.6 screws are used to secure the Lightning connector to the dock accessory.
- 2 M1.6 screws are used in the bottom holes of the Lightning connector and 2 locating posts are fed through the top holes. Apple recommends that the size and tolerance of the locating posts is 1.68 ± 0.05 mm.

The tip of the Lightning plug is made with a conductive material. Apple recommends that a continuity tester be designed and used during the accessory manufacturing process to assure that the minimum exposed plug length meets the requirement of 6.47 mm.

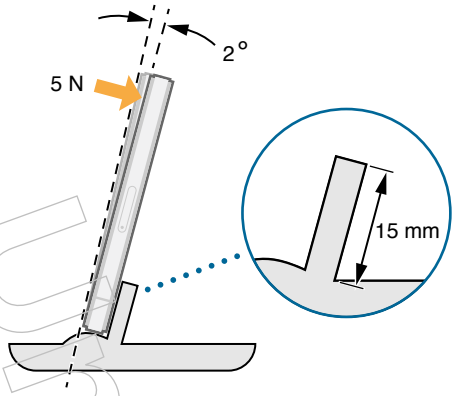
5.4.3.1 iPhone and iPod touch Dock Requirements

Any dock accessory that is intended to be used with an iPhone or iPod touch, or which can reasonably be connected to an iPhone or iPod touch, must comply with the requirements in this section in addition to the general requirements.

The accessory must have a backstop that contacts the iPhone or iPod touch at a point at least 15 mm vertically from the bottom of the device when a force of 5 N is applied perpendicular to the screen of the device at a point 100 mm vertically from the bottom of the device. The iPhone or iPod touch will rotate approximately 2 degrees from its rest position relative to the backstop when this force is applied. A typical backstop is shown in Figure 5-31 (page 145).

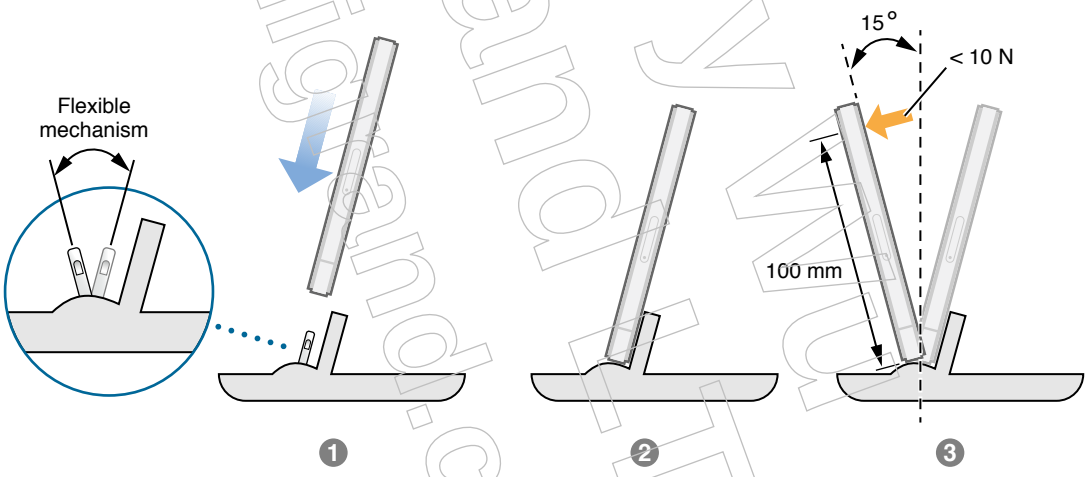
Preliminary Unpublished Draft
Apple Confidential

Figure 5-31 iPhone/iPod touch/iPod nano backstop



The accessory must have a flexible mechanism that allows a user to rotate the iPhone or iPod touch from the rest position to at least 15 degrees from vertical. This mechanism must prevent the user from applying 10 N perpendicular to the iPhone or iPod touch at a distance 100 mm from the bottom of the device before the device reaches the compliant limit.

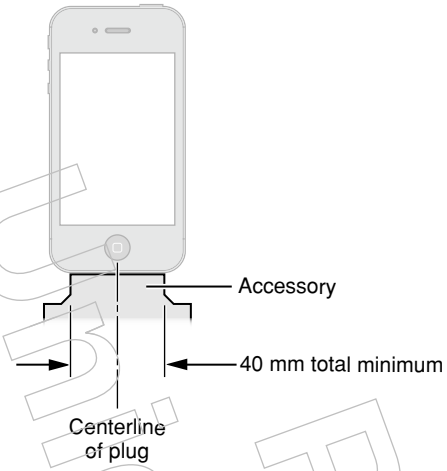
Figure 5-32 iPhone/iPod touch flexible mechanism



The accessory should contact the bottom of the device at two points, each of which is at least 20 mm away from the side of the connector, when the device is in the rest position.

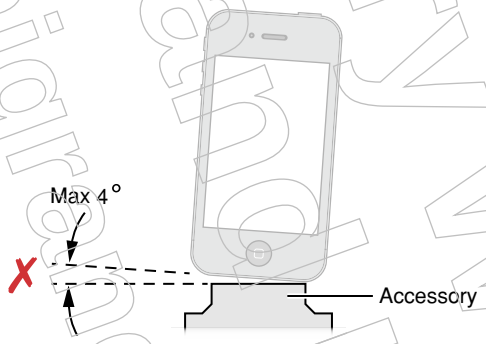
partywu@unionpharmtw.com.tw

Figure 5-33 iPhone/iPod touch contact



The Lightning connector is designed to disconnect if biased more than 4 degrees sideways.

Figure 5-34 iPhone/iPod touch sideways angle



The accessory should not block any iPhone or iPod touch speakers or microphones. See Cases (page 336) for a pointer to dimensional drawings of iPhone and iPod touch that call out locations of speakers and microphone.

5.4.3.2 iPad Dock Requirements

Any dock accessory that is intended to be used with an iPad, or which can reasonably be connected to an iPad, must comply with the requirements in this section in addition to the general requirements.

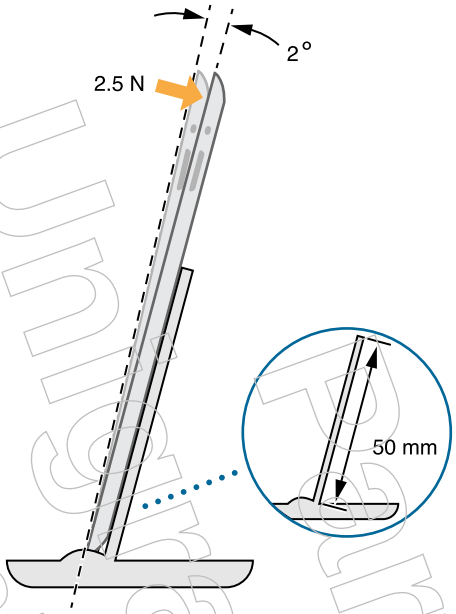
The accessory must have a backstop that contacts the iPad at a point at least 50 mm vertically from the bottom of the device when a force of 2.5 N is applied perpendicular to the screen of the device at a point 200 mm vertically from the bottom of the device.

The accessory should contact the bottom of the device at two points, each of which is at least 50 mm away from the side of the connector, when the device is in the rest position.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

The accessory and its backstop should not block any speakers or microphones.

Figure 5-35 iPad backstop

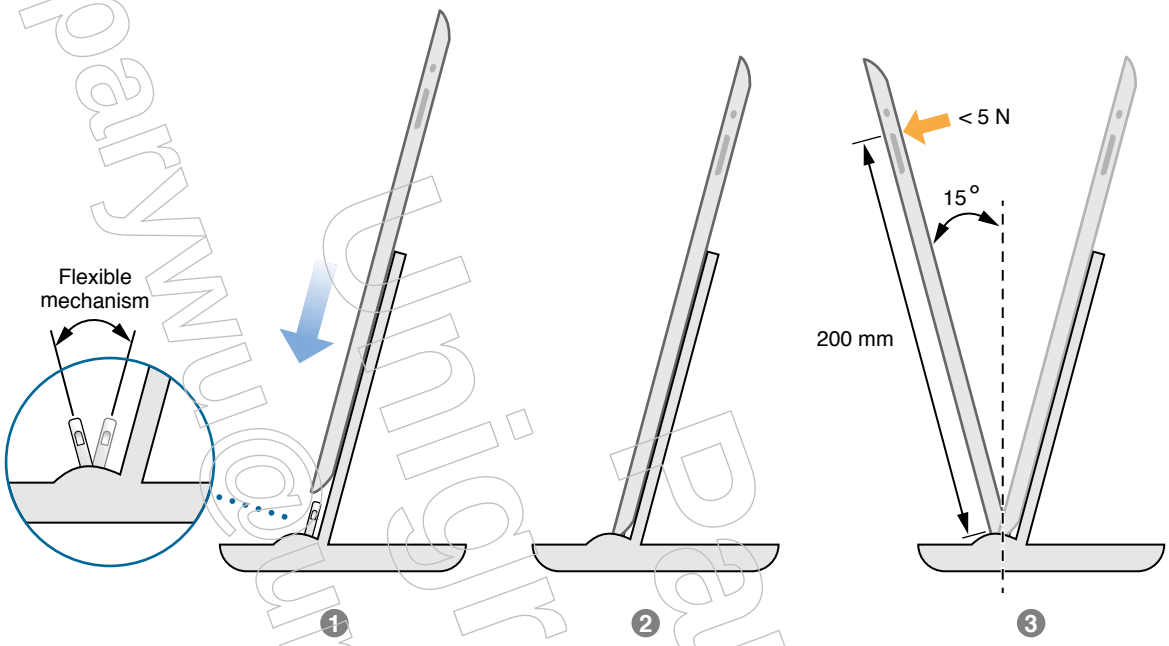


The accessory must have a flexible mechanism that allows a user to rotate the iPad from its rest position to at least 15 degrees from vertical. This mechanism must prevent the user from applying more than 5 N force perpendicular to the iPad at a distance of 200 mm from the bottom of the device before the device reaches the compliant limit.

This flexibility requirement is illustrated in Figure 5-36 (page 148). Drawings 1 and 2 show the iPad being engaged with the connector plug. Drawing 3 shows the device responding to a force of 5 N or less, applied 200 mm above the bottom of the device, by rotating up to 15 degrees from its vertical position.

Paryvu@unigadget.com.tw Unigadget LTD

Figure 5-36 iPad flexible mechanism



5.4.3.3 iPod nano Dock Requirements

Any dock accessory that is intended to be used with an iPod nano, or which can reasonably be connected to an iPod nano, must comply with the requirements in this section in addition to the general requirements.

The accessory must have a backstop that contacts the iPod nano at a point at least 15 mm vertically from the bottom of the device when a force of 10 N is applied perpendicular to the screen of the device at a point 50 mm vertically from the bottom of the device.

The accessory should contact the bottom of the device at two points, each of which is at least 20 mm away from the side of the connector, when the device is in the rest position.

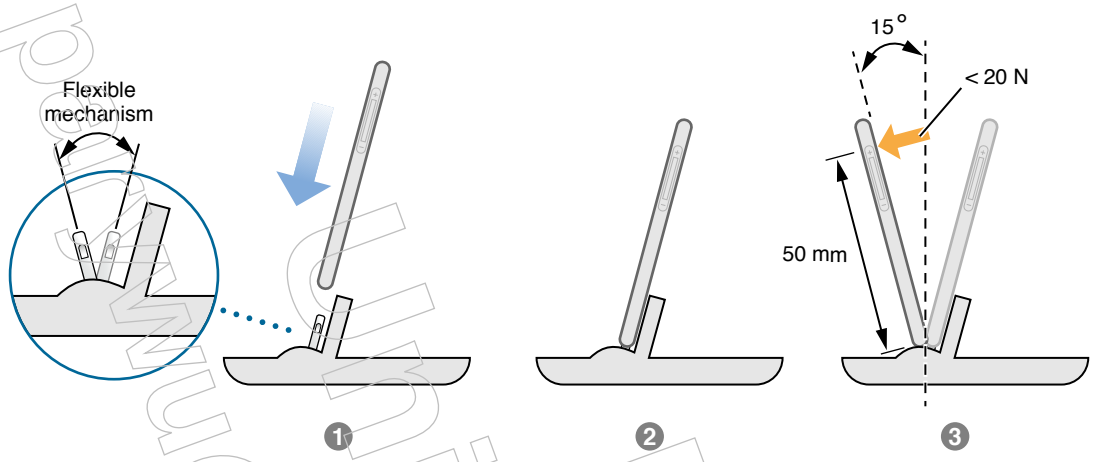
The accessory and its backstop should not block any speakers or microphones.

The accessory must have a flexible mechanism that allows a user to rotate the iPod nano from its rest position to at least 15 degrees from vertical by only applying a load to the device. This mechanism must prevent the user from applying more than 20 N force perpendicular to the iPhone or iPod touch at a distance of 50 mm from the bottom of the device before the device reaches the compliant limit.

This flexibility requirement is illustrated in Figure 5-37 (page 149). Drawings 1 and 2 show the iPod nano being engaged with the dock connector plug. Drawing 3 shows the device responding to a force of 20 N or less, applied 50 mm above the bottom of the device, by rotating up to 15 degrees from its vertical position.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 5-37 iPod nano flexible mechanism



5.4.3.4 Lightning (C48)/Lightning (C68) Docks

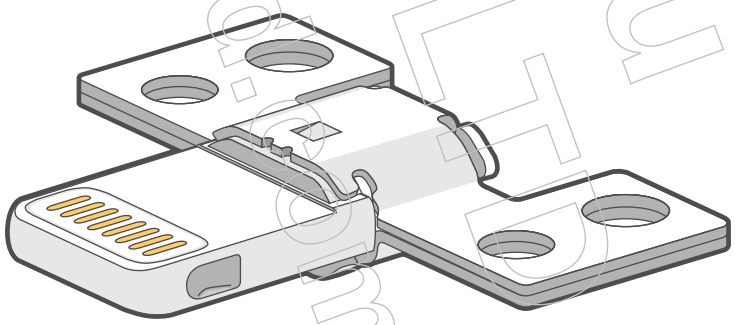
All docks incorporating a Lightning (C48) or Lightning (C68) connector must additionally comply with the requirements in this section.

All docks that incorporate a Lightning (C48) or Lightning (C68) connector must pass the test procedures specified in Dock Accessories (page 165) and Lightning (C48)/Lightning (C68) Accessories (page 156).

5.4.3.4.1 Shielding

All docks incorporating a Lightning (C48) or Lightning (C68) connector must incorporate a SUS stamped shield as shown in Figure 5-38 (page 149). Use of 0.40 mm thick SUS 304 1/2H material is recommended.

Figure 5-38 Lightning (C48) Dock Shielding Diagram

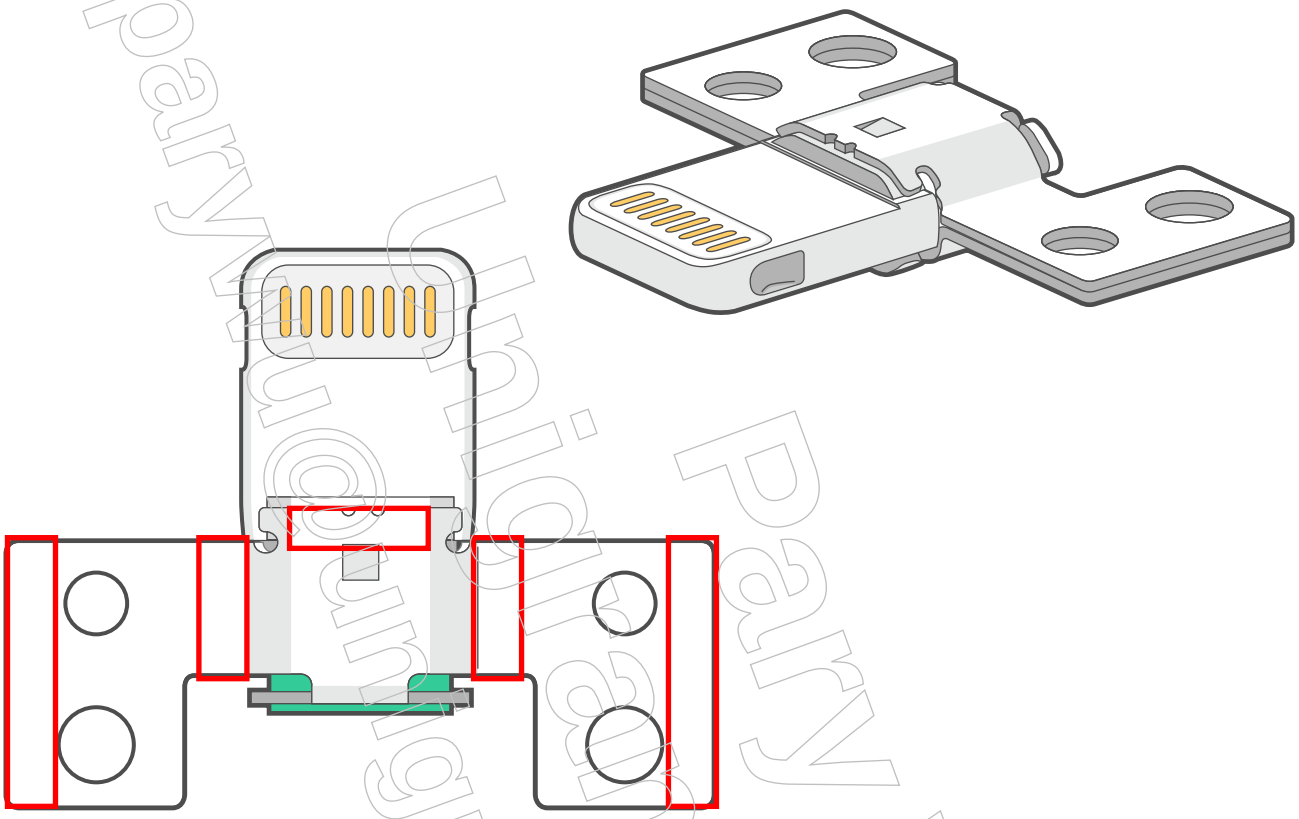


Shields must be free of burrs. Electropolishing is recommended as burrs can cut into wires and cause failures either in the factory or in the field.

The SUS stamped shield must be laser welded to the Lightning (C48) or Lightning (C68) ground ring as shown in Figure 5-39 (page 150). Recommended flange welding areas are also indicated. See Laser Welding (page 144).

Apple Confidential Information Under NDA Only

Figure 5-39 Lightning (C48) Dock Shield Laser Welding Diagram

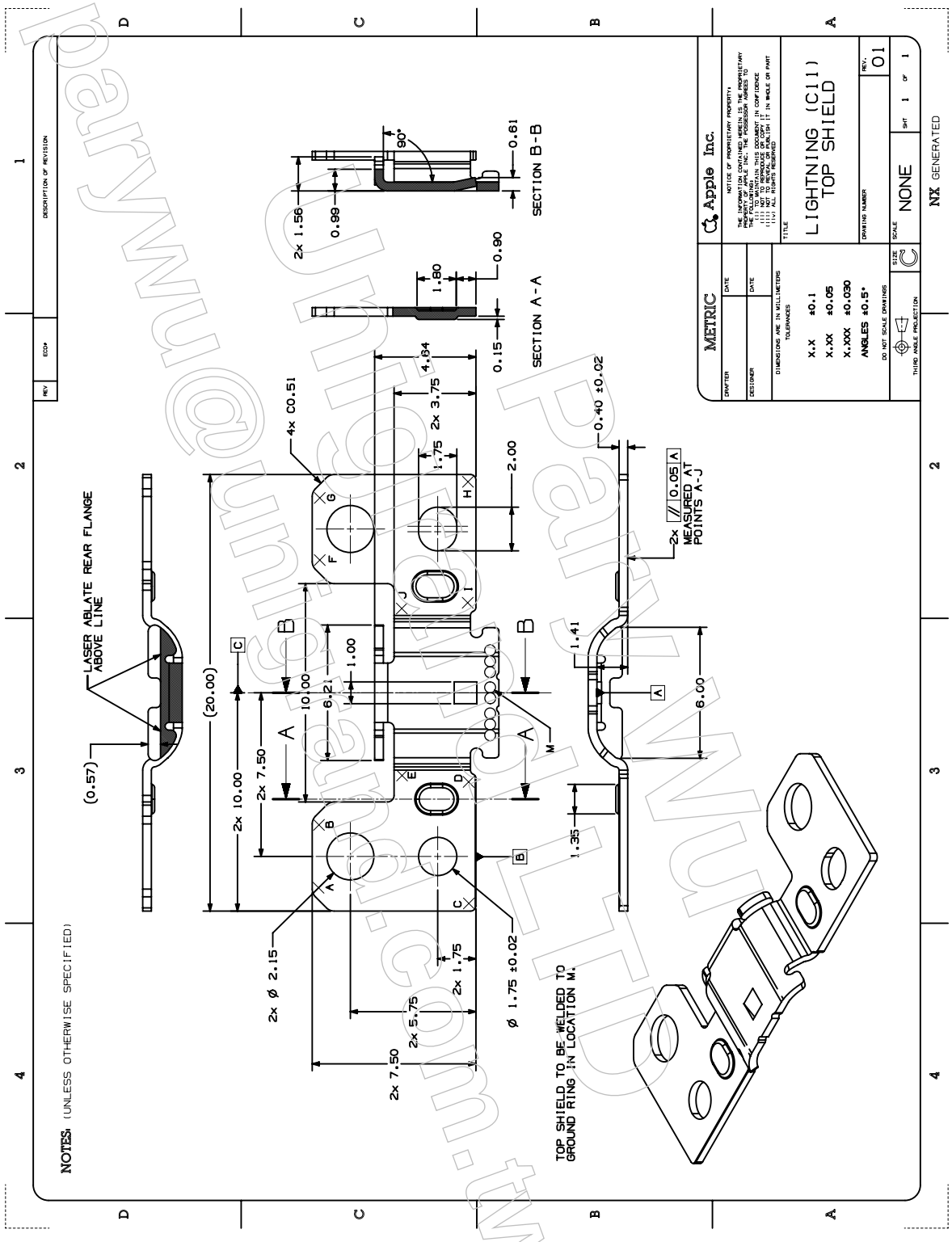


Laser welds between shields and Lightning (C48) or Lightning (C68) must withstand 2000 Nmm bending force. See Dock Shield Bend Test (C48, C68) (page 165).

Apple recommends the following two-part dock shield design for Lightning (C48).

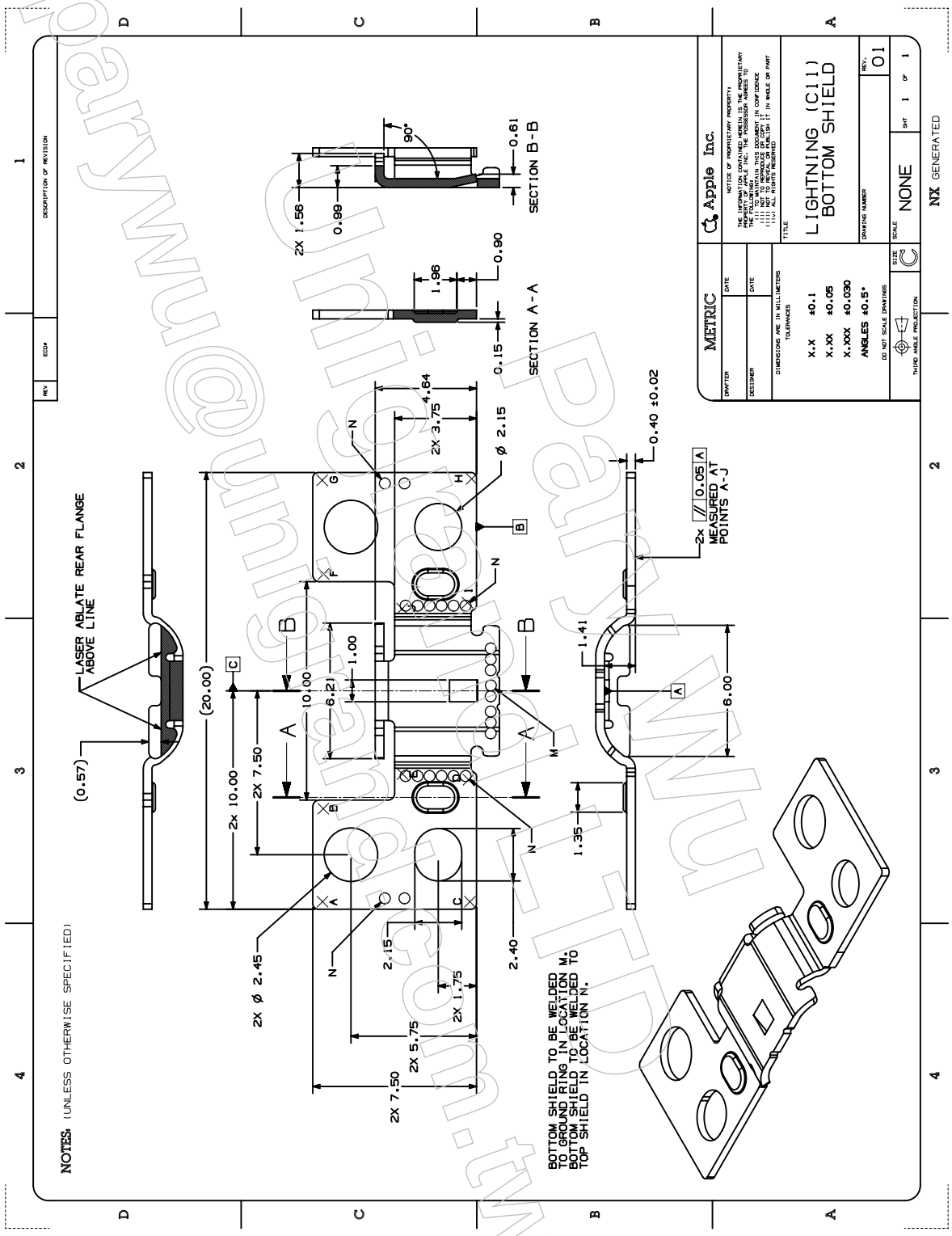
Pary Wu Original LTD
Pary Wu@originalpd.com.tw

Figure 5-40 Lightning (C48) Recommended Dock Shield Top



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 5-41 Lightning (C48) Recommended Dock Shield Bottom



Foxconn Confidential Property
Paywu@unionpeople.com.tw

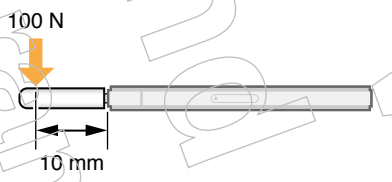
5.4.4 Dongle Accessories

A dongle accessory is an accessory that integrates the Lightning connector in an enclosure along with other components. A dongle accessory can be moved relative to the Apple device (unlike in a form-fitting accessory, see Form-Fitting Accessories (page 153)) and is not designed to hold up the Apple device (unlike in a dock accessory, see Dock Accessories (page 144)).

The following requirements apply to all dongle accessories:

- All dongle accessories must use either:
  - Lightning (C48) or Lightning (C68) with a shield (see Shielding (page 149))
  - Lightning (C11)
  - Lightning (C12)
- Dongle accessories that function exactly like the Apple Lightning to Micro-USB Adapter must use Lightning (C10B) or Lightning (C48B) with a shield. (See Shielding (page 137) and Shielding (page 149) for some options.)
- All dongle accessories must break or bend before the user can apply a 100 N force 10 mm away from the bottom of the device as illustrated in Figure 5-42 (page 153).

Figure 5-42 Dongle Maximum Allowable Force



5.4.5 Form-Fitting Accessories

A form-fitting accessory is an accessory that integrates the Lightning connector in such a way that it allows no relative movement between the Lightning connector and the Apple device when connected. Battery pack cases and game controllers which enclose the Apple device are examples of form-fitting accessories. Form-fitting accessories that substantally enclose the Apple device must comply with Cases (page 336).

> **Note:** Form-fitting accessories must must avoid degrading the wireless performance of Apple devices. See RF Transmission and Reception (page 52)

Form-fitting accessories may mount the Lightning connector rigidly without any compliance. However, the rigid mounting should not contact the connector's PCB in any way, as shown in Figure 5-16 (page 132).

Apple Confidential Information Licensed to Battery Wu

If a Lightning (C48) is used, it must incorporate a shield. The dock shield is recommended, see Shielding (page 149).

5.5 Connector Electrical Requirements

Accessories do not need to supply power to the Lightning connector. Instead, the connector draws a small amount of power from the Apple device upon insertion. All power supplied by the accessory to the Device Power pad/pin is routed to the Apple device.

5.5.1 Connector Ground Pad/Pin Connection Requirements

The following requirements apply to connections made to the ground pads/pins:

- Ground/Ground 1 must be permanently connected to the accessory's system ground.
- For Lightning (C10), Lightning (C11), Lightning (C48), and Lightning (C68), Ground 2 must be connected to the accessory's system ground if the accessory draws power from the Apple device, but may be left unconnected (NC) otherwise.
- For Lightning (C12), Ground 2 must be connected to the accessory's system ground.
- USB Host or Device Mode accessories must connect a drain wire to Ground/Ground 1.
- The Lightning connector shield must not be used as the only ground path.

5.5.2 Connector Shielding Requirements

All accessory components that carry power or data signals to and from the Lightning connector pads/pins must be shielded by grounded metal in accordance with EMI, EMC, and RF regulations.

5.5.3 Connector ESD Protection Requirements

ESD protection is present on all pads/pins in the Lightning connector.

5.5.4 Connector Signal Integrity Requirements

The following signal integrity requirements apply to all accessories that transmit or receive USB data.

- The accessory must comply with all differential insertion loss requirements as specified in Section 7.1.17 in the USB 2.0 specification.
- The accessory must comply with all impedance requirements as specified in Section 7.1.1.3 in the USB 2.0 specification.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

5.5.5 Lightning (C48)/Lightning (C68) Accessory Testing Requirements

Every accessory that integrates a Lightning (C48) or Lightning (C68) connector must pass the following tests on the factory line:

- Over voltage protection (see Over Voltage Protection Test (page 156))
- Current limit (see Current Limit Test (page 157))
- Quiescent current consumption (see Quiescent Current Consumption Test (page 158))

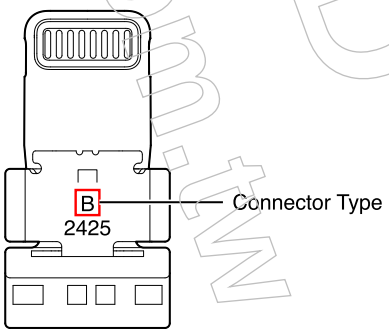
5.6 Test Procedures

5.6.1 All Accessories

5.6.1.1 Connector Version and Configuration

1. Via visual inspection, verify that the accessory uses Lightning (C10), Lightning (C11), Lightning (C12), Lightning (C48), or Lightning (C68) in accordance with the requirements in Connector Versions (page 115).
2. Via visual inspection, verify that the accessory uses the correct Lightning configuration (A, B, C, E, etc.)
   - Lightning (C10), Lightning (C11) and Lightning (C12) connectors have the configuration laser etched on the shield as shown in Figure 5-43 (page 155).
   - Lightning (C48) and Lightning (C68) connector trays will have a unique color stripe down the center.
     - Lightning (C48A)/Lightning (C68A) trays have a green stripe.
     - Lightning (C48B)/Lightning (C68B) trays have a yellow stripe.
     - Lightning (C48C)/Lightning (C68C) trays have a blue stripe.
     - Lightning (C68E) trays have an orange stripe.

Figure 5-43 Identifying Lightning (C10), Lightning (C11), or Lightning (C12) connector configuration



Paul Wu
paul@unigiantech.com.tw
Unigiantech LTD

5.6.1.2 Static Plug Exposure

Using a caliper, verify that at least 6.47 mm of the connector plug is exposed throughout the contact area of the Apple device as shown in Figure 5-17 (page 132).

5.6.1.3 Plug Exposure Under Insertion Force

1. Place the accessory in a clamp or other mechanism to securely hold it in place.
2. Using a calibrated force gauge, apply 30 N of pressure to the connector in the direction of device insertion via the measuring tip of the force gauge.
3. Using a caliper, verify that at least 6.47 mm of the connector plug is exposed throughout the contact area of the Apple device as shown in Figure 5-17 (page 132).

5.6.1.4 Plug Mounting Under Extraction Force

1. Place the accessory in a clamp or other mechanism to securely hold it in place.
2. Using a calibrated force gauge, apply 30 N of pressure to the connector in the direction of device extraction by clamping on to the plug and pulling on it.
3. Verify that the connector does not pull or break away from the accessory.

5.6.1.5 Plug Bias

Using a protractor or similar measuring device, measure the angular bias and verify that no more than 2 degrees of sideways bias is present as shown in Figure 5-17 (page 132).

5.6.1.6 Hidden Connector Faces

Via visual inspection, verify that all Apple Lightning Connector faces besides the cosmetic plug faces are hidden within the accessory as shown in Figure 5-18 (page 133).

5.6.2 Lightning (C48)/Lightning (C68) Accessories

5.6.2.1 Over Voltage Protection Test

Note: This procedure is required for all Lightning (C48) or Lightning (C68) integrations on the factory line. See Lightning (C48)/Lightning (C68) Accessory Testing Requirements (page 155).

Pericom Semiconductor Corporation
Pericom Technology Inc.
Pericom Engineering Corporation
Pericom Semiconductor (HK) LTD

Verify that the Lightning (C48) or Lightning (C68) connector is functioning correctly after integration by performing the following test procedure:

1. For cable integrations that terminate in a USB type-A connector:
   a. Connect the USB side of the cable to the MFi USB Breakout.
   b. Connect the MFi USB Breakout to the power supply (pin 9 to power and pin 3 to ground).
2. For all other integrations, connect the accessory's power and ground terminating points to the power supply.
3. Connect the Lightning (C48) or Lightning (C68) side to the MFi Lightning Breakout.
4. Attach digital multimeter across PWR and GND on MFi Lightning Breakout to measure voltage.
5. Apply 5V on the power supply.
6. Ramp the power supply from 5V to 9V
7. Verify that the output voltage on PWR tracks VBUS until 6.0V ≤ VBUS ≤ 6.6V. At this point PWR must drop to 0V and remain until VBUS has dropped back below this threshold.

5.6.2.2 Current Limit Test

Note: This procedure is required for all Lightning (C48) or Lightning (C68) integrations on the factory line. See Lightning (C48)/Lightning (C68) Accessory Testing Requirements (page 155).

Verify that the Lightning (C48) or Lightning (C68) connector is functioning correctly after integration by performing the following test procedure:

1. For cable integrations that terminate in a USB type-A connector:
   a. Connect the USB side of the cable to the MFi USB Breakout.
   b. Connect the MFi USB Breakout to the power supply (pin 9 to power and pin 3 to ground).
2. For all other integrations, connect the accessory's power and ground terminating points to the power supply.
3. Set current limit on power supply to at least 20 mA.
4. Connect the Lightning (C48) or Lightning (C68) side to the MFi Lightning Breakout.
5. Attach digital multimeter inline with PWR to measure current.
6. Apply 5V on the power supply.
7. Attach 3.5V SMU load (see Note below) across PWR and GND on MFi Lightning Receptacle, measure current through load. Current must be < 8 mA.

Purvi Uniyal
pruniyal@uniyalfield.com

8. Attach a short across PWR and GND on MFi Lightning Receptacle, measure current through load. Current must not exceed 15 mA.

Note: The following four-quadrant SMU or equivalent set of power loads are required:

- 500Ω, > 100 mW
- "short" load, > 20 mA

5.6.2.3 Quiescent Current Consumption Test

Note: This procedure is required for all Lightning (C48) or Lightning (C68) integrations on the factory line. See Lightning (C48)/Lightning (C68) Accessory Testing Requirements (page 155).

Verify that the Lightning (C48) or Lightning (C68) connector is functioning correctly after integration by performing the following test procedure:

1. For cable integrations that terminate in a USB type-A connector:
   a. Connect the USB side of the cable to the MFi USB Breakout.
   b. Connect the MFi USB Breakout to the power supply (pin 9 to power and pin 3 to ground).
2. For all other integrations, connect the accessory's power and ground terminating points to the power supply.
3. Attach digital multimeter inline with VBUS to measure current.
4. Apply -1 V on the power supply and measure current consumption. Current must not exceed 17 mA.
5. Apply 5 V on the power supply and measure current consumption. Current must not exceed 1 mA.

5.6.3 Mixed 30-pin/Lightning Connector Accessories

5.6.3.1 iAP1/iAP2 Messages

1. Run ATS in "ATS Capture" mode on each connector.
2. Run through all possible use cases.
3. Verify there are no errors and warnings in the summary panel and ATS trace.
4. Verify that both accessories are using the same data transport.

Pegatron Confidential
Unauthorized Copy

5.6.3.2 Power Only

1. Verify that both connectors can provide the required current when all ports are under a full load, see Lightning Connectors on Power-Providing Accessories (page 483).
2. Verify connectors charge each Apple device simultaneously when connected.
3. Verify that iAP1/iAP2 messages are used to inform the device of available power.

5.6.3.3 Power and Data

1. Connect an Apple device on one connector, verify that a data transport has been established.
2. Connect a second Apple device to the secondary connector.
3. Verify that a data transport has been terminated with the first attached Apple device.
4. Verify that a data transport has been established with the last attached Apple device.

5.6.4 Cable Accessories

Using a caliper, verify that the connector enclosure dimensions are smaller than the limits specified in Figure 5-19 (page 133).

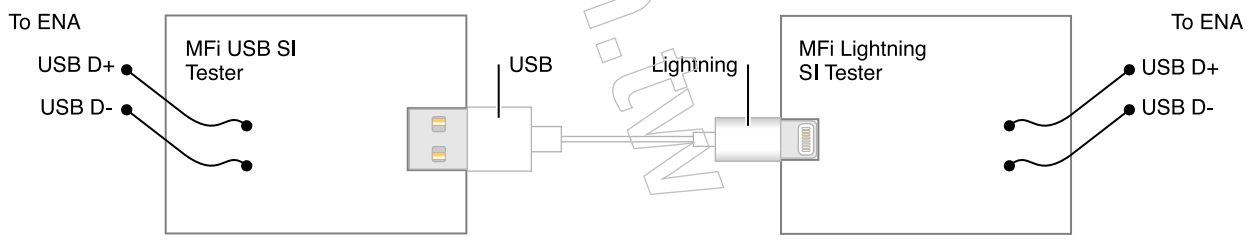
5.6.4.1 Signal Integrity Tests (C10A, C10B, C48A, C48B, C68A, C68B)

All cables that incorporate a Lightning (C10A), Lightning (C10B), Lightning (C48A), Lightning (C48B), Lightning (C68A), or Lightning (C68B) connector on one end and a USB connector plug on the other must be tested for differential insertion loss and differential impedance.

Equipment needed for differential insertion loss and differential impedance testing:

- Agilent E5071C-4K5 ENA with Enhanced TDR
- Agilent N4433A-010 RF ECal module, 300 kHz to 20 GHz, 3.5 mm, 4-port
- MFi USB SI Tester (MFI677-2103)
- MFi Lightning SI Tester (MFI677-1642)

Figure 5-44 Signal Integrity Test Setup for USB cables



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

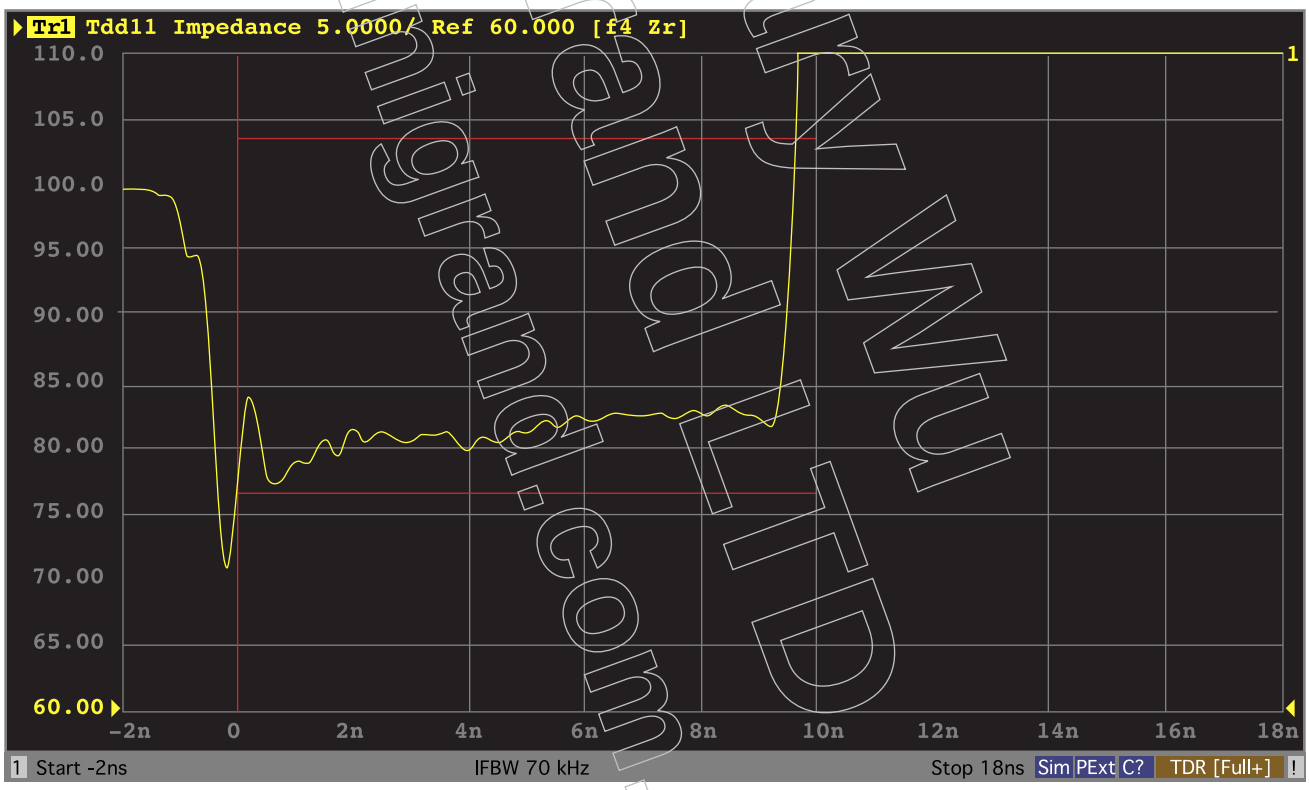
On the Agilent ENA, enter TDR mode and run the Setup Wizard and select "Full Calibration". Make sure "Differential Two-Port" is selected when prompted. Run through the steps required to calibrate and test the accessory.

In the "Fixture Compensation" step, make the following connections:

1. MFi USB SI Tester "PIN3" to port 1 of Agilent ENA
2. MFi USB SI Tester "PIN2" to port 2 of Agilent ENA
3. MFi Lightning SI Tester "PIN2" to port 3 of Agilent ENA
4. MFi Lightning SI Tester "PIN3" to port 4 of Agilent ENA

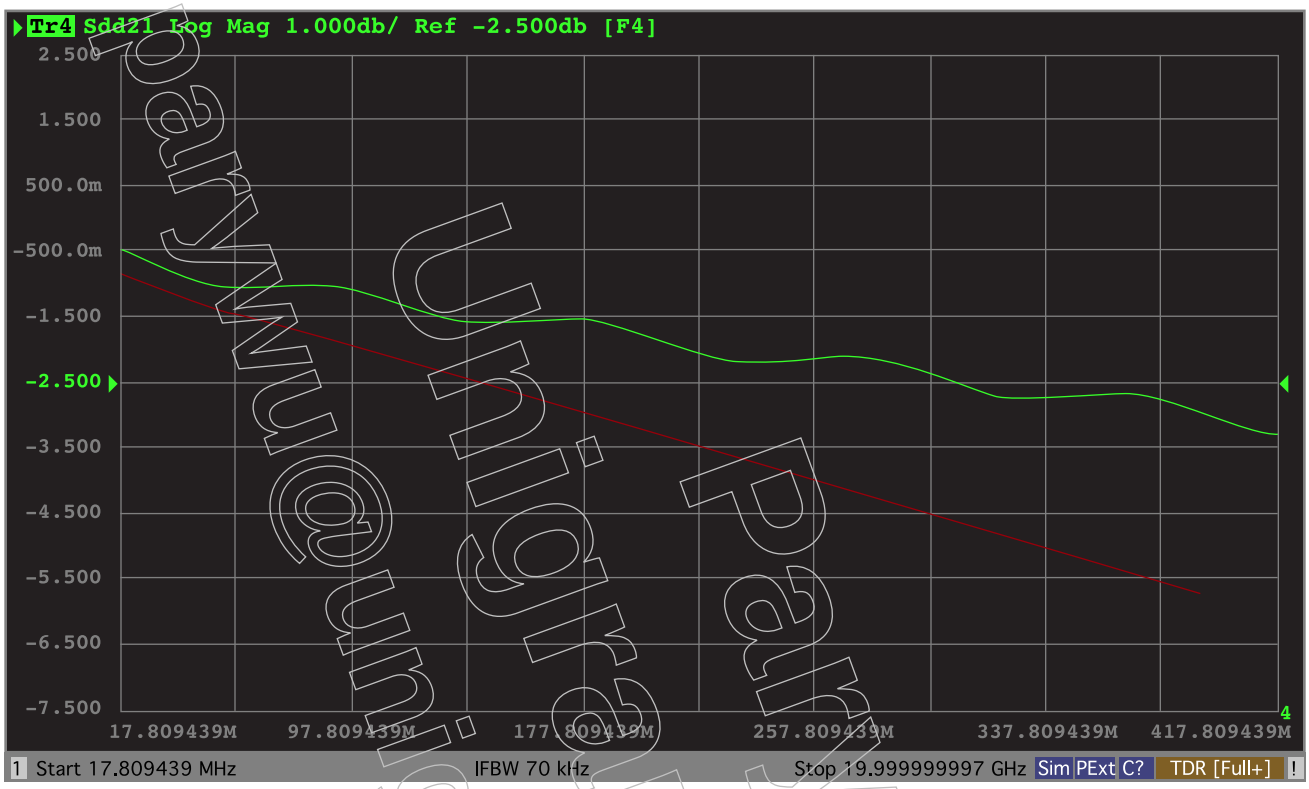
Set limit lines for Tdd11 (differential impedance graph) and Sdd12/Sdd21 (differential insertion loss graph). Results should look similar to Figure 5-45 (page 160) and Figure 5-46 (page 161) as shown below.

Figure 5-45 Differential impedance graph with limit lines



Apple Confidential
Tony Wu
tonywu@nigrane.com.tw

Figure 5-46 Differential insertion loss graph with limit lines



Note: Differential insertion loss and differential impedance requirements can be found in chapter 7 of the *Universal Serial Bus Specification*, revision 2.0.

5.6.4.2 DC Resistance Tests (C10, C48, C68)

For cables that incorporate a Lightning (C10), Lightning (C48), or Lightning (C68) connector, refer to Figure 5-47 (page 162), Figure 5-48 (page 162), and Figure 5-49 (page 162) for DC resistance test setup.

Equipment needed for DC resistance testing:

- Digital multi-meter
- Power supply
- 2.5A load
- MFi Lightning Tester (MFI677-00167)
- MFi USB Breakout (MFI677-2104)

Party Wu
Unigroup LTD
party.wu@unigroup.com.tw

Figure 5-47 DCR Ground in Parallel with Shield Conductor Test Setup ($R_{GND||SHIELD}$)

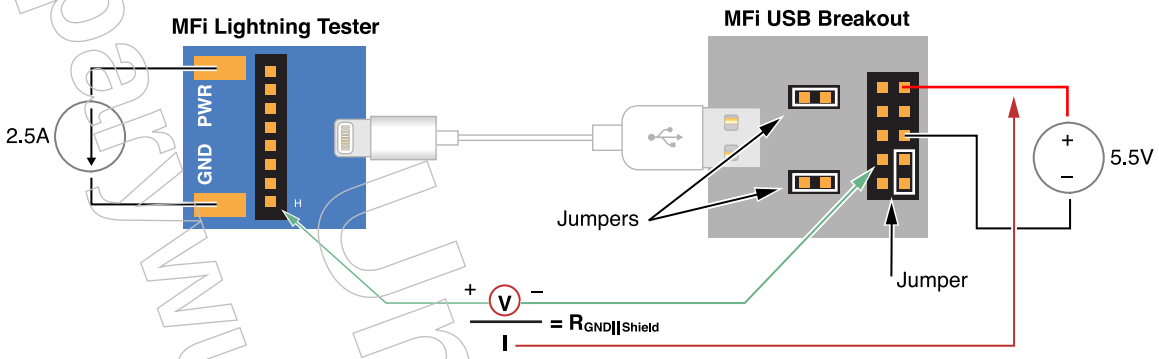


Figure 5-48 DCR VBUS Conductor Test Setup (R_{VBUS})

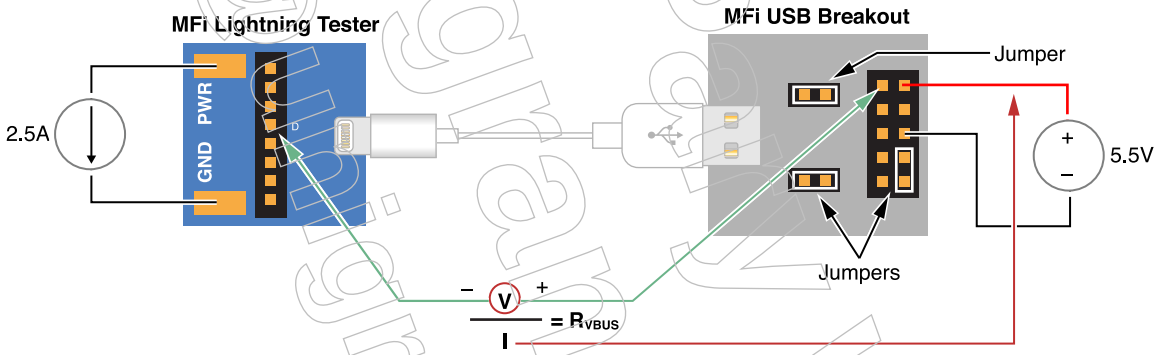
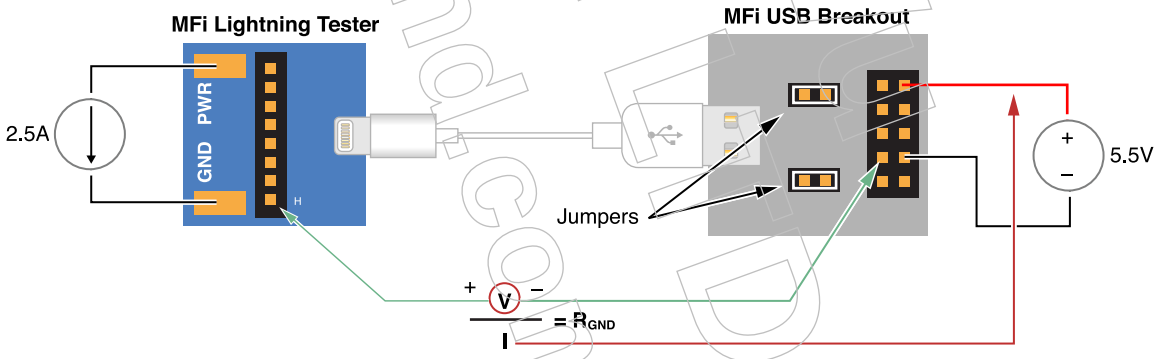


Figure 5-49 DCR Ground Conductor Test Setup (R_{GND})



1. For cable integrations that terminate in a USB type-A connector:
   a. Connect the USB side of the cable to the MFi USB Breakout.
   b. Connect the MFi USB Breakout to the power supply (pin 9 to power and pin 3 to ground).
2. For cable integrations that do not terminate in a USB type-A connector:
   a. Connect the cable accessory's power and ground terminating ends of the PCB to the power supply.

Apple Confidential
Preliminary Draft Not Reviewed for Content

3. Connect the Lightning side to the MFi Lightning Tester.
4. Attach a 2.5A load across PWR and GND on MFi Lightning Tester.
5. Apply 5.5V on the power supply.
6. For ground conductor in parallel with shield resistance ($R_{GND||SHIELD}$):
   a. Measure the voltage difference between pin H of the MFi Lightning Tester and pin 4 of the MFi USB Breakout.
   b. Measure the current flowing through pin 9 of the MFi USB Breakout and the power supply.
   c. Calculate $R_{GND||SHIELD}$. $R_{GND||SHIELD}$ should not exceed what is specified in Table 39-2 (page 484). See Figure 5-47 (page 162) for more information.
7. For VBUS conductor resistance (R_{VBUS}):
   a. Measure the voltage difference between pin D of the MFi Lightning Tester and pin 10 of the MFi Lightning Breakout.
   b. Measure the current flowing through pin 9 of the MFi USB Breakout and the power supply.
   c. Calculate R_{VBUS}. R_{VBUS} should not exceed what is specified in Table 39-2 (page 484). See Figure 5-48 (page 162) for more information.
8. For ground conductor resistance (R_{GND}):
   a. Measure the voltage difference between H of the MFi Lightning Tester and pin 4 of the MFiUSB Breakout.
   b. Measure the current flowing through pin 9 of the MFi USB Breakout and the power supply.
   c. Calculate R_{GND}. R_{GND} should not exceed what is specified in Table 39-2 (page 484). See Figure 5-49 (page 162) for more information.

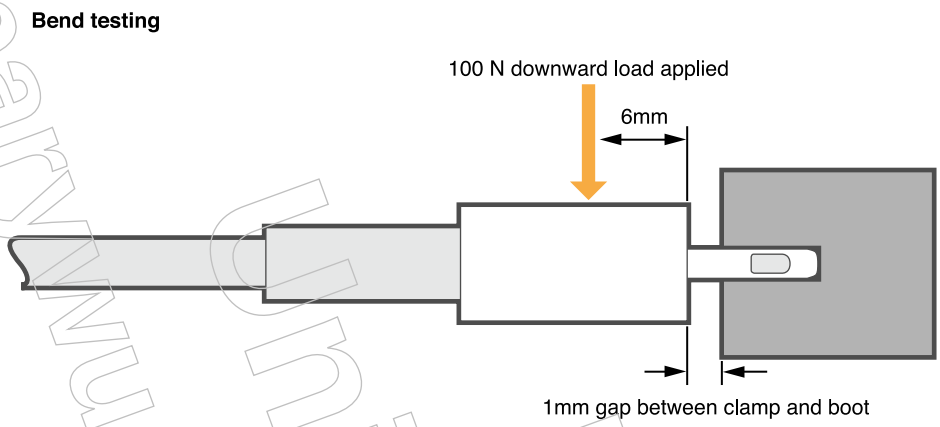
5.6.4.3 Cable Shield Bend Test (C48, C68)

All cables that incorporate a Lightning (C48) or Lightning (C68) connector must past this test.

Verify that the cable shielding is still intact after performing a bend test as shown in Figure 5-50 (page 164).

Property of Unigrand Ltd.
Unigrand@unigrand.com.tw

Figure 5-50 Bend Test Setup for Lightning Cable Shield

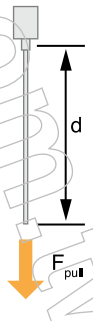


5.6.4.4 180° Cable Bend Test

Verify that the cable shielding is still intact after performing a 180° cable bend test:

1. F_{Pull} = 0.453 kgf, d = 225 mm. See Figure 5-51 (page 164)
2. Cable tester should rotate at 42 RPM. Apple suggests a cable tester from http://www.yhtec.com.cn, part number: YH-8801K.
3. Fix connector so strain relief is centered on a rotating surface. Strain relief and cable should be free to bend. See Figure 5-52 (page 165).
4. Rotate the tester counter clockwise by 180°. See Figure 5-53 (page 165).
5. Rotate the tester clockwise by 360°. See Figure 5-54 (page 165).
6. Return the tester to its original position. See Figure 5-52 (page 165).
7. Cable should go through at least 50 cycles and still remain functional.

Figure 5-51 Bend Test Setup for Lightning Cables



Partwu@unicornpartwu.com Unicorn Partwu LTD

Figure 5-52 Tester Setup for Lightning Cables

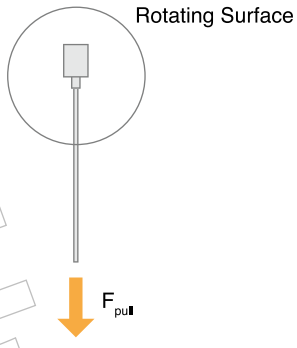


Figure 5-53 180° Cable Bend Test Setup for Lightning Cables

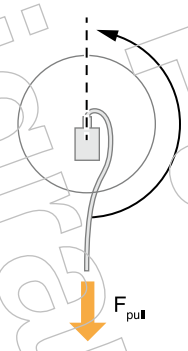
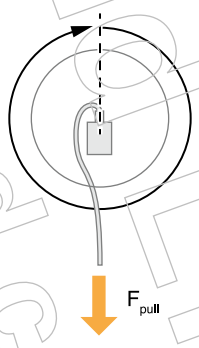


Figure 5-54 360° Cable Bend Test Setup for Lightning Cables



5.6.5 Dock Accessories

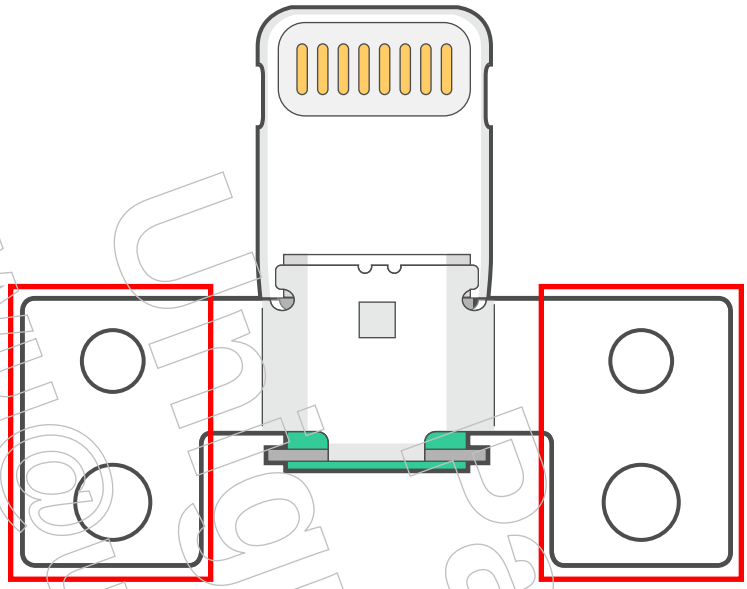
5.6.5.1 Dock Shield Bend Test (C48, C68)

All docks that incorporate a Lightning (C48) or Lightning (C68) connector must past this test.

Clamp on the shields as shown in Figure 5-55 (page 166).

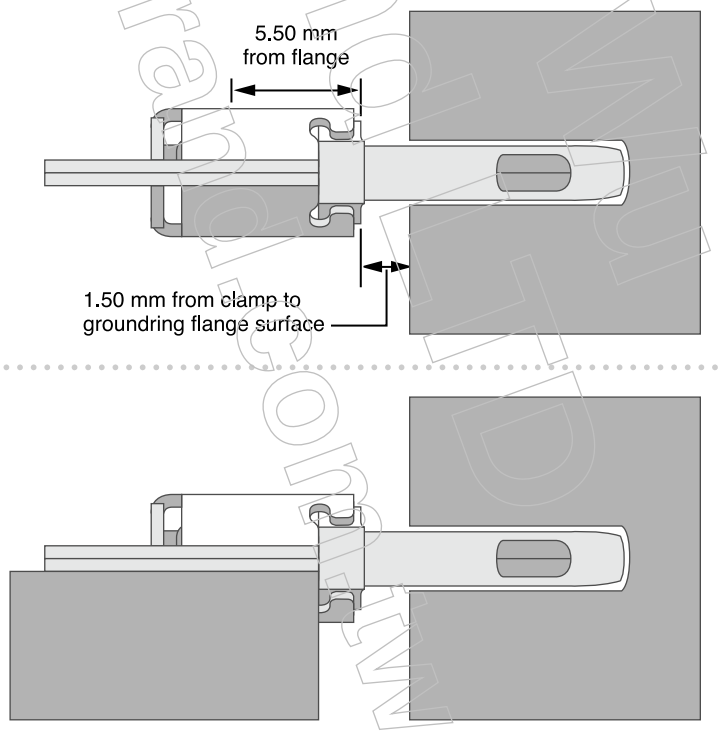
Pary Wu
Unigrand LTD
paryw@unigrand.com.tw

Figure 5-55 Bend Test Clamp Region for Lightning Dock Shield



Ensure a 1.5 mm gap between plug clamp and groundring flange as shown in Figure 5-56 (page 166) and apply a force of at least 286 N at a distance of 5.50 mm from flange.

Figure 5-56 Bend Test Setup for Lightning Dock Shield



Verify that laser welds are intact.

paywall@uniquepatents.com
Unique Patents

5.6.5.2 iPhone and iPod touch Docks

5.6.5.2.1 Backstop

1. Connect the iPhone or iPod touch to the dock.
2. Using a calibrated force gauge, apply 5 N of force perpendicular to the screen at a point 100 mm vertically from the bottom of the device as shown in Figure 5-32 (page 145).
3. Verify that a backstop contacts the device at a point at least 15 mm vertically from the bottom of the device when the force is applied, as shown in figure Figure 5-31 (page 145).

5.6.5.2.2 Flexible Mechanism

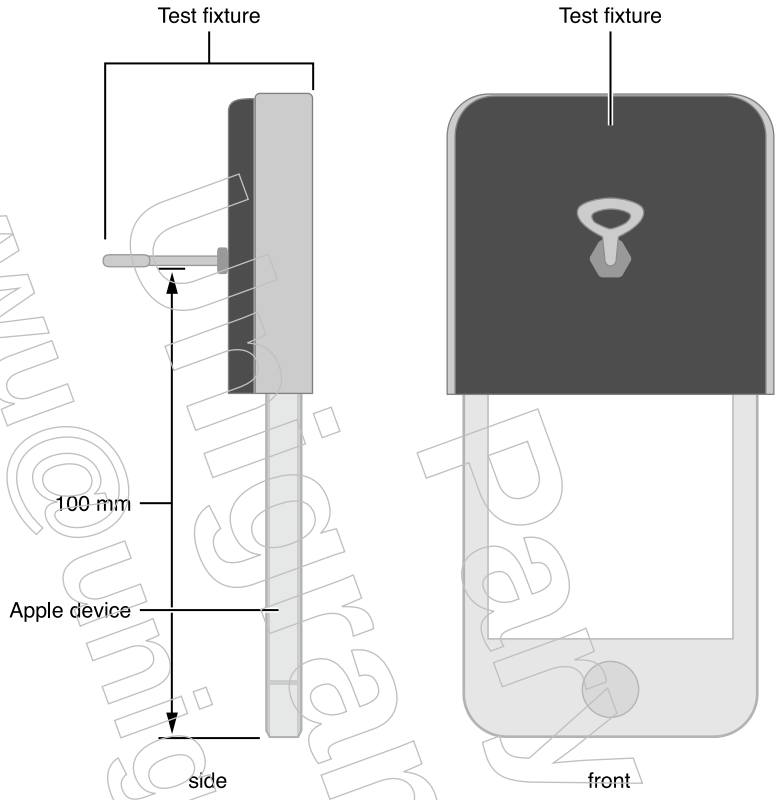
The flexible mechanism requirement is shown in Figure 5-32 (page 145). Drawings 1 and 2 show the device being engaged with the dock connector plug. Drawing 3 shows the device responding to a force of 10 N or less, applied 100 mm above the bottom of the device, by rotating up to 15 degrees from vertical.

1. Connect the iPhone or iPod touch to the dock and allow both to settle into a rest position.
2. Using a calibrated force gauge, apply force perpendicular to the device at a point 100 mm from the bottom of the device. Verify that a flexible mechanism prevents the user from applying more than 10 N of force to the device.
3. Using a protractor, verify that the device can be rotated 15 degrees from vertical.

Some docks have a backstop or other surface that limits access to the back of the device. A mechanism fitting over the front of the device similar to the one shown in Figure 5-57 (page 168) should be constructed to test them.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 5-57 Fitting mechanism to enable testing of certain docks that limit device access



5.6.5.3 iPad Docks

5.6.5.3.1 Backstop

1. Connect the iPad to the dock.
2. Using a calibrated force gauge, apply 2.5 N of force perpendicular to the screen at a point 200 mm vertically from the bottom of the device as shown in Figure 5-36 (page 148).
3. Verify that a backstop contacts the device at a point at least 50 mm vertically from the bottom of the device when the force is applied.

5.6.5.3.2 Flexible Mechanism

The flexible mechanism requirement is shown in Figure 5-36 (page 148). Drawings 1 and 2 show the device being engaged with the dock connector plug. Drawing 3 shows the device responding to a force of 5 N or less, applied 200 mm above the bottom of the device, by rotating up to 15 degrees from vertical.

1. Connect the iPad to the dock and allow both to settle into a rest position.

Paywu@uniqrand.com Uniqrand Ltd Paywu

2. Using a calibrated force gauge, apply force perpendicular to the device at a point 200 mm from the bottom of the device. Verify that a flexible mechanism prevents the user from applying more than 5 N of force to the device.
3. Using a protractor, verify that the device can be rotated 15 degrees from vertical.

Some docks have a backstop or other surface that limits access to the back of the device. A mechanism fitting over the front of the device similar to the one shown in Figure 5-57 (page 168) should be constructed to test them.

5.6.5.4 iPod nano Docks

5.6.5.4.1 Backstop

1. Connect the iPod nano to the dock.
2. Using a calibrated force gauge, apply 20 N of force perpendicular to the screen at a point 50 mm vertically from the bottom of the device.
3. Verify that a backstop contacts the device at a point at least 15 mm vertically from the bottom of the device when the force is applied, as shown in figure Figure 5-31 (page 145).

5.6.5.4.2 Flexible Mechanism

The flexible mechanism requirement is shown in Figure 5-37 (page 149). Drawings 1 and 2 show the device being engaged with the dock connector plug. Drawing 3 shows the device responding to a force of 20 N or less, applied 50 mm above the bottom of the device, by rotating up to 15 degrees from vertical.

1. Connect the iPod nano to the dock and allow both to settle into a rest position.
2. Using a calibrated force gauge, apply force perpendicular to the device at a point 50 mm from the bottom of the device. Verify that a flexible mechanism prevents the user from applying more than 20 N of force to the device.
3. Using a protractor, verify that the device can be rotated 15 degrees from vertical.

Some docks have a backstop or other surface that limits access to the back of the device. A mechanism fitting over the front of the device similar to the one shown in Figure 5-57 (page 168) should be constructed to test them.

Paul Wang@Unigroup Tech.com.tw

5.6.6 Dongle Accessories

5.6.6.1 Shield Bend Test (C48, C68)

All dongles that incorporate a Lightning (C48) or Lightning (C68) connector must past one of the shield bend tests:

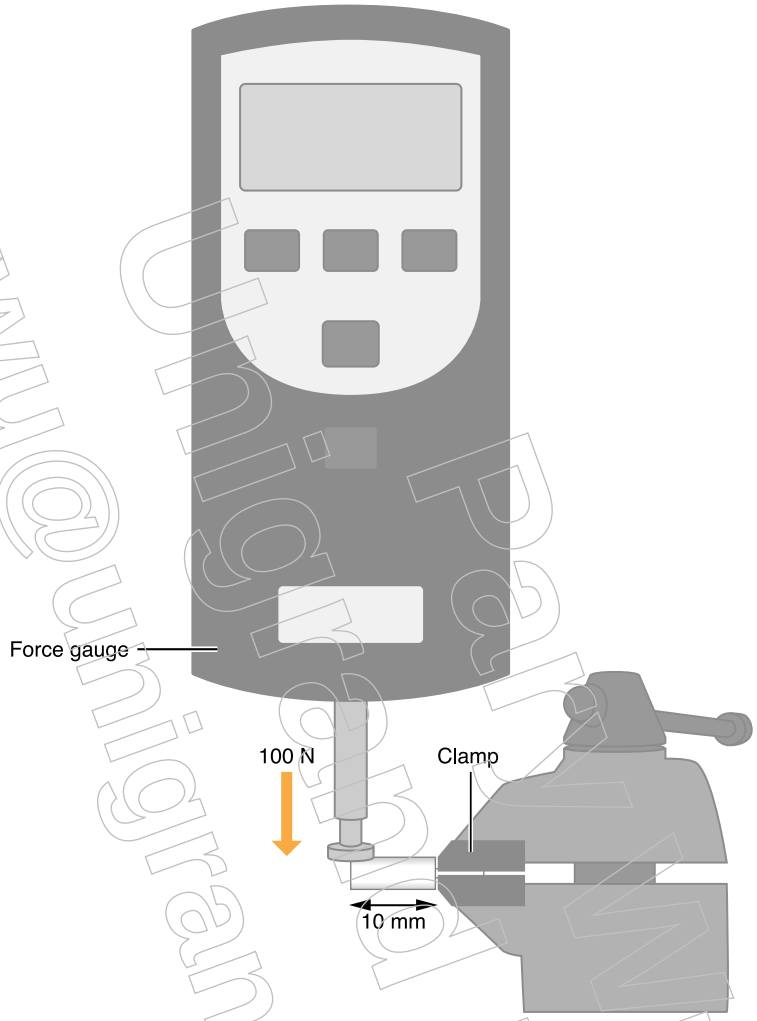
- Dock Shield Bend Test (C48, C68) (page 165)
- Cable Shield Bend Test (C48, C68) (page 163)

5.6.6.2 Break / Bend Test

1. Place the dongle's exposed plug surface into a clamp or other secure holding mechanism.
2. Position the measuring tip of a force gauge 10 mm away from the bottom of the dongle and apply 100 N of force as shown in Figure 5-58 (page 171).
3. Verify that the dongle breaks or bends before 100 N of force is applied.

Tony Wu
Unigrand LTD
tonywu@unigrand.com.tw

Figure 5-58 Dongle testing using a force gauge



partwu@unigrand.com.tw
Unigrand LTD Partner

6. Apple Lightning Receptacle

Note: This chapter is a Developer Preview and is not intended for use in the development of Proposed Products or Licensed Products under a MFi License. Although this content has been reviewed for accuracy, it is not final. Apple is supplying this content to help accessory developers plan for the adoption of the accessory interface features described herein. This information is subject to change, and accessories implemented according to this content must be tested with final operating system software and final documentation before going through self certification.

Accessories may incorporate a Lightning Receptacle (C37) to

- Draw power from Apple branded or MFi certified USB power adapters.
- Enable passthrough USB charge/sync for an Apple device that is connected to the accessory.

The Lightning Receptacle (C37) does not support any accessory interface features other than the ones listed above.

6.1 Lightning Receptacle (C37) Requirements

Accessories that incorporate an Lightning Receptacle (C37) must also support/integrate one or more of the following features:

- iAP2
- Lightning Audio Module (see Apple Lightning Audio Module (page 80))
- AirPlay
- Wireless Accessory Configuration

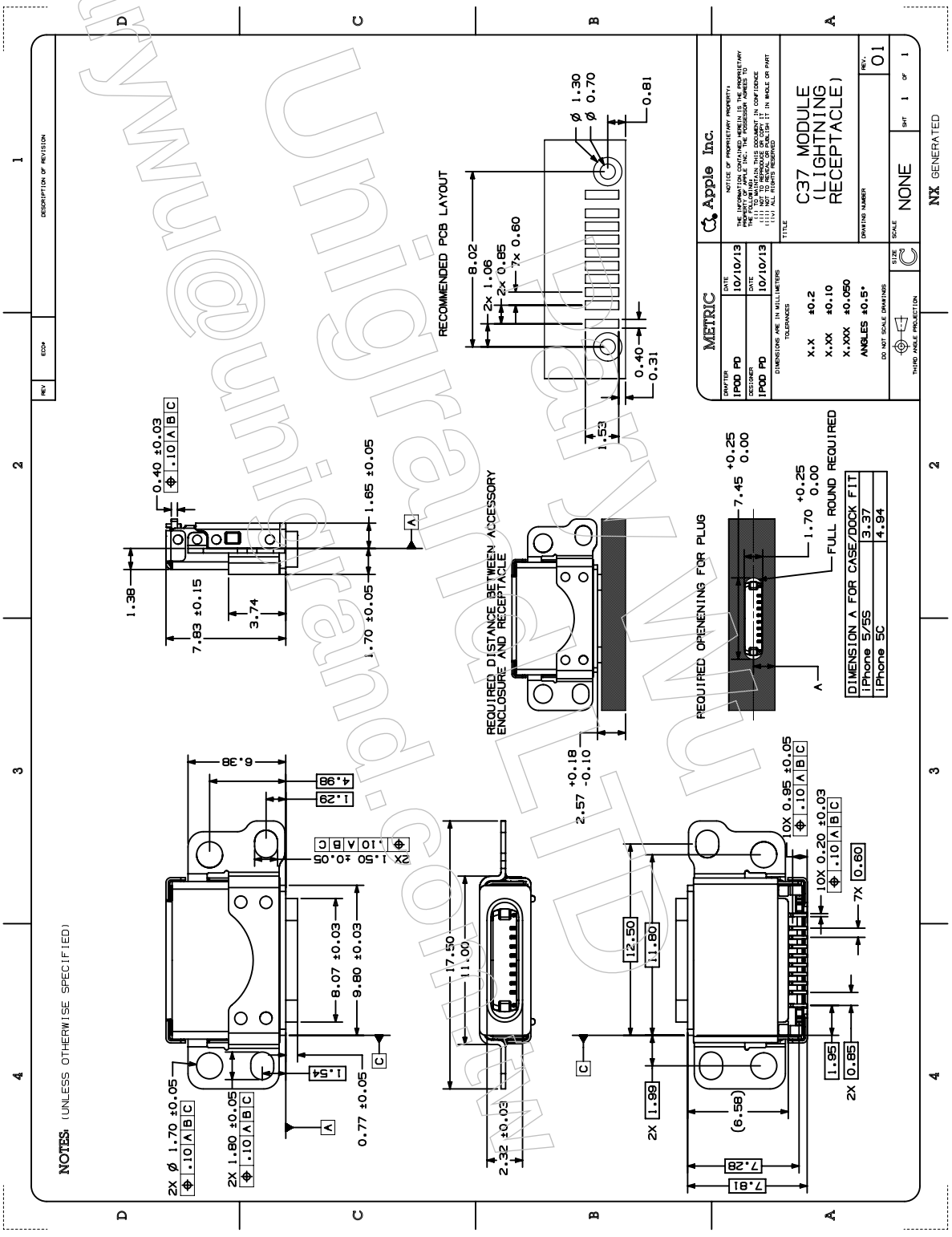
6.2 Lightning Receptacle (C37) Mechanical Requirements

Accessories must not incorporate more than one Lightning Receptacle (C37).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

The Lightning Receptacle (C37) is designed to withstand a standard reflow profile. Apple recommends a peak reflow temperature below 250 degrees C for 20 seconds maximum to prevent damage to the receptacle.

Figure 6-1 Lightning Receptacle (C37) Dimensions



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

6.2.1 Mounting

A stainless steel trim ring is recommended around the opening of the receptacle for maximum durability and robustness.

The receptacle must be recessed from the surface of the accessory as specified in Figure 6-1 (page 173). The depth of the receptacle to the surface of the opening must be at least 6.47 mm and at most 6.65 mm.

The opening for the Lightning connector must be fully rounded and meet the dimensions in Figure 6-1 (page 173).

The receptacle must be mounted in all 4 mounting holes with screws. Holes are designed to fit M1.2 screws.

6.2.2 Receptacle Encapsulation

Exposed contacts, solder joints and exposed PCB copper should be encapsulated after soldering with electrically nonconductive and liquid sealing compound, for moisture protection. Some suggested compounds that can be used for this purpose are:

- Loctite 3703 UV Encapsulant
- Henkel UV9060 UV Encapsulant
- Cemedine SuperX 8008 Neo Encapsulant (RTV)
- Dymax 29990 UV Encapsulant

A jet dispense process is recommended for applying encapsulation. Some recommended vendors of encapsulation equipment are:

- Nordson EFD/ASYMTEK (http://www.nordson.com)
- Musashi Engineering (http://www.musashi-engineering.co.jp)

6.3 Lightning Receptacle (C37) Electrical Requirements

Accessories that integrate a Lightning Receptacle (C37) must include the Lightning Receptacle Controller. See Apple Lightning Receptacle Controller (page 180).

Accessories that integrate a Lightning Receptacle (C37) must use one or more of the following data transports to connect to the Apple device:

- Serial
- Lightning Audio Module

Pary Wu
Unicraft and Interface Co., Ltd.
paryw@unicraft.com.tw

- Wi-Fi
- Bluetooth

Accessories that integrate a Lightning Receptacle (C37) and also connect to an Apple device via the Lightning connector may support passthrough USB charge/sync for the Apple device. To support passthrough charge/sync, such accessories must:

- Integrate one of the following Lightning connectors:
  - Lightning (C10B), Lightning (C48B), Lightning (C11E), Lightning (C12B), or Lightning (C68B)
  - Lightning (C10E), Lightning (C11E), Lightning (C12E), or Lightning (C68E)
- Connect USB D+, USB D-, Device Power, and Ground from the Lightning connector directly to the Lightning Receptacle Controller
- Not draw power from the Lightning Receptacle (C37) unless they also integrate an Lightning Audio Module and use it to reserve a portion of pass-through power for the accessory's own use (see accPassthroughPowerSourceInformation (0x24) (page 109)).

Accessories that draw power from the Lightning Receptacle (C37) must:

- Correctly identify all Apple-branded or MFi certified USB power adapters that connect resistor networks to USB D+ and USB D- (see Connectors That Do Not Implement iAP2 (page 485)).
- Correctly identify all USB Dedicated Charging Ports (DCP) as defined in the USB 2.0 specification.
- Enumerate as a USB device when connected to a USB host, such as a Mac, and:
  - Not draw more than 100 mA of current until they have been successfully enumerated.
  - Request 500 mA of charging current in their USB device descriptor.
- Not draw more power than the USB power source claims it is capable of providing via one of the above methods.

Accessories that draw power from a Lightning Receptacle (C37) to charge an internal battery must be capable of drawing at least 1.0 A from an Apple-branded or MFi-certified USB power source. 2.4 A is recommended.

Confidential Property of Foxlink
Foxlink@foxlink.com.tw

6.3.1 Pins and Assignments

Figure 6-2 Lightning Receptacle (C37) recommended PCB layout

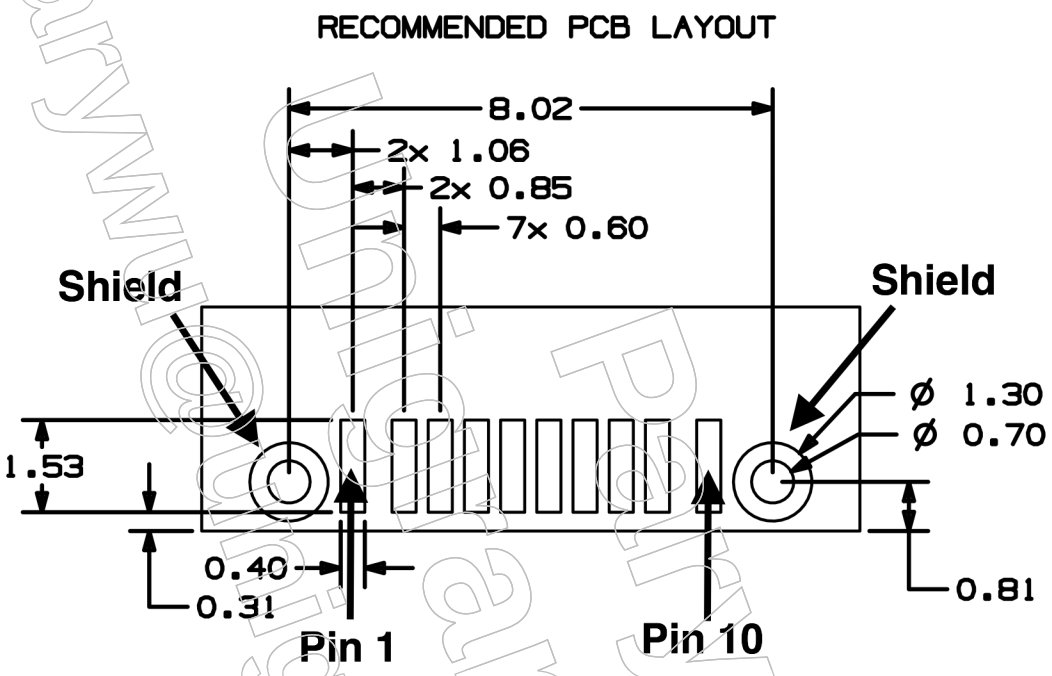


Table 6-1 Lightning Receptacle (C37) pins

Pin	Name	Assignment
1	RCP_1	see Pad Layout and Assignments (page 180)
2	RCP_2	Ground, see Pad Layout and Assignments (page 180)
3	RCP_3	see Pad Layout and Assignments (page 180)
4	RCP_4	see Pad Layout and Assignments (page 180)
5	RCP_5	see Pad Layout and Assignments (page 180)
6	RCP_6	Receptacle Power, see Pad Layout and Assignments (page 180)
7	RCP_7	see Pad Layout and Assignments (page 180)
8	RCP_8	see Pad Layout and Assignments (page 180)
9	RCP_9	see Pad Layout and Assignments (page 180)
10	RCP_10	Must be connected to RCP_1.

Preliminary Pinout and Layout Guidelines

The receptacle shield must be grounded.

Signals from the following pads must be routed as differential pairs:

- RCP_3 and RCP_4
- RCP_7 and RCP_8

6.4 Lightning Receptacle (C37) Example Integrations

Examples of accessories that may benefit from including Lightning Receptacle (C37) include docks, battery pack cases, game controllers, Lightning headphones, and Bluetooth speakers.

6.4.1 Lightning Audio Module Accessories

Accessories may combine the Lightning Audio Module (see Apple Lightning Audio Module (page 80)) with a Lightning Receptacle (C37) to enable passthrough/sync of an Apple device concurrently with iOS app communication between the device and accessory.

This can be useful for headphones (see Headphones (page 378)), smart cases (see Cases (page 336)), and battery power banks.

6.4.2 Wireless Accessories

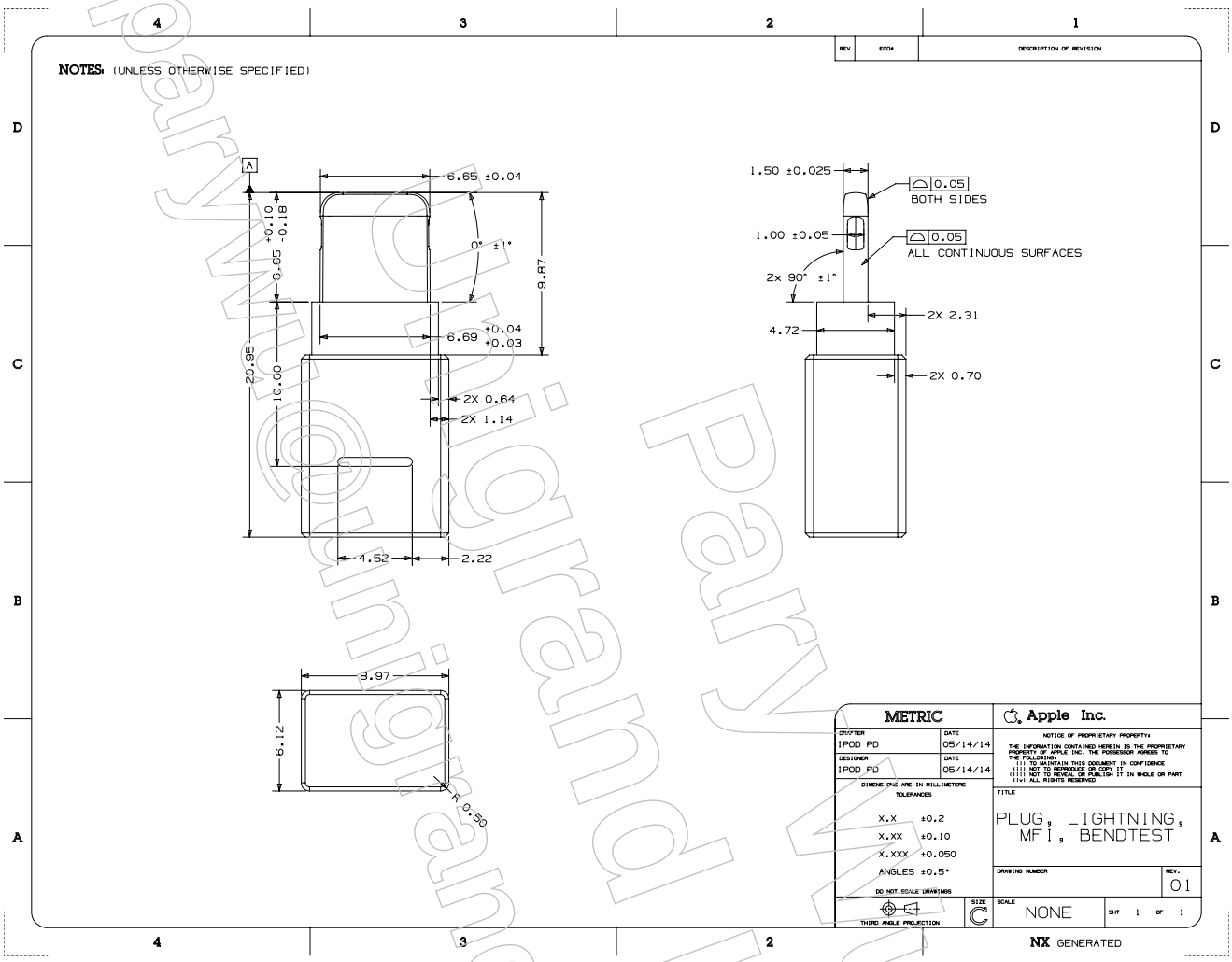
Wireless accessories may use a Lightning Receptacle (C37) to charge their internal batteries at higher rates than those permitted by the USB specification using any USB to Lightning cable and Apple branded or MFi certified USB power adapters. This can be useful for game controllers (see HID Game Controller (page 416)), headphones (see Headphones (page 378)), and speakers.

6.5 Test Procedures

These test procedures require the use of a test plug as specified in Figure 6-3 (page 178).

Apple Confidential
Unigrand LTD
ray.wu@unigrand.com.tw

Figure 6-3 Lightning Receptacle (C37) Mechanical Test Plug



6.5.1 Static Receptacle Opening

1. Verify that an Apple branded Lightning cable inserts fully into the receptacle.
2. Using a caliper,
   a. Verify that the width of the receptacle opening is at least 7.45 mm and at most 7.70 mm.
   b. Verify that the height of the receptacle opening is at least 1.70 mm and at most 1.95 mm.
   c. Verify that the depth of the receptacle to the surface of the opening is at least 6.47 mm and at most 6.65 mm.
3. Verity that the receptacle opening is fully rounded.

Party Original and Ltd
larry@original-and.com.tw

6.5.2 Receptacle Mounting Under Insertion Force

1. Place the accessory in a clamp or other mechanism to securely hold it in place.
2. Using a caliper, measure the depth of the receptacle to the surface of the opening.
3. Fully insert the mechanical test plug into the receptacle.
4. Using a calibrated force gauge, apply 30 N of pressure to the mechanical test plug in the direction of plug insertion via the measuring tip of the force gauge.
5. Verity that the receptacle does not break away from the surface of the accessory.
6. Verify that the depth of the receptacle to the surface of the opening has not changed.

6.5.3 Receptacle Mounting Under Extraction Force

1. Place the accessory in a clamp or other mechanism to securely hold it in place.
2. Fully insert the mechanical test plug into the receptacle.
3. Using a calibrated force gauge, apply 30 N of pressure to the connector in the direction of plug extraction by clamping on to the plug and pulling on it.
4. Verify that the receptacle does not pull or break away from the accessory.

6.5.4 Bend Test

1. Place the accessory in a clamp or other mechanism to securely hold it in place.
2. Fully insert the mechanical test plug into the receptacle.
3. Using a calibrated force gauge, apply 100 N of pressure to the side of the mechanical test plug with the edge of the force gauge at a distance of 10 mm from the accessory surface.
4. Rotate the accessory +/- 90 degrees, two times each, and apply the same force from each position.
5. Perform a continuity test on all receptacle contacts and verify that continuity is maintained and there are no open circuits.

Confidential Property of Foxconn Interconnect Technology Limited

7. Apple Lightning Receptacle Controller

Note: This chapter is a Developer Preview and is not intended for use in the development of Proposed Products or Licensed Products under a MFi License. Although this content has been reviewed for accuracy, it is not final. Apple is supplying this content to help accessory developers plan for the adoption of the accessory interface features described herein. This information is subject to change, and accessories implemented according to this content must be tested with final operating system software and final documentation before going through self certification.

Accessories may incorporate a Lightning Receptacle Controller to establish electrical connections with Apple branded or MFi certified power supplies. The Lightning Receptacle Controller may also be used to enable pass-through Apple device charge/sync.

The Apple Lightning Receptacle Controller chip must be used in conjunction with the Lightning Receptacle (C37), see Apple Lightning Receptacle (page 172).

7.1 Mechanical

The Lightning Receptacle Controller has the following mechanical properties:

- Package: WLCSP 36 balls, 2.07 mm x 2.07 mm x 0.435 mm height, 0.35 mm pitch
- Operating temperature: -20 degrees C to 85 degrees C

PCB design rules:

- For this 0.35 mm BGA, a UBM size of 0.2 mm with a max solder ball size of 0.21 mm is recommended.
- On the PCB side, a copper defined pad of size 0.2 mm is recommended.

7.2 Pad Layout and Assignments

This section details the layout, names, and assignments of the Lightning Receptacle Controller BGA pads. All un-assigned or Reserved signals must be soldered but left Not Connected (NC).

Property of Foxconn Interconnect Technology Limited

Figure 7-1 Lightning Receptacle Controller pad layout (top view)

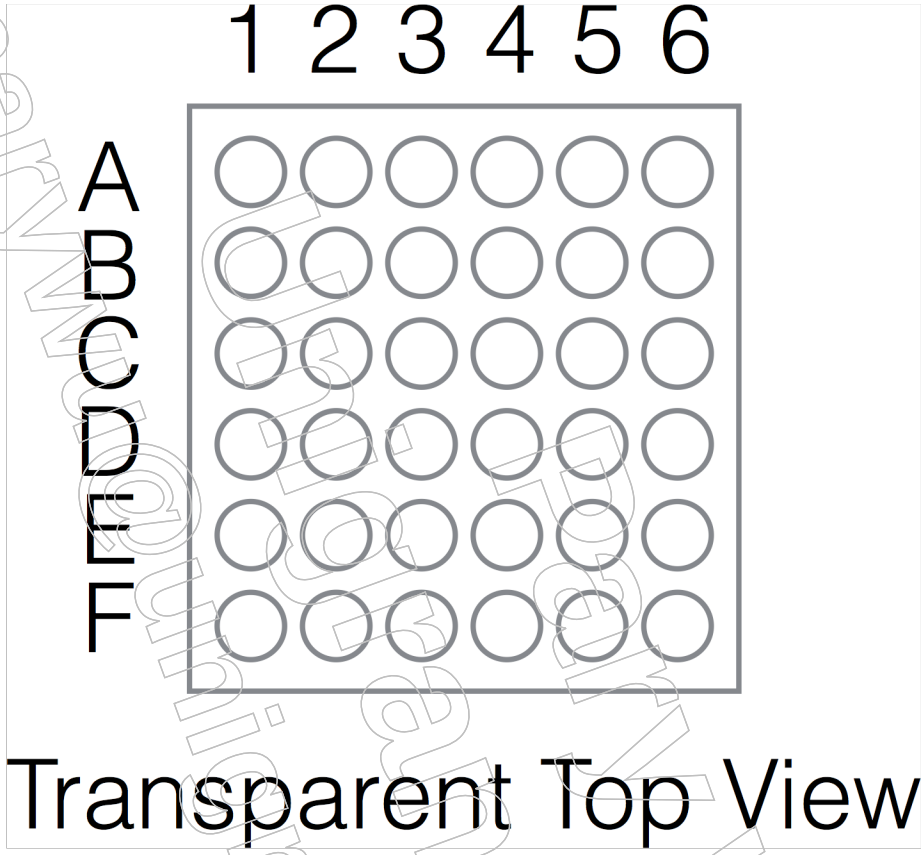


Table 7-1 Lightning Receptacle Controller pad assignments

Pad	Assignment
A1	Reserved
A2	RCP_3, see Pins and Assignments (page 176)
A3	USB D+, see Reference Circuit (page 184)
A4	RCP_8, see Pins and Assignments (page 176)
A5	Reserved
A6	Ground
B1	Reserved
B2	RCP_4, see Pins and Assignments (page 176)
B3	USB D-, see Reference Circuit (page 184)

paryweu@unionpacific.com.tw
Union Pacific Industrial LTD

Pad	Assignment
B4	RCP_7, see Pins and Assignments (page 176)
B5	Reserved
B6	Reserved
C1	Ground
C2	Reserved
C3	Reserved
C4	Reserved
C5	RCP_5, see Pins and Assignments (page 176)
C6	Reserved
D1	Reserved
D2	Reserved
D3	10 kΩ pull-up to V_DD(1V8), see Reference Circuit (page 184)
D4	10 kΩ pull-up to V_DD(1V8), see Reference Circuit (page 184)
D5	Reserved
D6	Power Gate Enable (Low) with 10 kΩ pull-up to Receptacle Power, see Reference Circuit (page 184)
E1	Reserved
E2	Reserved
E3	RCP_1 and RCP_10 with 220 kΩ pull-up to V_DD(3V3) and 10 kΩ in series, see Pins and Assignments (page 176)
E4	10 kΩ pull-up to V_DD(1V8), see Reference Circuit (page 184)
E5	RCP_9, see Pins and Assignments (page 176)
E6	1.0 uF to Ground, see Reference Circuit (page 184)
F1	Reserved

Paywall Unlocked LTD
Paywallunlocked.com

Pad	Assignment
F2	Reserved
F3	V_DD(1V8), see Reference Circuit (page 184)
F4	V_DD(3V3), see Reference Circuit (page 184)
F5	Ground
F6	Receptacle Power, see Pins and Assignments (page 176)

Signals from D3, D4, and E4 must be combined and pulled-up to V_DD(1V8) using a 10 kΩ resistor.

7.3 Electrical

The Lightning Receptacle Controller has the following electrical properties:

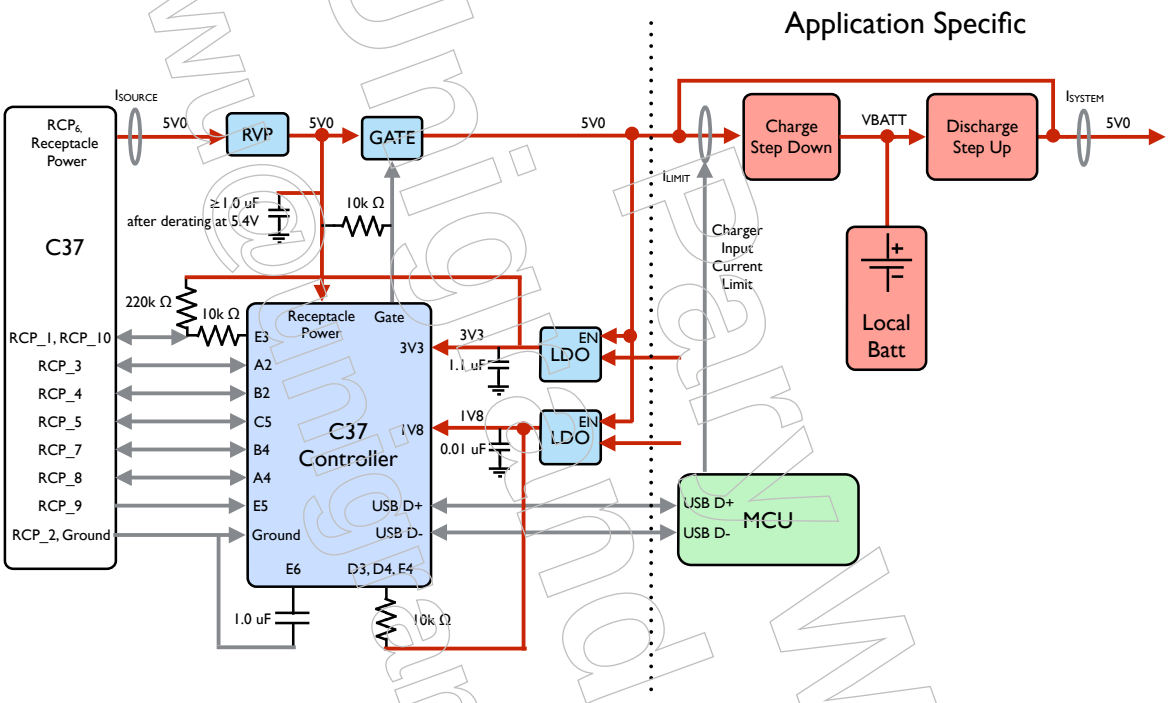
- Power supplies: V_DD(1V8) and V_DD(3V3)
- ESD 1500 V HBM, 500 V CDM

V_DD(3V0) may be used in place of V_DD(3V3).

Pary Wu
Unigrand LTD
paryw@unigrand.com.tw

7.3.1 Reference Circuit

Figure 7-2 Lightning Receptacle Controller reference circuit



1. MCU identifies max source current, I_{SOURCE}, via USB D+/D-.
2. Design ensures $I_{LIMIT} \leq I_{SOURCE} - I_{SYSTEM}$

The capacitor connected to Receptacle Power at F6 must be at least 1.0 uF after derating at 5.4 V.

The two LDOs must only be enabled when Receptacle Power is 5 V. The LDO power source may come from Receptacle Power or System Power.

7.3.2 Input/Output Electrical Characteristics

The following electrical characteristics apply to the Lightning Receptacle Controller.

TBD

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

8. Apple 30-pin Connector

These connectors must not be the sole or main point of physical support for an iPod, iPhone, or iPad. Accessories must use the universal dock & dock adapter approach to provide support, or, alternatively, must implement another means (other than the connector) to provide physical support to an iPod, iPhone, or iPad.

See *MFi Accessory Hardware Specification R9* for full details of the 30-pin connector.

8.1 Connector Versions

8.1.1 Mid-Plane Plug

This connector should be used with products that require a metal shell around the connector (for shielding) or with products that do not require latching pins (friction or detent). This plug is designed so that the pins on the connector will straddle both sides of a PCB and is not designed for surface mount solutions. It is commonly used with products that have electronics right in the plug that plugs into the iPod.

The mid-plane plug is designed for use with the Mid-Plane Plug (Top Shell) (page 186) & Mid-Plane Plug (Bottom Shell) (page 186), which provide a passive/friction latching mechanism.

*Note: Only 16 pins are present in this connector. See the connector drawing for details. Licensees are **strongly** encouraged to purchase samples of this connector to confirm that it meets their requirements.*

See Figure 8-1 (page 189) for a dimensional drawing.

8.1.2 Mid-Plane Plug with Friction Latches

This connector is designed so that the pins on the connector will straddle both sides of a PCB. It is commonly used with products that have electronics within the connector's housing.

This connector should be used with products where a metal shell around the connector is not desired or supported, but does require friction latching pins.

See Figure 8-2 (page 190) for a dimensional drawing.

Paul Wang Guangzhou Pinsheng Electronics Co., LTD

8.1.3 Mid-Plane Plug with Friction Latches, Power Only

This connector is designed so that the pins on the connector will straddle both sides of a PCB.

This connector should be used with products where a metal shell around the connector is not desired or supported, but does require friction latching pins.

Note: Only 10 pins are present in this connector. See the connector drawing for details.

See Figure 8-3 (page 191) for a dimensional drawing.

8.1.4 Mid-Plane Plug (Top Shell)

This component is used in combination with Mid-Plane Plug (Bottom Shell) (page 186). It includes latch 'fingers' that hold the connector into the iPod. Users are not required to press detent buttons on the side of the connector to disconnect the accessory.

This is not a plastic over-mold or shell, and helps shield against EMI.

See Figure 8-4 (page 192) for a dimensional drawing.

8.1.5 Mid-Plane Plug (Bottom Shell)

Used in combination with Mid-Plane Plug (Top Shell) (page 186).

See Figure 8-5 (page 193) for a dimensional drawing.

8.1.6 Cable-End Plug

This connector is designed to be soldered to the end of a cable. It should be used with cable assemblies, where the connector cannot be unplugged unless the user presses release buttons on the side of the connector housing. This connector has solder tail pins, and is not suitable for mounting onto a PCB.

The cable-end plug is designed for use with Cable-End Plug (Top Shell) (page 187) & Cable-End Plug (Bottom Shell) (page 187), which provide an active latch mechanism.

See Figure 8-6 (page 194) for a dimensional drawing.

8.1.7 Cable-End Plug, Power Only

This version of the cable-end connector is modified for use in 'Power & Sync Only' products.

Confidential
Property of Apple Inc.
Larry Wu
larry@unionchip.com.tw

This connector is designed to be soldered to the end of a cable. It should be used with cable assemblies, where the connector cannot be unplugged unless the user presses release buttons on the side of the connector housing. This connector has solder tail pins, and is not suitable for mounting onto a PCB.

Designed for use with Cable-End Plug (Top Shell) (page 187) & Cable-End Plug (Bottom Shell) (page 187), which provide an active latch mechanism.

Note: Only 10 pins are present in this connector. See the connector drawing for details.

See Figure 8-7 (page 195) for a dimensional drawing.

8.1.8 Cable-End Plug (Top Shell)

This component is used in combination with Cable-End Plug (Bottom Shell) (page 187)

It includes latch 'fingers' that hold the connector into the iPod. Users are required to press detent buttons on the side of the connector to disconnect the accessory.

This is not a plastic over-mold or shell, and helps shield against EMI.

See Figure 8-8 (page 196) for a dimensional drawing.

8.1.9 Cable-End Plug (Bottom Shell)

This component is used in combination with Cable-End Plug (Top Shell) (page 187)

See Figure 8-9 (page 197) for a dimensional drawing.

8.1.10 Vertical Dock Plug

The vertical dock is used in products where an Apple device will sit vertically (perpendicular to the PCB), such as battery packs.

See Figure 8-10 (page 198) for a dimensional drawing.

8.1.11 Angled Dock Plug

The angled dock is tilted 15 degrees off vertical, with a support tab. It is designed for use with plastic sleeve and adapters used with the universal dock.

See Figure 8-11 (page 199) for a dimensional drawing.

Authorized Third Party Wu Qiang Electronic Technology Co. LTD
Authorized@ifunbar.com.tw

8.1.12 Receptacle

The receptacle connector is only available in sample quantities, and is only for use in manufacturing and R&D test fixtures.

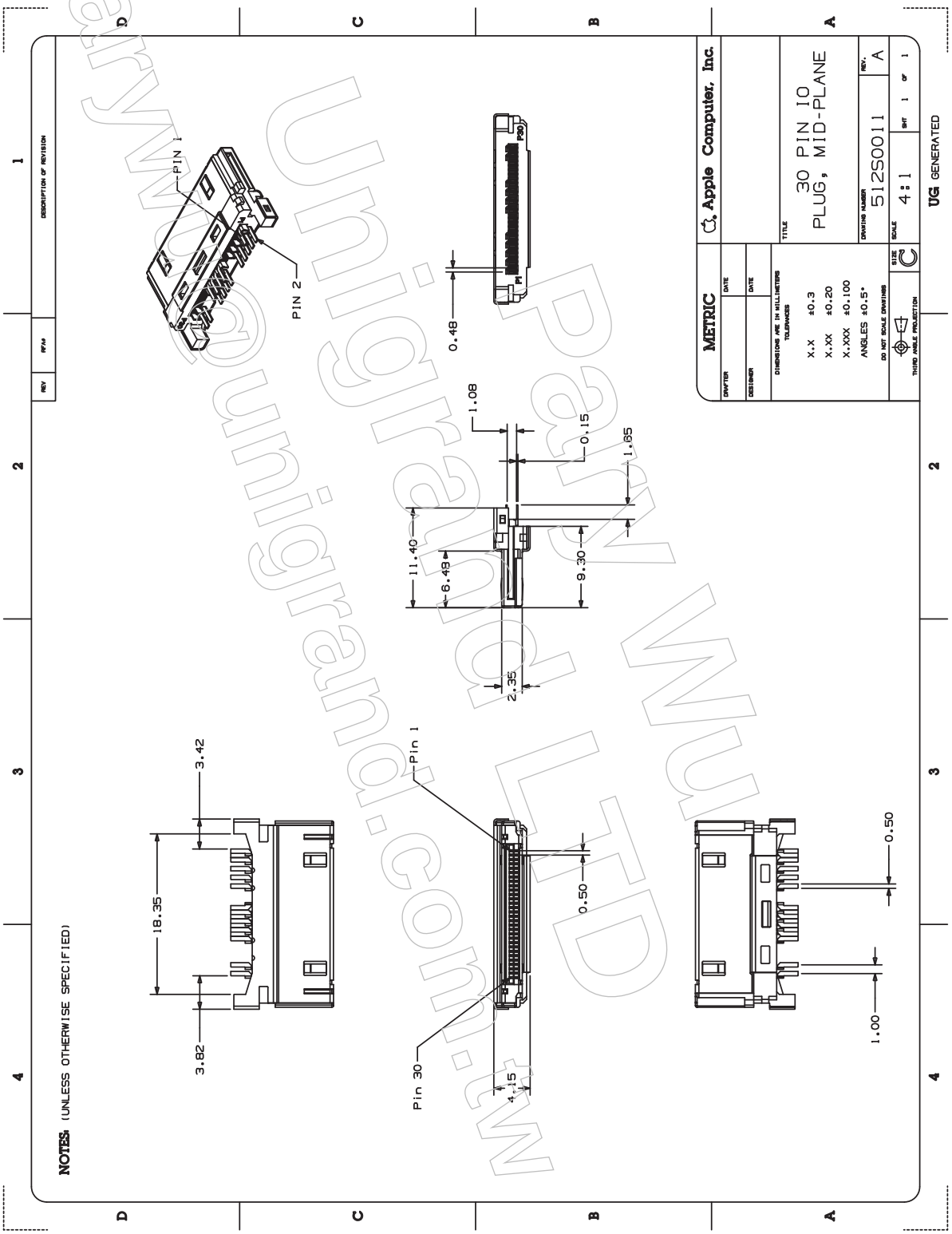
Note: This connector is not available for use in production and/or shipping products.

See Figure 8-12 (page 200) for a dimensional drawing.

Pary Wu
Unigrand LTD
pary.wu@unigrand.com.tw

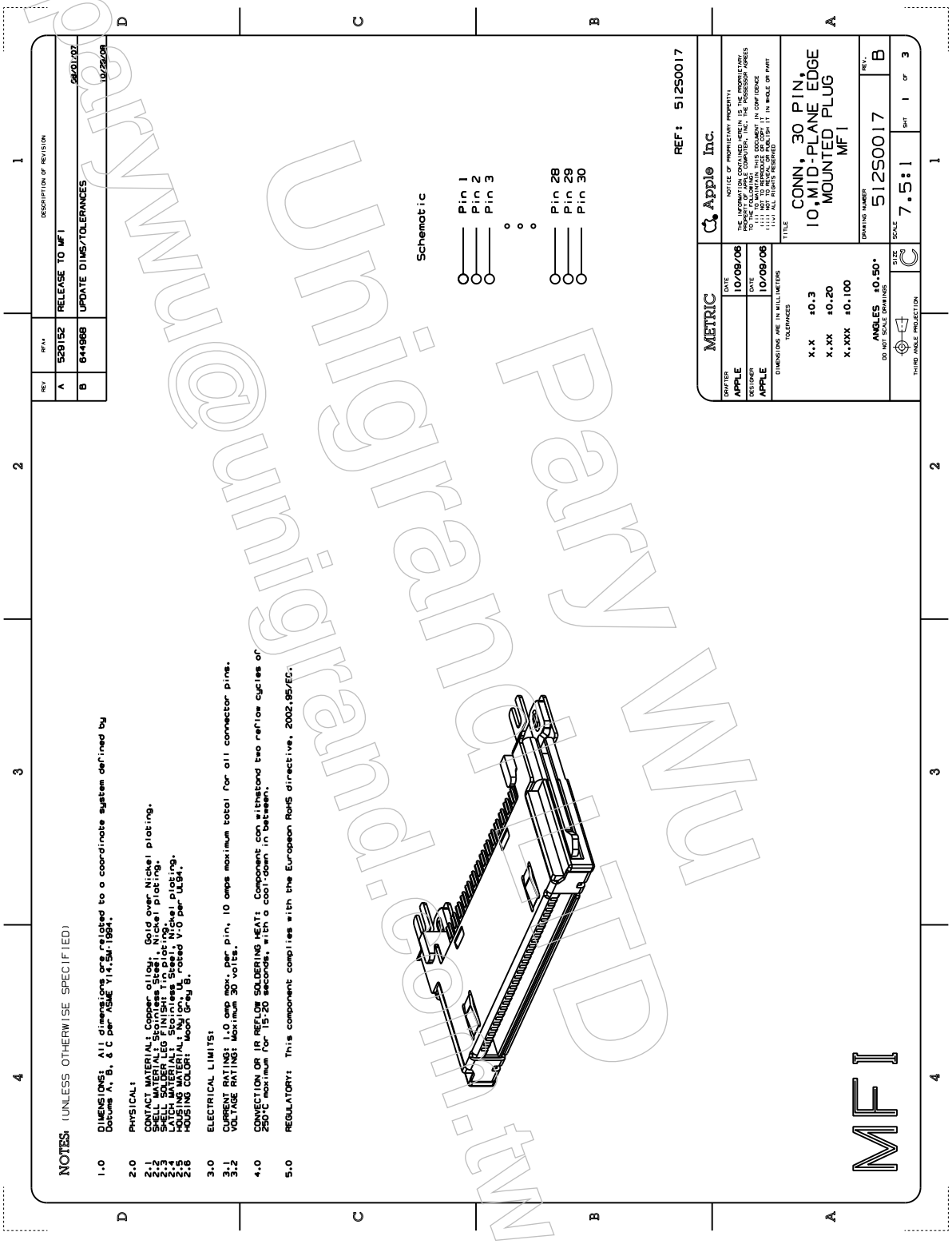
8.2 Connector Dimensional Drawings

Figure 8-1 Mid-Plane Plug



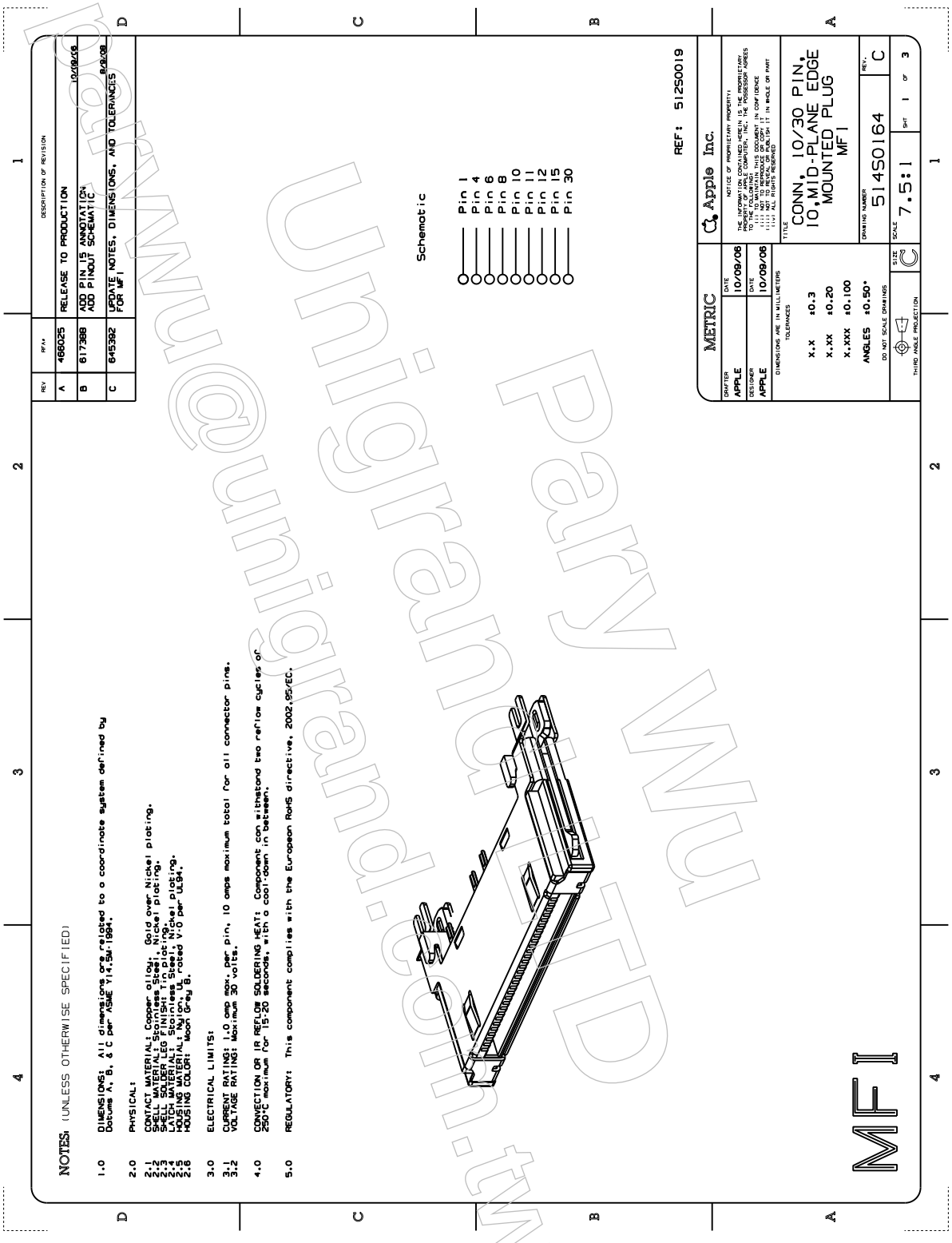
Party Wu LTD
Unigrand.com.tw

Figure 8-2 Mid-Plane Plug, with Friction Latches



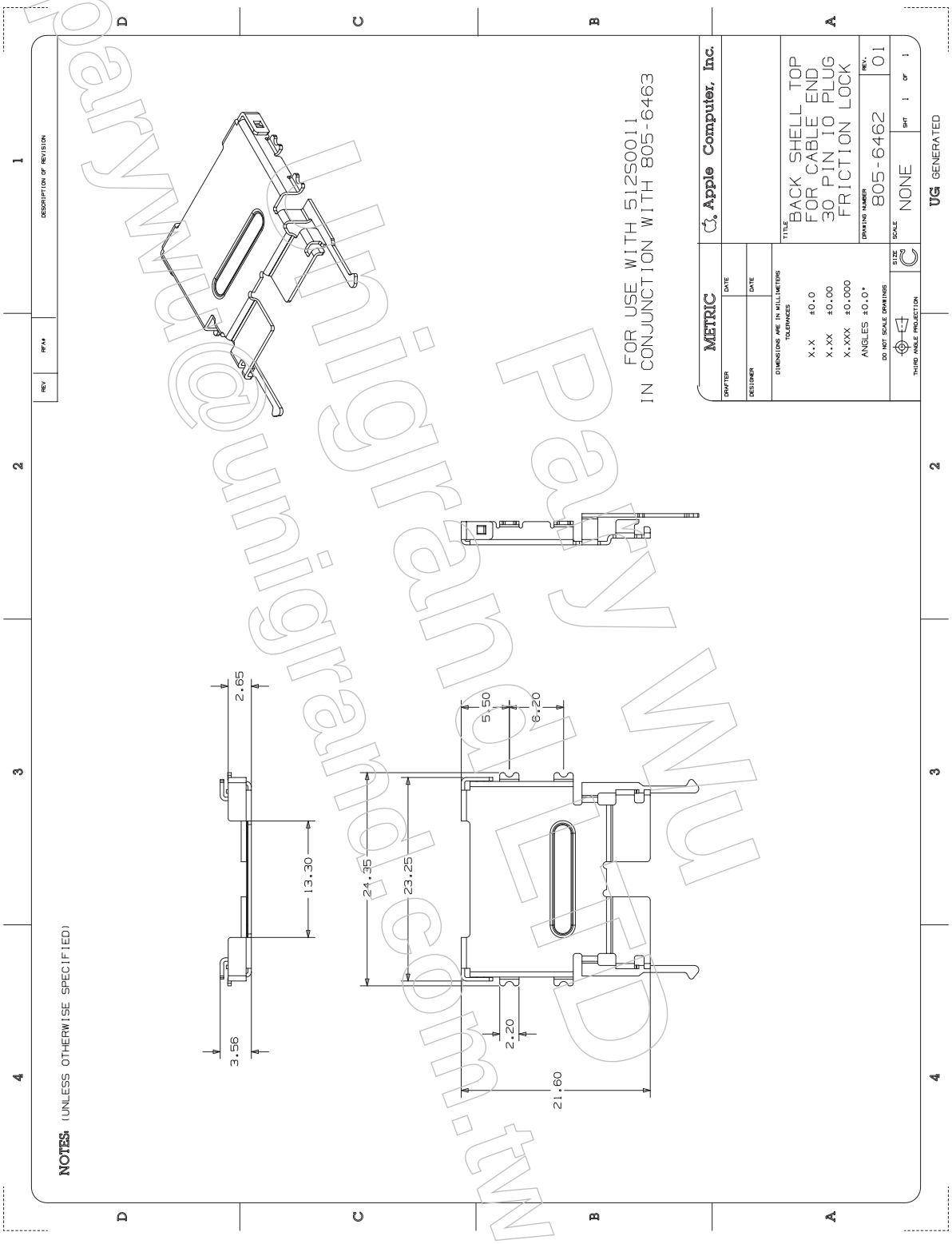
Pary Wu
Unigrand LTD
pary.wu@unigrand.com.tw

Figure 8-3 Mid-Plane Plug with Friction Latches, Power Only



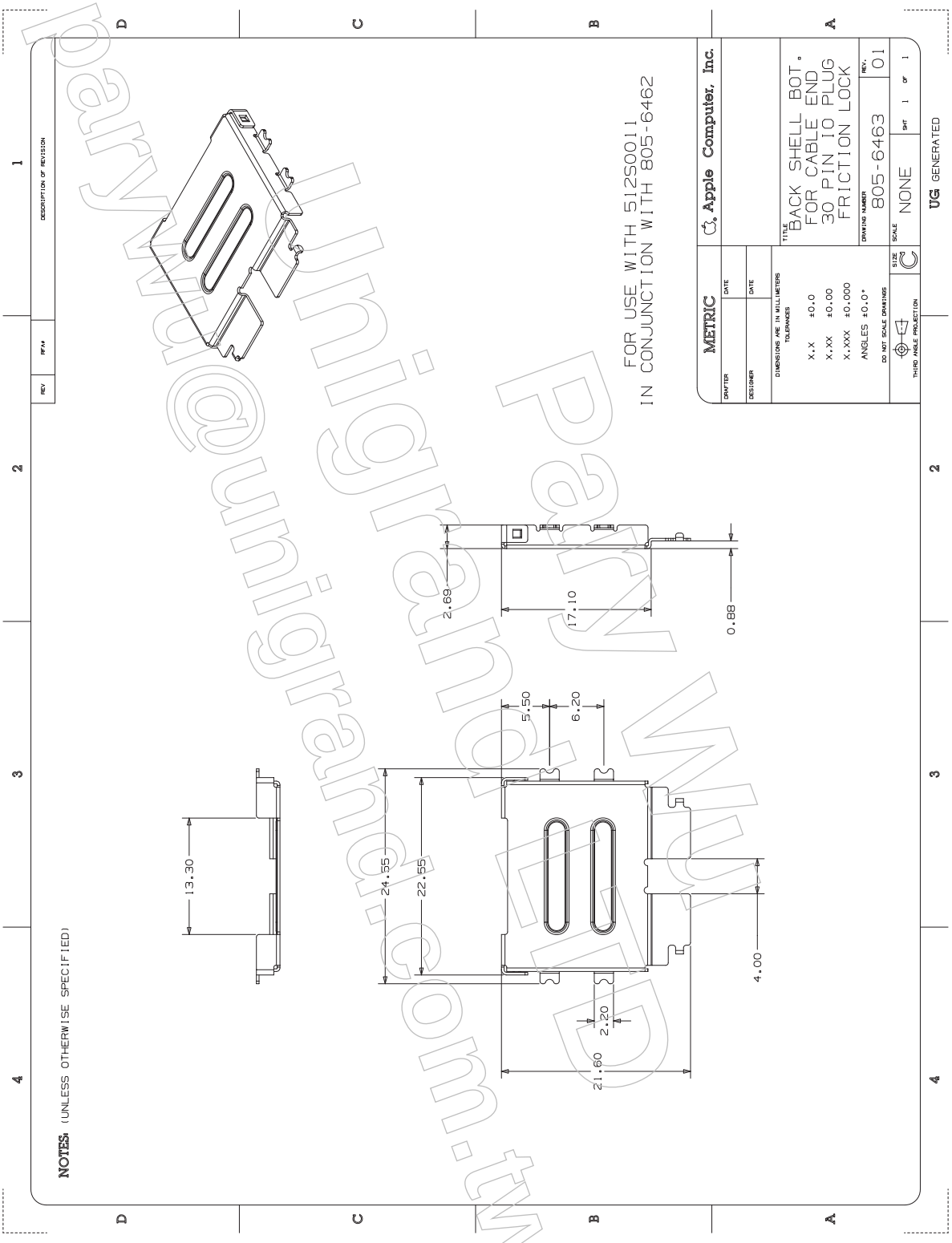
2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 8-4 Top Shell for Mid-Plane Plugs



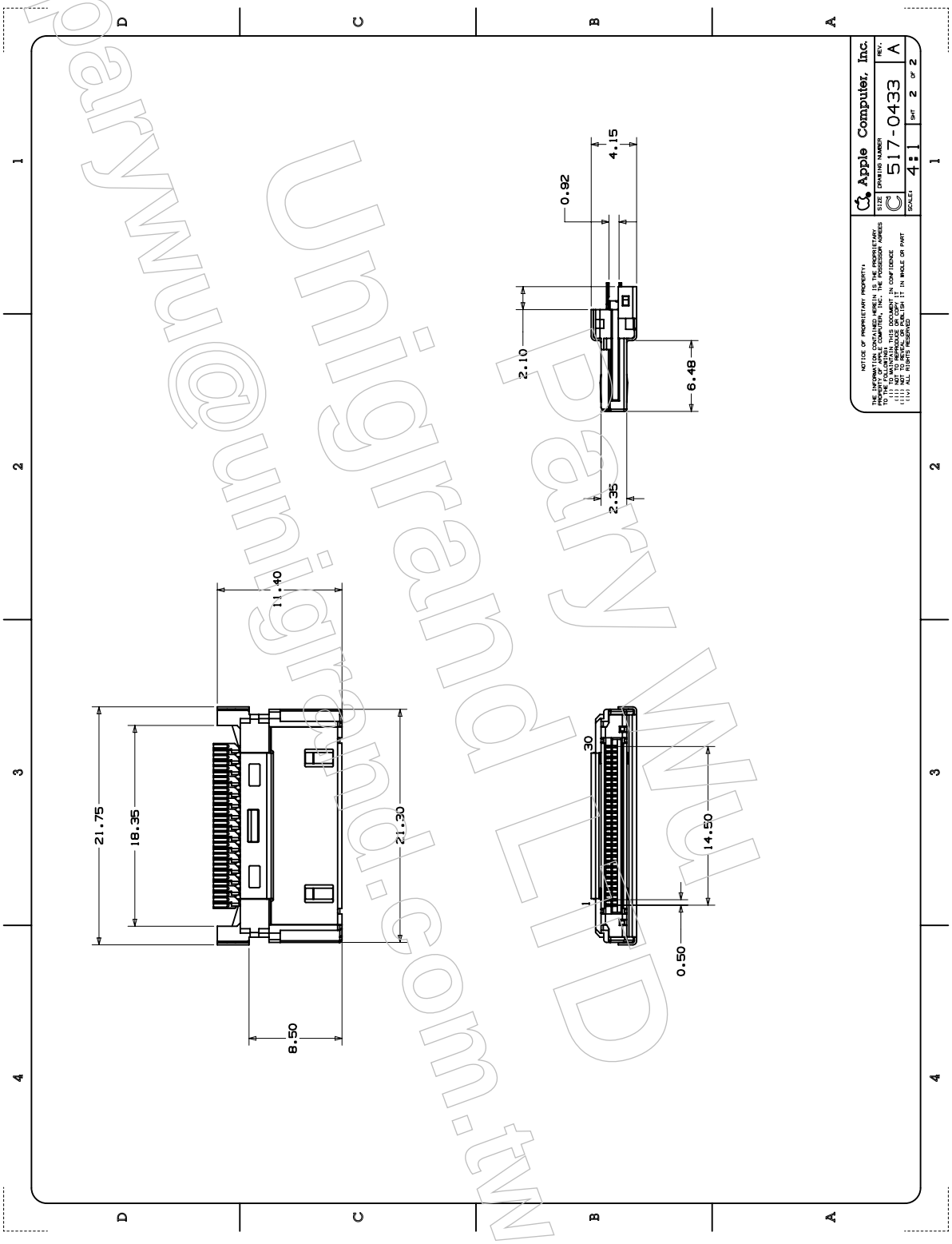
Pay Wu Unigrand LTD
Unigrandtech.com.tw

Figure 8-5 Bottom Shell for Mid-Plane Plugs



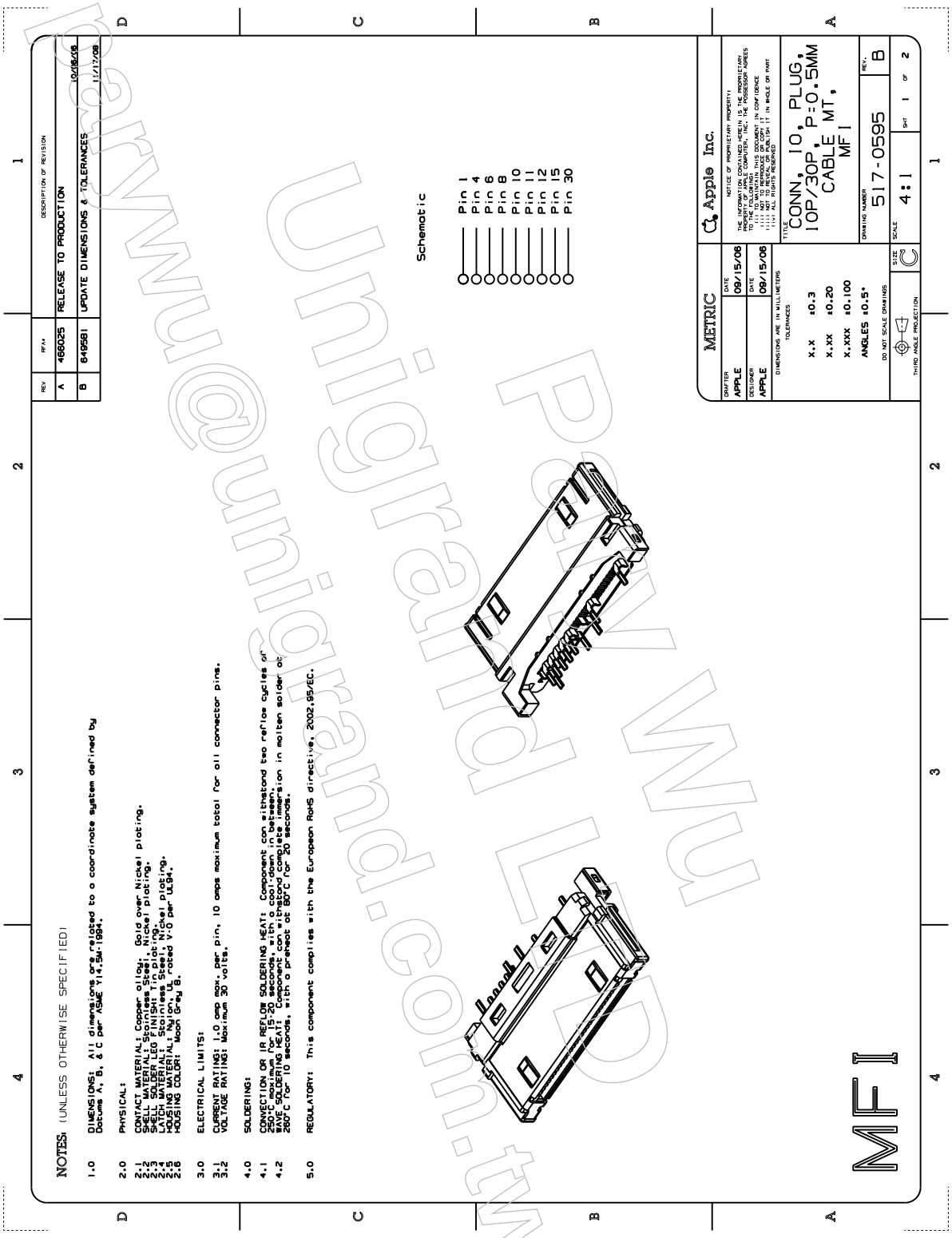
partner@unigrand.com.tw Partner Unigrand LTD

Figure 8-6 Cable-End Plug



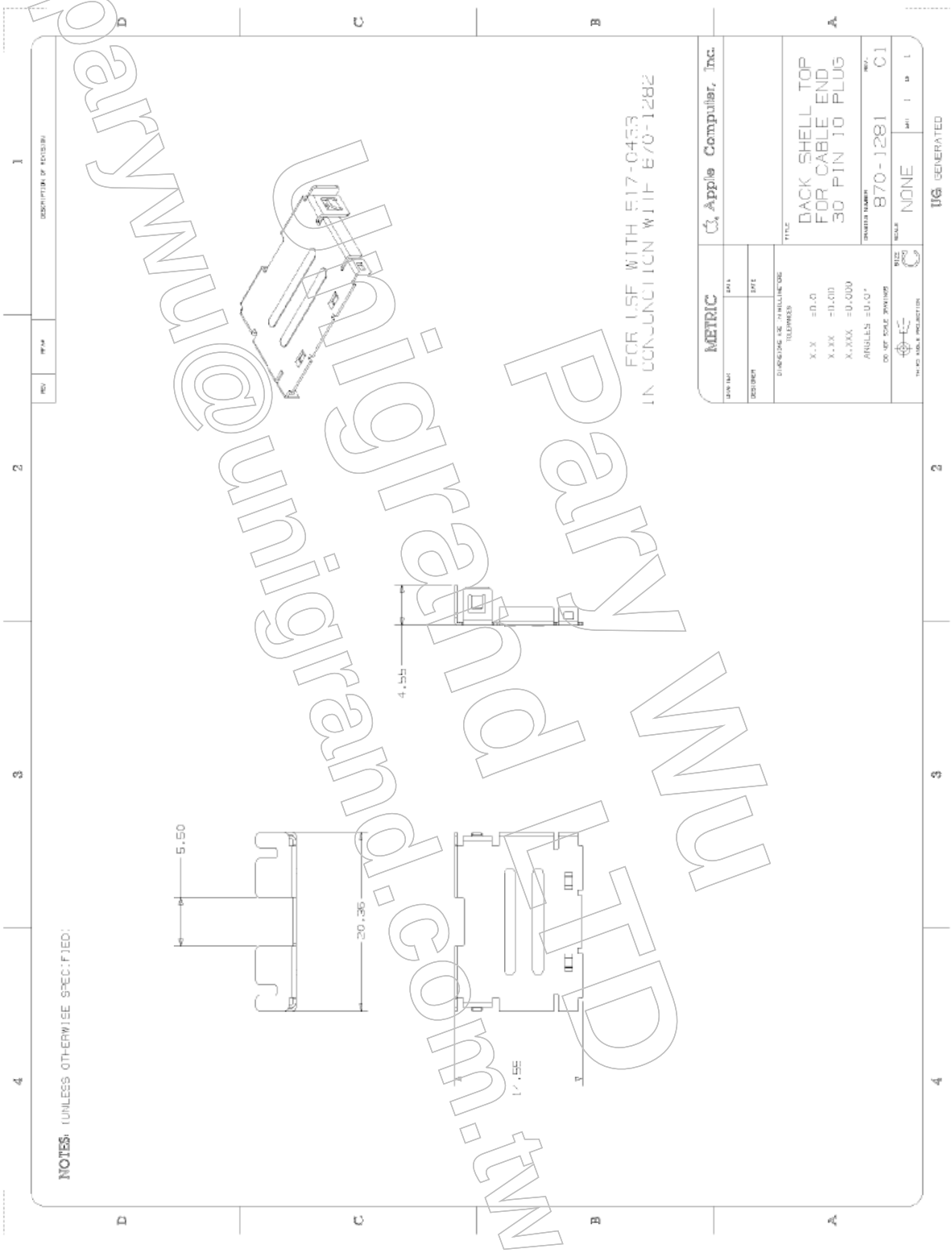
Party@unigrand.com.tw Unigrand Ltd

Figure 8-7 Cable-End Plug, Power Only



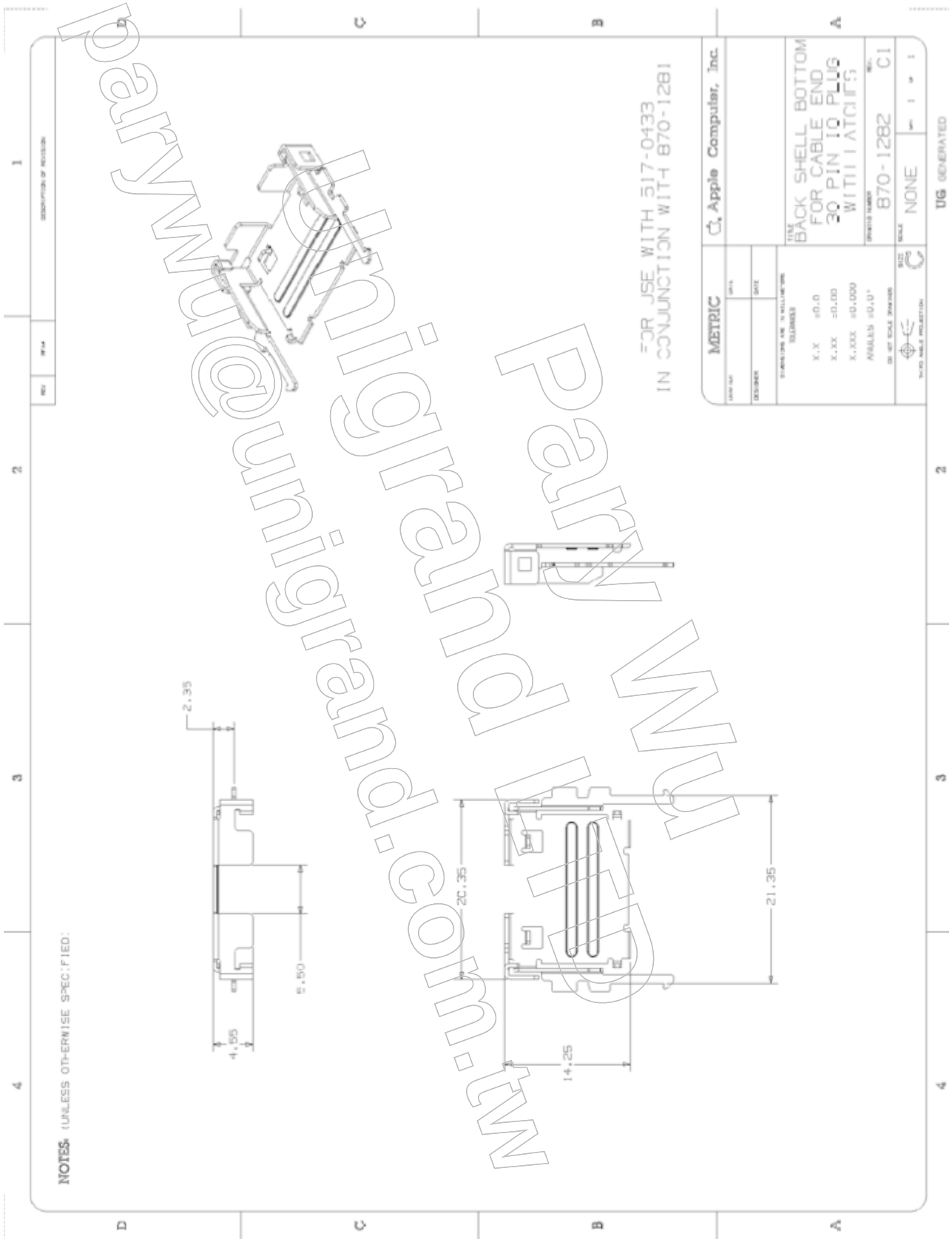
Perry Wu Unigrand LTD perrywu@unigrand.com.tw

Figure 8-8 Top Shell for Cable End Plugs



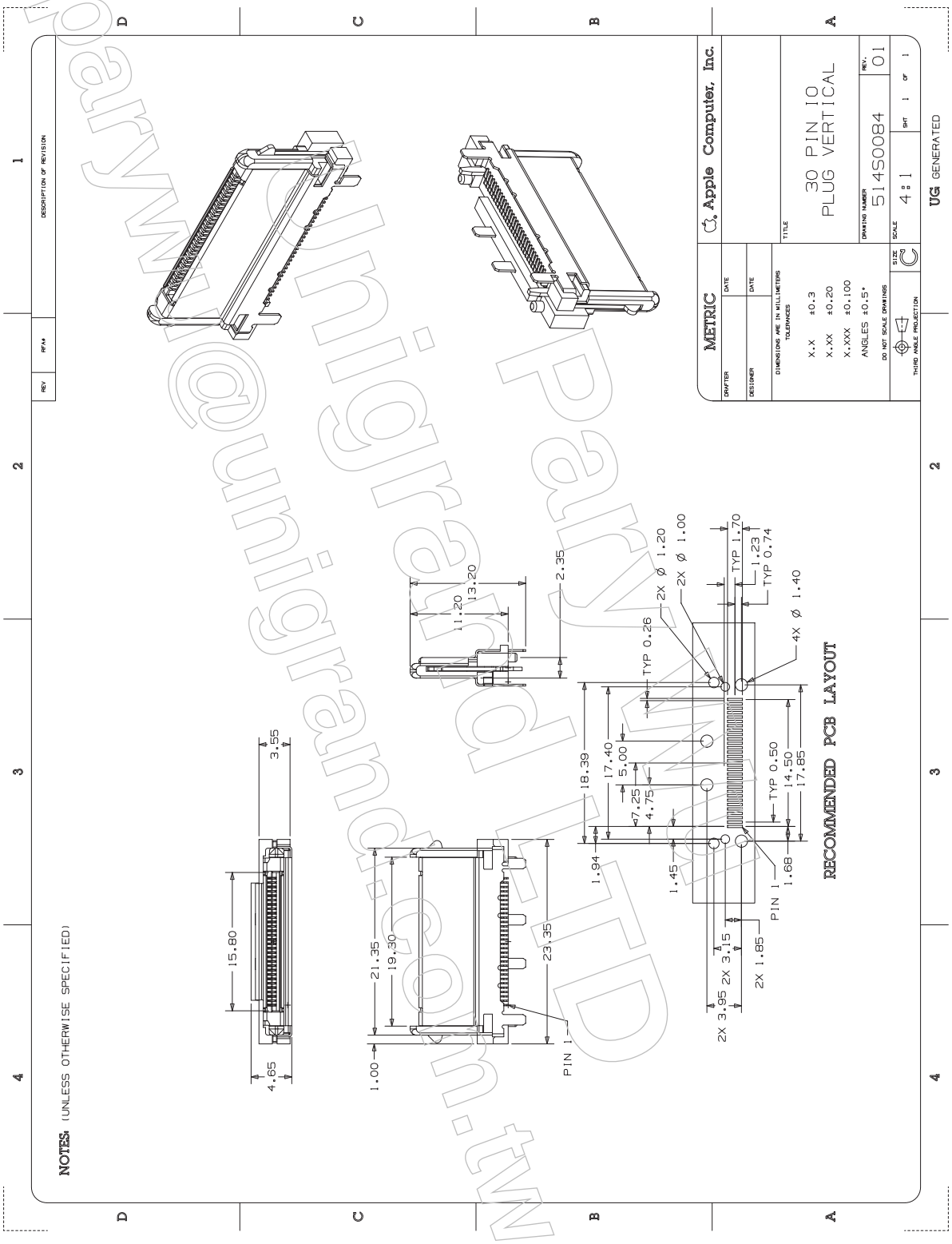
partywugunigrand.com.tw
Party Wu
Unigrand LTD

Figure 8-9 Bottom Shell for Cable End Plugs



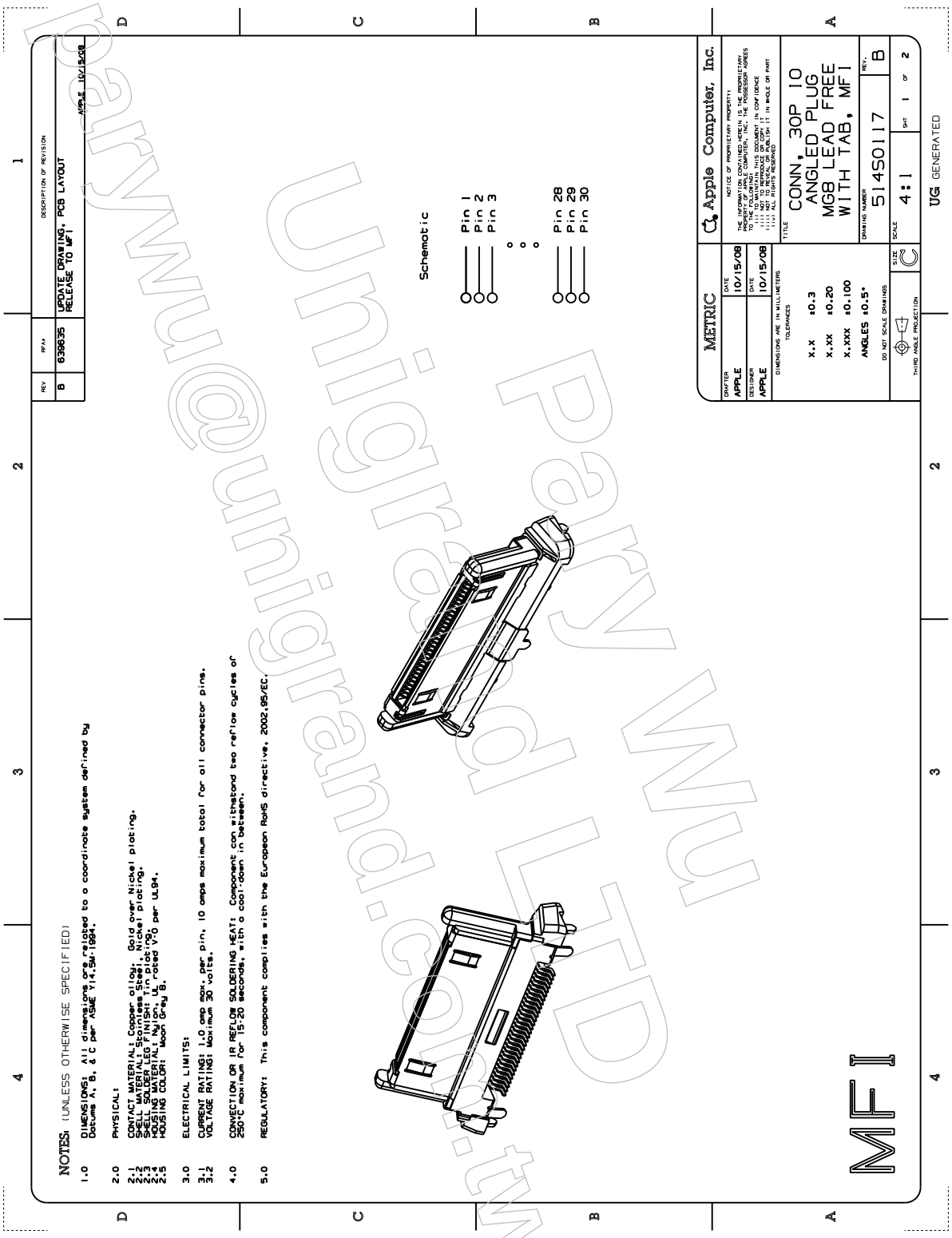
Party Wu
partywu@unigrand.com.tw
Unigrand Ltd.

Figure 8-10 Vertical Dock Plug



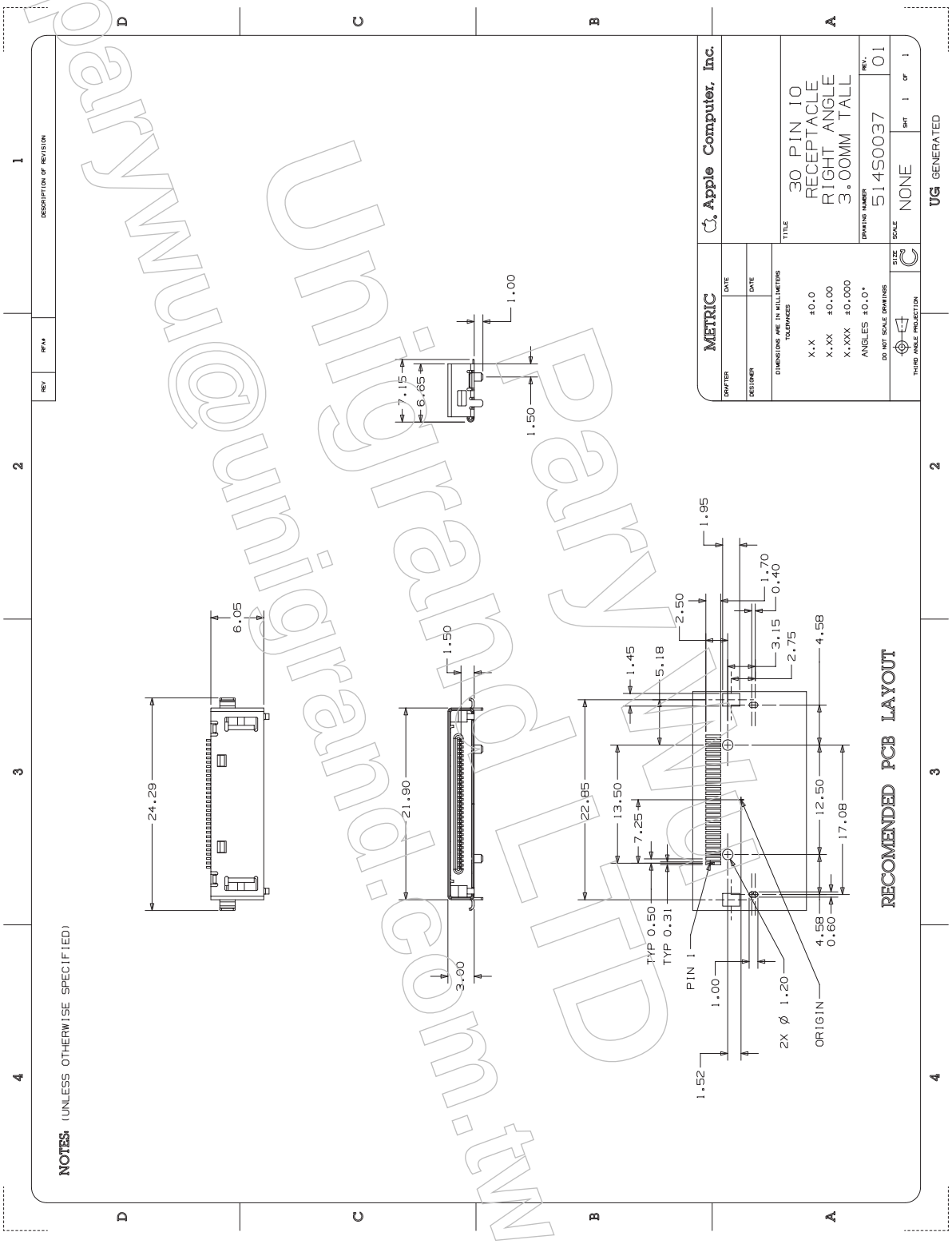
2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 8-11 Angled Dock Plug



Party Wu Unigrand LTD
Party@unigrand.com

Figure 8-12 Receptacle



Paryww@unigrandltd.com.tw
Unigrand LTD

9. Accessory Authentication

Accessory authentication provides a mechanism for Apple devices to trust the identities and feature sets of accessories that are compliant with this specification.

It is also possible for an accessory to require the Apple device to authenticate itself to the accessory. For more details, see Device Authentication (page 362).

9.1 Accessory Authentication Requirements

The accessory hardware must incorporate an Apple Authentication Coprocessor, version 2.0B or 2.0C. Version 2.0C is strongly recommended and specified in Apple Authentication Coprocessor 2.0C (page 54). Earlier versions of the authentication coprocessor are not allowed.

An accessory may use one authentication processor to authenticate itself to more than one Apple device. Conversely, multiple accessories must not share one authentication coprocessor or authenticate on behalf of each other to the same Apple device.

9.1.1 Accessory Authentication over iAP2 Requirements

The accessory must have successfully established an iAP2 link over an iAP2 transport.

All accessories that support the Accessory Authentication feature via iAP2 must send or receive the following iAP2 control session message(s):

- RequestAuthenticationCertificate (page 577)
- AuthenticationCertificate (page 577)
- RequestAuthenticationChallengeResponse (page 577)
- AuthenticationResponse (page 578)
- AuthenticationFailed (page 578)

Confidential Property of FIH Mobile Ltd. Partner Use Only

9.2 Accessory Authentication Usage

9.2.1 Accessory Authentication over iAP2 Usage

Authentication over iAP2 is always initiated with the transmission of a RequestAuthenticationCertificate (page 577) message from the Apple device to the accessory. After the RequestAuthenticationCertificate (page 577) message is received, the accessory must retrieve the X.509 certificate from its Apple authentication coprocessor and transmit it to the Apple device using an AuthenticationCertificate (page 577) message. The certificate must be sent within 1 second of receiving the RequestAuthenticationCertificate (page 577).

If the certificate is valid, the Apple device will respond with a RequestAuthenticationChallengeResponse (page 577) message. The accessory then must use the Apple authentication coprocessor to compute a response to the authentication challenge and send an AuthenticationResponse (page 578) message with the response. This response must be sent within 2 seconds of receiving the RequestAuthenticationChallengeResponse (page 577) message.

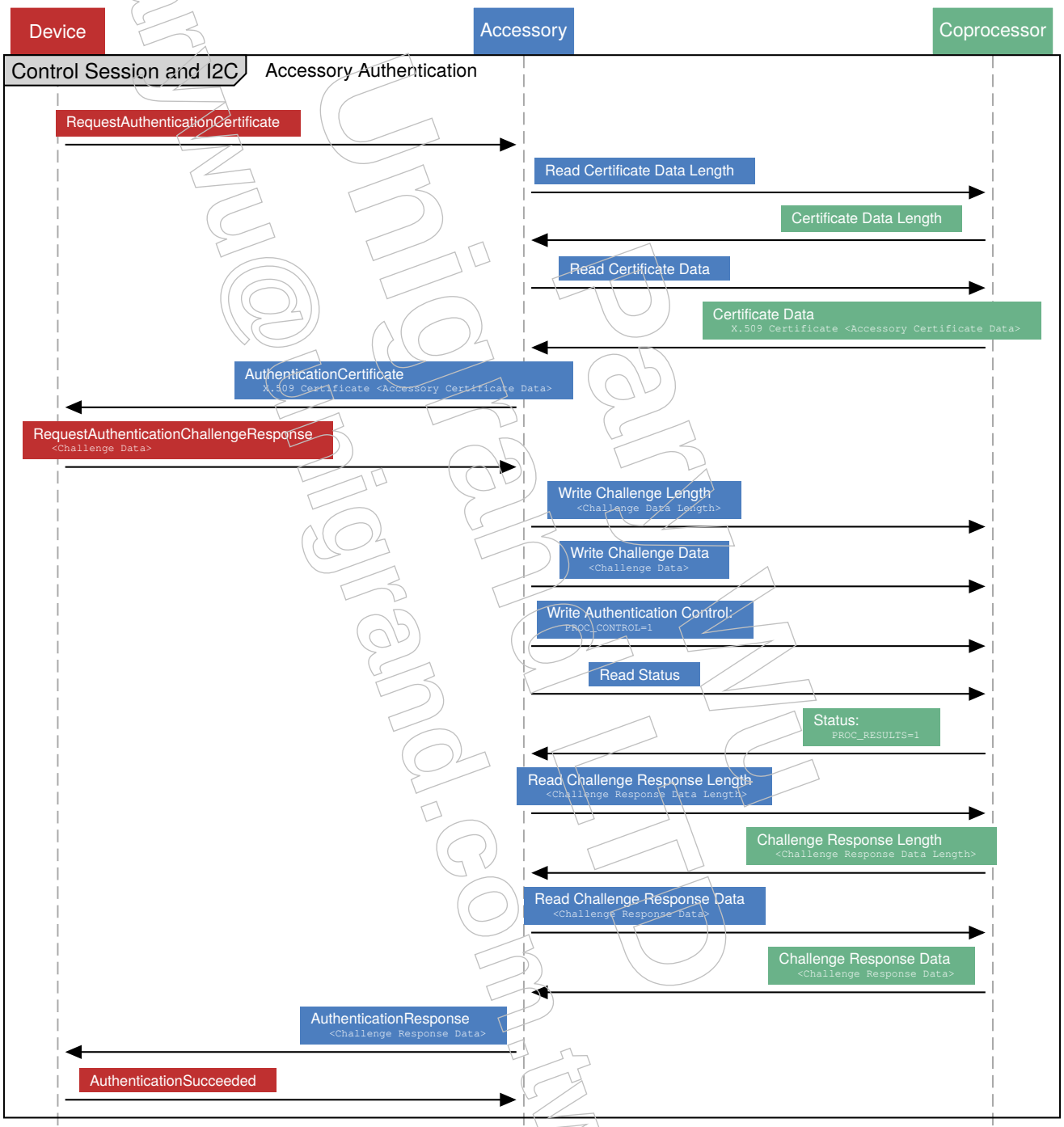
If the authentication response is correct, then the Apple device will send an AuthenticationSucceeded (page 578) message. The accessory must immediately proceed to Accessory Identification (page 205) unless it is using control session version 2 and has already identified (see Control Session (page 562)). With control session version 2, once the Apple device starts authentication, the accessory must complete the process within 15 seconds.

To demonstrate correct handling of the AuthenticationFailed (page 578) message during self-certification, accessories must send an empty or invalid certificate when the Apple Authentication Coprocessor is not present.

Royal Global LTD
royal@global.com.tw

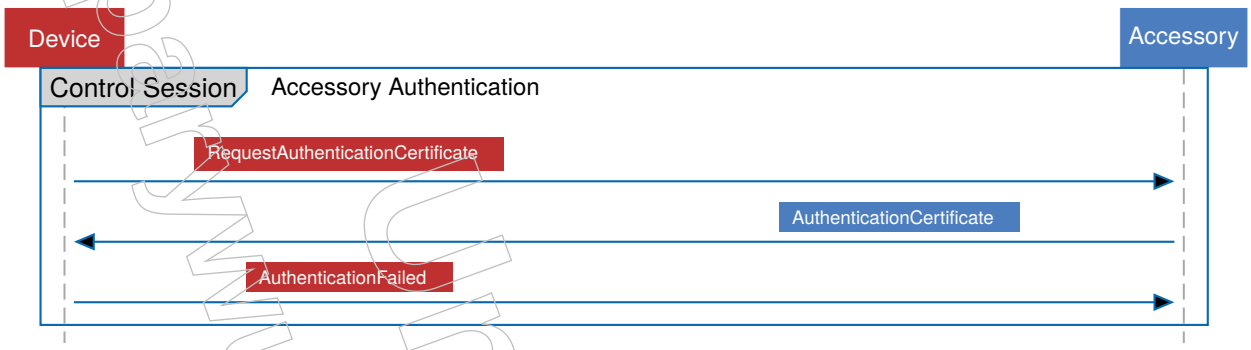
9.3 Accessory Authentication Examples

9.3.1 Typical Accessory Authentication over iAP2

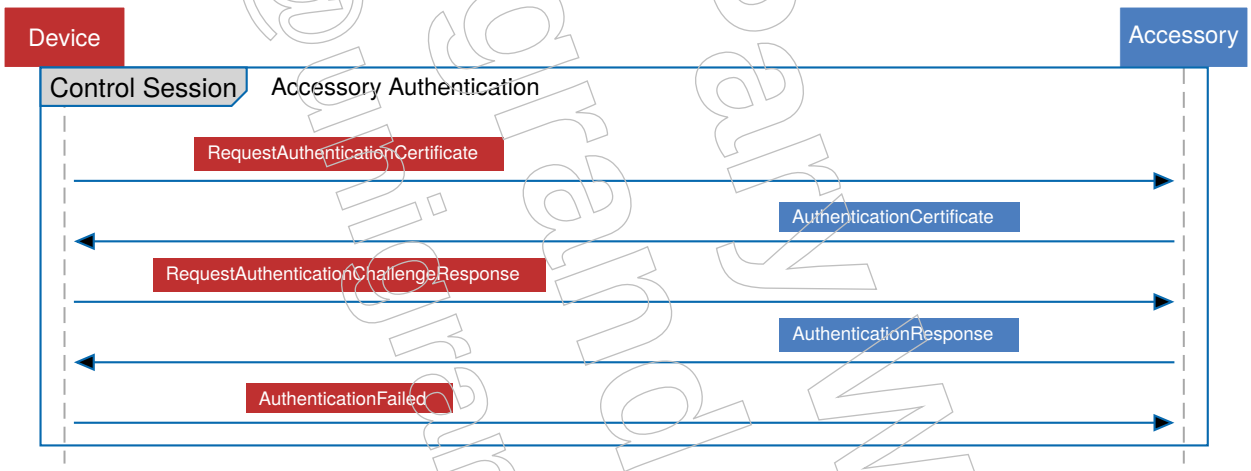


2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

9.3.2 Accessory Authentication over iAP2 Failure Due To Invalid Certificate



9.3.3 Accessory Authentication over iAP2 Failure Due To Invalid Response



Paywu International LTD
paywu@international.com.tw

10. Accessory Identification

All accessory interface features other than charging require that the identification feature be supported.

10.1 Accessory Identification Requirements

The accessory must have successfully completed Accessory Authentication (page 201) unless using control session version 2. See Control Session (page 562).

Control session version 2 allows the Apple device to choose the order of identification and authentication, so the accessory must be ready for either StartIdentification (page 579) or RequestAuthenticationCertificate (page 577) after successful link layer negotiation.

10.2 Accessory Identification Usage

All accessories that support the Accessory Identification feature via iAP2 must send or receive the following iAP2 control session message(s):

- StartIdentification (page 579)
- IdentificationInformation (page 579)
- IdentificationAccepted (page 587)
- IdentificationRejected (page 587)
- CancelIdentification (page 589)

All accessories that support the Accessory Identification feature via iAP2 may also send or receive the following iAP2 control session message(s):

- IdentificationInformationUpdate (page 589)

Identification is always initiated with the transmission of a StartIdentification (page 579) message from the Apple device to the accessory. The accessory must respond with a IdentificationInformation (page 579) message. The IdentificationInformation (page 579) message is evaluated by the Apple device to determine whether all of the requested iAP2 connection features are supported. If the parameters are valid and the Apple device can support all of the requested messages, it will send a IdentificationAccepted (page 587) message to the accessory.

Third Party Wu Hardware and Software Ltd. www.wuhardware.com.tw

Otherwise, a IdentificationRejected (page 587) message is sent detailing which invalid parameters and unsupported messages were present in the previous IdentificationInformation (page 579) message. Note that it is possible for the list of unsupported messages to be empty; that is an indication of incorrect accessory implementation, rather than unsupported features on a specific device.

Accessory Test System (ATS) output and Apple device console logs (see 'Getting Crash Logs and Console Output' at http://developer.apple.com/library/ios/#qa/qa1747/ will also provide additional supporting information which is useful for accessory development. Apple strongly recommends that accessory developers pay close attention to both ATS and the device logs.

If the identification information has been rejected, the accessory must send another IdentificationInformation (page 579) message with different parameters or abort the identification process using CancelIdentification (page 589). Cancellation of the identification process will cause the Apple device to inform the user that the accessory is not supported.

Failure of the accessory to send IdentificationInformation (page 579) within 1 second of either StartIdentification (page 579) or IdentificationRejected (page 587) is grounds for failing to pass self certification.

Once IdentificationAccepted (page 587) is received, the accessory must not send any more identification messages other than IdentificationInformationUpdate (page 589).

It is possible for an accessory to update its identification information after IdentificationAccepted (page 587) by sending an IdentificationInformationUpdate (page 589) message. For example, the user may change the accessory's active language. When that occurs, the accessory must send an IdentificationInformationUpdate (page 589) message containing both the new active language and accessory information strings in the new language.

If the identification process fails or the accessory cancels identification, the Apple device may choose to restart the identification process or terminate the iAP2 link.

10.2.1 Accessory Identification of Manufacturing Information

Accessories must provide values for the following IdentificationInformation (page 579) message parameters that match the accessory's marking and packaging:

- `Name`
- `ModelIdentifier`
- `Manufacturer`
- `SerialNumber`

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Additionally, the `FirmwareVersion` and `HardwareVersion` parameter values must uniquely reflect the current revision of the accessory's firmware and hardware respectively.

Accessories must not provide blank or generic string values (such as those provided by a reference design or component vendor) for any manufacturing information parameters.

10.2.2 Accessory Identification of Sent/Received iAP2 Control Session Messages

The `MessagesSentByAccessory` and `MessagesReceivedFromDevice` parameters in IdentificationInformation (page 579) are critical to accessory identification. Accessories must exhaustively enumerate all iAP2 control session messages that they can send or receive. During the self certification process, the accessory must be used in a manner that demonstrates correct implementation of all declared messages.

There is only one exception to this requirement. Messages associated with the Authentication and Identification features do not need to be in the `MessagesSentByAccessory` or `MessagesReceivedFromDevice` lists, as all accessories that implement iAP2 must support those features.

10.2.3 Accessory Identification of Power Capabilities

All accessories must provide values for the `PowerProvidingCapability` and `MaximumCurrentDrawnFromDevice` parameters in their IdentificationInformation (page 579) messages.

If an accessory does not provide power to the Apple device, `PowerProvidingCapability` must be set to `None`.

If an accessory does not draw power from the Apple device, `MaximumCurrentDrawnFromDevice` must be set to `0`.

Otherwise, the accessory must set those parameters as required by Power (page 480).

10.2.4 Accessory Identification of iAP2 Transport Components

Every accessory that implements iAP2 must identify one and only one transport component that supports an iAP2 connection. That transport component must contain a `TransportSupportsiAP2Connection` parameter. Depending on the type of transport, additional identifying information, such as a Media Access Control (MAC) address, may be required. The iAP2 connection must be implemented using the identified transport component.

Some accessory interface features require a specific transport component to be implemented in order to function. This requirement is independent of whether or not an iAP2 connection is implemented on that transport. For example, an accessory must identify a `USBDeviceModeTransportComponent` for Digital Audio

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

over the USB Device Mode transport. Every identified transport component must have a unique identifier and human-readable name that reflects the intended purpose of the component, such as 'iAP2' or 'Digital Audio Speakers'.

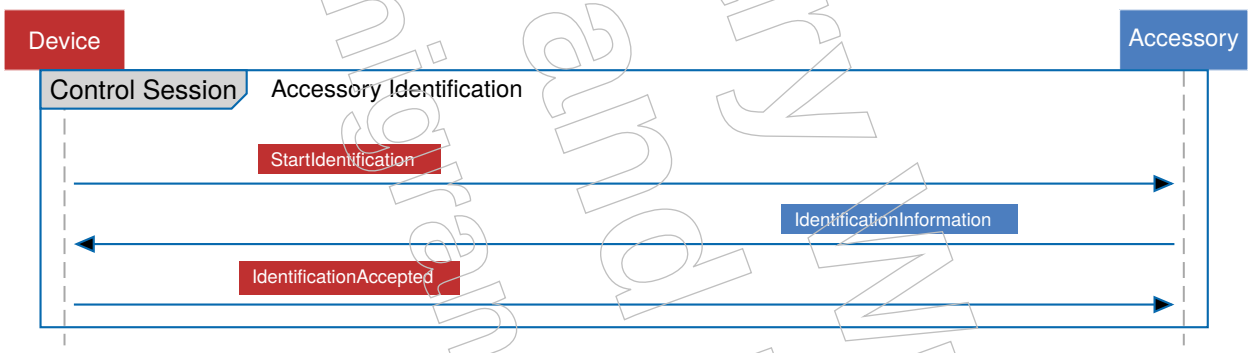
All transport components must have unique identifiers.

10.2.5 Additional Accessory Identification Parameters

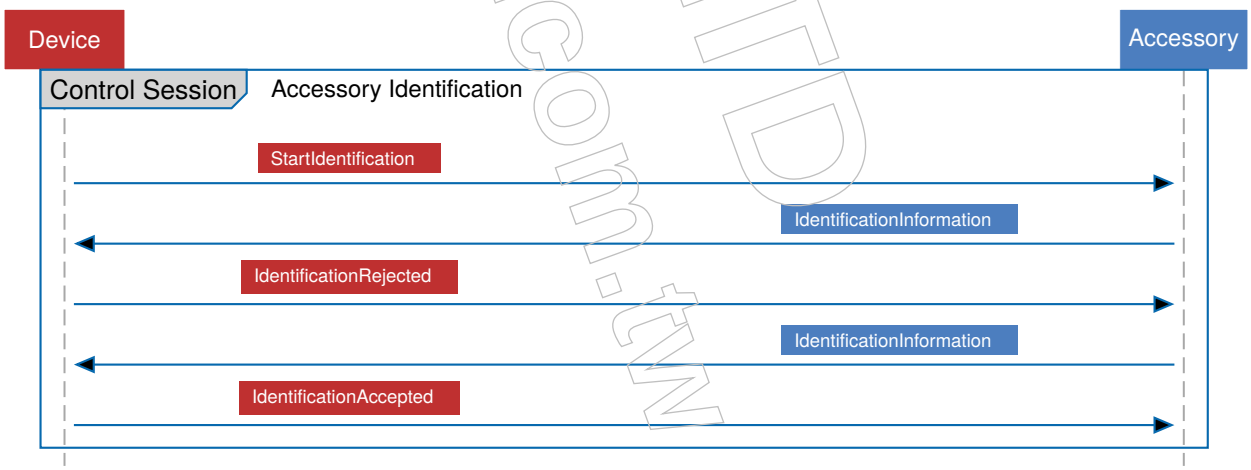
Several accessory interface features require the accessory to supply additional identification parameters. See the appropriate sections in the specification for details.

10.3 Accessory Identification Examples

10.3.1 Typical Accessory Identification

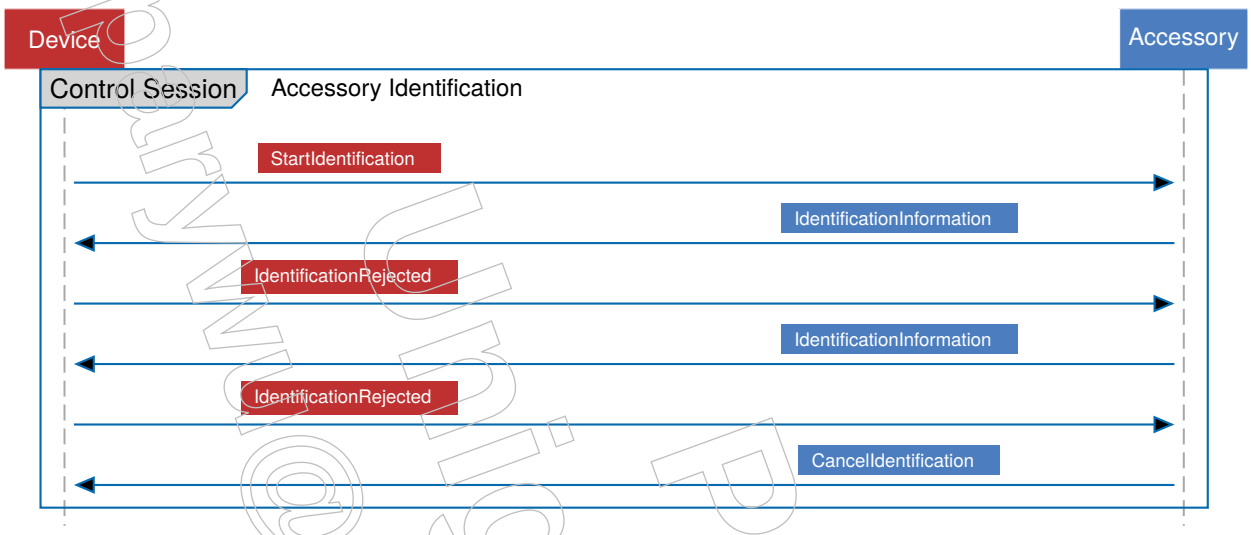


10.3.2 Successful Accessory Identification With Two Tries

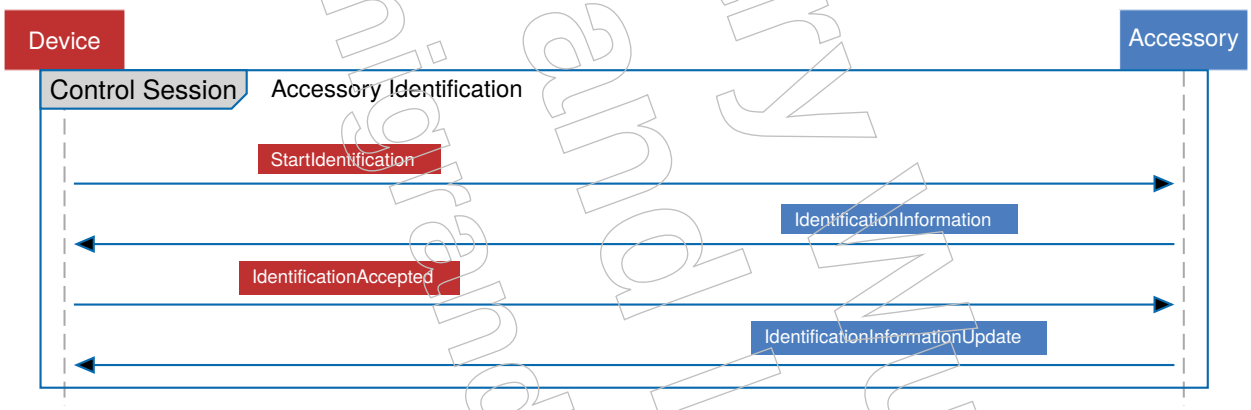


Party Wu Unigiantal LTD
partywu@unigiantal.com.tw

10.3.3 Unsuccessful Accessory Identification After Two Tries



10.3.4 Accessory Identification With Information Update

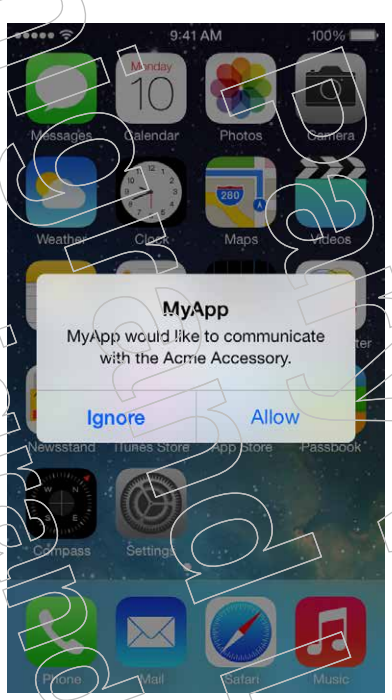


Copyright © 2014 Apple Inc. All Rights Reserved.

11. App Launch

An accessory that supports the App Launch feature can request that an Apple device launch an app on its behalf.

Figure 11-1 App Launch Alert



11.1 App Launch Requirements

All accessories that support the App Launch feature via iAP2 must send or receive the following iAP2 control session message(s):

RequestAppLaunch (page 590)

Note: This message must only be sent in response to direct user action, such as pushing a button on the accessory. Autonomous launch of apps is explicitly prohibited unless the accessory is an automotive headunit.

parvaz@unigiland.com.tw Unigiland.com.tw

The launched app must communicate with the accessory via the External Accessory Protocol (page 372).

11.2 App Launch Usage

If an Apple device does not support the app launch feature it will reject all related accessory identification attempts that call for RequestAppLaunch (page 590) to be sent by the accessory.

Accessories that establish an iAP connection via the Lightning connector may set `LaunchAlert` to 'no alert'. All other accessories must set `LaunchAlert` to 'alert'.

Accessories must not assume that the app has launched, because the request may be denied by iOS or the user. If positive confirmation of app launch is needed, the app must open an External Accessory Protocol (page 372) session with the accessory.

Pary Wu
Unigrand LTD
pary.wu@unigrand.com.tw

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

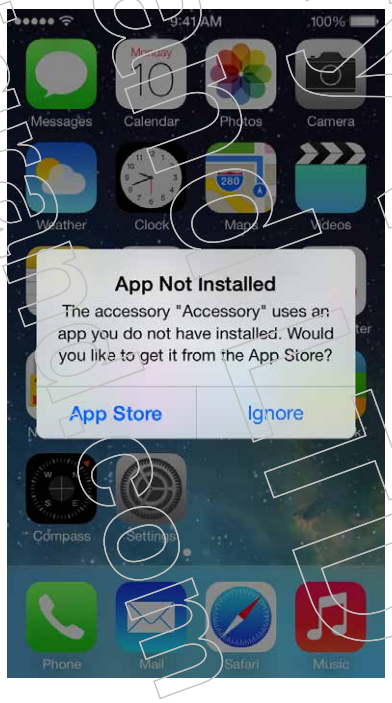
12. App Match

The App Match feature enables accessories that support the External Accessory Protocol feature (see External Accessory Protocol (page 372)) to match with compatible apps on the App Store.

When connected for the first time, the Apple device will ask the user if they would like to visit the App Store and view apps that work with the accessory. Subsequently, this action may be repeated by the user via *Settings > General > About > 'Accessory Name' > 'Find App for this Accessory'*.

Matched apps are listed in alphabetical order with one exception. If the accessory will work with apps from multiple development teams/companies, the accessory may provide a preferred Team ID that places apps from that development team at the top of the list.

Figure 12-1 App Match Alert



12.1 App Match Requirements

To support App Match during Accessory Identification (see Accessory Identification (page 205)), the accessory:

Paul Wu paul@pbtech.com.tw iPod/iPhone/iPad

- Must declare one or more supported External Accessory Protocols (see External Accessory Protocol (page 372)) that matches against one or more apps that are allowed to declare support for the same External Accessory Protocol(s).
- May additionally declare a preferred Team ID that matches against one or more apps from a single app development team.

12.1.1 External Accessory Protocol

Unless otherwise specified, accessories must fully implement at least one External Accessory Protocol (see External Accessory Protocol (page 372)) and use it to communicate with iOS apps in order to support App Match.

Accessories may use the App Match feature with more than one External Accessory Protocol. If multiple protocols are listed, the App Store will display a list of apps that support one or more of those protocols.

12.1.2 Team ID

- The Team ID used for App Match must belong to the accessory developer.
- A maximum of one Team ID can be declared for App Match purposes.
- The Team ID also must not change dynamically.

12.2 App Match Usage

12.2.1 External Accessory Protocol

Accessories must support App Match for a particular External Accessory Protocol by setting the MatchAction parameter (see Table 52-11 (page 582)) in the protocol's parameter group (see Table 52-10 (page 581)) to an appropriate value during accessory identification.

12.2.2 Team ID

Accessories may declare compatibility with apps by providing a AppMatchTeamID parameter in the IdentificationInformation message (see IdentificationInformation (page 579)). The parameter's value must be a valid iOS Developer Program Team ID.

The Team ID is supplied by Apple and is unique to a specific development team. More information on the Team ID can be found in the iOS Developer Library (see https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/AppID.html).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

13. AssistiveTouch

The AssistiveTouch feature lets users generate iOS Multi-Touch gestures using one finger, a stylus, or a compatible accessory. Accessory developers should consider supporting AssistiveTouch if the accessory is intended for use by users with special needs who can see the display and manipulate more complex input peripherals, such as joysticks or touchpads. If the user cannot see the display, consider supporting the VoiceOver feature instead.

When an AssistiveTouch-compatible accessory connects to an Apple device and the feature is enabled via the iOS Settings app, an accessory-controlled virtual cursor is displayed. All AssistiveTouch interaction then proceeds as if the user's own fingers were being used to interact with the device.

Figure 13-1 AssistiveTouch Pointer

13.1 AssistiveTouch Requirements

All accessories supporting AssistiveTouch must be targeted at users with special needs. The accessory must also implement a `HIDComponent` that complies with all HID AssistiveTouch Pointer Requirements (page 413).

Prerelease Universal Binary Unsigned

All accessories that support the AssistiveTouch feature via iAP2 must send or receive the following iAP2 control session message(s):

StartAssistiveTouch (page 591)

StopAssistiveTouch (page 591)

StartAssistiveTouchInformation (page 591)

AssistiveTouchInformation (page 592)

StopAssistiveTouchInformation (page 592)

13.2 AssistiveTouch Usage

Accessories supporting AssistiveTouch must identify themselves as sending and receiving all messages associated with the feature. There are no optional messages. If the device does not support AssistiveTouch then it will reject all identification attempts that include AssistiveTouch messages. The accessory must also use the HID feature to register the HID Mouse descriptor with the device.

After identification has been completed and accepted, the accessory must send a StartAssistiveTouchInformation (page 591) message to the device. While these notifications are active, an AssistiveTouchInformation (page 592) message will be sent from the device to inform the accessory whether the feature is currently enabled or disabled. Accessories must be prepared to have the AssistiveTouch feature disabled or enabled at any time, whether by the user or by another accessory.

The AssistiveTouch feature can be disabled and enabled via the StartAssistiveTouch (page 591) and StopAssistiveTouch (page 591) messages while the accessory is still attached to an Apple device. This can be useful if the accessory needs to enter a different operating mode that does not provide an external pointing input to AssistiveTouch.

Confidential Property of Pegatron LTD
Reproduction Prohibited
Pegatron.com.tw

14. Bluetooth

Accessories that integrate Bluetooth technology must comply with the requirements stated in this chapter.

14.1 Conformity With Bluetooth Specifications

Every accessory that is compatible with an Apple product must support the *Bluetooth Core Specification* Version 2.1 + EDR or higher. This specification introduced the important security feature Secure Simple Pairing as well as Extended Inquiry Response.

14.1.1 Enhanced Data Rate

The Enhanced Data Rate (EDR) feature introduced in the *Bluetooth 2.0* specification enables accessories to communication more efficiently. Every accessory must use EDR for the following reasons:

- It provides higher data rates compared to Basic Data Rate (BDR).
- It communicates more efficiently, transferring more data bits per unit of time.
- It reduces the power consumption used per bit transferred.
- It improves coexistence with Wi-Fi and other connected Bluetooth devices because it frees up more air time.
- It improves performance in multipoint configurations.

14.1.2 Sniff Mode for Low Power Consumption

Minimizing power consumption is critical for all mobile devices. Therefore, every accessory that is compatible with an Apple product:

- Must support and must request Bluetooth sniff mode.
- Must accept requests for sniff mode and support all valid parameters listed in the Bluetooth specification.
- Must support a sniff interval of 15 ms.
- Must not renegotiate sniff after being established.
- Must support sniff subrating.

Taiwan Unigrand LTD
taiwan@unigrand.com

Accessories that are compatible with Apple products must also use sniff mode as much as possible, especially when there is little or no data being transmitted over the Bluetooth link. Besides its power consumption advantages, sniff mode enables better antenna sharing with Wi-Fi.

The sniff mode parameters are specific to the usage model and Bluetooth Profile. The Apple product expects the accessory to request sniff mode with appropriate parameters for a specific usage. If the accessory does not send such a request, the Apple product may send a sniff mode request. When the Apple product sends a request for sniff mode, the remote device must accept the request and its parameters without negotiation.

If the accessory sets the sniff mode parameters, the accessory must set the sniff interval to less than a third of the Bluetooth baseband Link Supervision Timeout. This makes the Bluetooth link less susceptible to interference. To improve link robustness, the accessory must use a shorter sniff interval instead of multiple sniff attempts.

Links with a sniff interval of 1 second or more make the slave device open up a large correlation window, which has to be taken into account when calculating the number of sniff attempts. With sniff intervals shorter than 1 second, multiple sniff attempts can improve link robustness but will increase power consumption.

14.1.3 Role and Topology Management

Every accessory that is compatible with an Apple product must:

- Accept a request for Role Switch from an Apple product.
- Continue with the connection when the Apple product rejects a request for Role Switch.

In a Bluetooth connection, one device is the master and the other the slave. The master can have multiple slaves, thus forming a piconet. The master device can also be a slave role to another master, creating a scatternet.

Such a scenario creates complications since the device has to alternate between the two piconets and thus wastes valuable bandwidth. Managing the topology of the network is therefore important for maximum performance. The Apple product may request a Role Switch, depending on its current topology, and the remote device must accept the request. The Apple product may also reject a request for a Role Switch because of topology concerns. Having a suboptimal topology may degrade the audio quality and the user's experience.

Only when it is maintaining multiple links, either Bluetooth or Wi-Fi, will the Apple product request or deny role switches. Hence, it will grant a role switch if there is no reason for the Apple product to be master. It is expected that the accessory will behave the same, only trying to be master when there is a legitimate reason.

The accessory must not always request to be master by default if there is no need in the system topology to do so. If later the accessory needs to be master in order to maintain multiple links, it must ask to be master at that time.

Preliminary Wireless Interface Specification

14.1.4 Extended Inquiry Response

Every accessory that is compatible with an Apple product must provide the following information in its Extended Inquiry Response packet:

- The Local Name of the product (Complete or Shortened).
- The TX Power Level.
- The Service Class UUID for the iAP2 protocol, if applicable (see iAP2 (page 229)).

During the Bluetooth discovery process, the Apple product prefers to display the Friendly Name of discovered accessories. Before the 2.1 version of the Bluetooth specification the Apple product would have to set up a connection to the accessory and do a Remote Name Request, which takes power, antenna time, and user's time. The Extended Inquiry Response feature, introduced in Bluetooth 2.1, lets an accessory send its Local Name and other information as part of the Inquiry Response and thereby increase the speed and efficiency of the discovery process.

14.1.5 Secure Simple Pairing

Every accessory that is compatible with an Apple product must:

- Use Secure Simple Pairing.
- Use the Numerical Comparison method if it has a display and input device supporting it.

Secure Simple Pairing greatly increases security and is a mandatory security feature introduced in the Bluetooth 2.1 specification. To protect against a man-in-the-middle attack, the Numerical Comparison association model must be used whenever feasible. See Volume 1, Section 5.4 in the *Bluetooth Core Specification*, Version 2.1 + EDR.

14.2 Profiles

The Apple knowledge base article support.apple.com/kb/ht3647 provides a complete list of the Bluetooth profiles that certain Apple devices support. The Bluetooth specifications are the starting point for designing accessories that are compatible with these products. The following sections add information and requirements for some profiles, which can help accessory developers achieve superior results.

14.2.1 Device ID Profile (DID)

Every accessory that is compatible with an Apple product must:

- Support the Bluetooth Device ID Profile, version 1.3 or higher.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- Use the company identifier from the Assigned Numbers specification assigned by the Bluetooth SIG as its Vendor ID value (VID). See http://www.bluetooth.org/Technical/AssignedNumbers/identifiers.htm (requires login). Bluetooth HID Profile accessories may use a VID assigned by the USB Implementers Forum (USB-IF at http://www.usb.org) if the manufacturer does not have a Bluetooth SIG company identifier.
- Use its VID value for the end product manufacturer.
- Use the Vendor ID Source field to identify which organization assigned the value used in Vendor ID field value. See Section 5.6 of the *Bluetooth Device ID Profile Specification*.
- Use a ProductID value that uniquely identities the product.
- Use a Version value that uniquely identifies the software version.

The Device ID profile lets the Apple product identify the implementation of the remote accessory. This is valuable information and can be used to bridge alternate interpretations of the Bluetooth specification when communicating with a remote accessory. Therefore it is important that the information in the Device ID record uniquely identify the implementation.

In the case of Bluetooth car kit devices, for instance, the same car kit might go into two different car models. Ideally the two car kits must have different Product IDs. However, it is acceptable for them to have the same ProductID as long as they have identical hardware, software, and features. If the implementations differ at all, they must have different Product IDs. The accessory can also use a secondary Device ID to uniquely identify the product ID or model number.

14.2.2 Hands-Free Profile (HFP)

Every accessory that is compatible with an Apple product and supports the Hands Free Profile must meet the requirements of the Bluetooth *Hands-Free Profile* specification, Version 1.5 or higher. Additional Apple requirements are specified in this section.

Remote accessories can use the Bluetooth *Hands-Free Profile* for phone communications. To achieve the best user experience, the remote accessory must support the following features, which are optional in the Bluetooth specification.

14.2.2.1 Remote Audio Volume Control

Every accessory that is compatible with an Apple product and supports HFP must:

- Support Remote Audio Volume Control so the speaker volume on the Hands-Free accessory can be controlled from the Apple product as described in Section 4.28 in the Bluetooth *Hands-Free Profile* specification version 1.5.
- Set the Remote volume control bit in the Supported Features bitmap sent with the `AT+BRSF=` command.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

In some situations it is easier for the user to control the output volume through the Apple product instead of directly on the remote accessory. For example, a passenger (or-if the car is parked-the driver) in a car could use the volume slider on the phone to control the audio volume. Volume control synchronization is outlined in Section 4.48.2 in the Bluetooth *Hands-Free Profile* specification version 1.5.

14.2.2.2 Indicator Event Reporting

Every accessory that is compatible with an Apple product and supports HFP must use indicator events reporting and not perform repetitive polling of status.

Apple products support all mandatory and optional indicators specified in HFP version 1.5 (service, call, callsetup, callheld, signal, roam, battchg). To minimize unnecessary polling of status using the `AT+CIND?` command, the remote accessory must enable indicator events reporting by sending an `AT+CMER` command. The Apple product will then send a `+CIEV` event when there is a change in status of an indicator. The remote accessory must request the initial status using the `AT+CIND=?` and `AT+CIND?` commands, according to the HFP specification.

14.2.2.3 Voice Recognition Activation

Every accessory that is compatible with an Apple product and supports HFP must:

- Support Voice Recognition Activation, both AG and HF initiated as described in Section 4.25 in the Bluetooth *Hands-Free Profile* specification version 1.5.
- Set the "Voice recognition activation" bit in the "SupportedFeatures" bitmap sent with the `AT+BRSF=` command.

Apple products support voice recognition initiated by remote (Hands-Free) accessories and iOS (Audio Gateway) accessories.

14.2.2.4 Echo Cancellation and Noise Reduction

When echo cancellation and noise reduction are performed locally on a Hands-Free accessory, it must turn off echo cancellation and noise reduction on the Apple product by sending an `AT+NREC` command, as described in Section 4.24 in the Bluetooth *Hands-Free Profile* specification version 1.5.

Apple products support echo cancellation and noise reduction; these features are active by default. If a Hands-Free accessory also does echo cancellation and noise reduction it needs to turn these features off on the Apple product (the Audio Gateway). This avoids unnecessary degradation of audio quality due to double audio processing.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

14.2.2.5 In-Band Ringing

Every accessory that is compatible with an Apple product and supports HFP must also support In-Band Ringing as specified in Section 4.13.1 in the Bluetooth *Hands-Free Profile* specification version 1.5. If the user sets a ring tone on the Apple product, the same ring tone must sound on the hands-free accessory.

14.2.2.6 Synchronous Connection

Every accessory that is compatible with an Apple product and supports HFP must:

- Support eSCO parameter set S2 and S3 and accept requests for these settings. See Section 5.6 of the Bluetooth *Hands-Free Profile* specification version 1.5.
- Request eSCO parameter set S2 or S3 when setting up a Synchronous Connection. Note that eSCO parameter set S1 must not be requested.
- Render audio within 40 ms after the SCO/eSCO connection has been set up.

The eSCO packet types offers retransmission of packets; traditional SCO packets are not retransmitted. This improves audio quality and the user's experience. The eSCO packet types 2-EV3 and 3-EV3 offer a greater time interval between packets, which can improve Wi-Fi performance and allow time for other concurrent Bluetooth connections to send data. Apple strongly recommends the use of 2-EV3 and 3-EV3 packets for SCO connections. Using HV3 packets is highly discouraged. HV3 packets require more link time and does not allow for retransmission of audio packets which impacts the audio performance in presence of RF interference.

14.2.2.7 Wide Band Speech

Every accessory that is compatible with an Apple product and supports HFP must support a Wide Band Speech Connection as described in Section 5.7.4 of the Bluetooth *Hands-Free Profile* specification version 1.6. If Wide Band Speech Connection is supported, it must support the T2 link parameter settings.

All Apple devices running iOS 5 or later support Wide Band Speech. If both the Apple device and the accessory support Wide Band Speech then Wide Band Speech link will be used for eSCO connection for use cases like cellular calls, FaceTime and Siri.

14.2.3 Message Access Profile (MAP)

Every accessory that is compatible with an Apple product and supports MAP must:

- Support Message Notification as described in Section 4.1 of the Bluetooth *Message Access Profile* specification, version 1.0.
- Register for notifications immediately after the connection is established, as described in Section 4.5 in the *Message Access Profile* specification, version 1.0.

Confidential Property of Apple Inc.

- Not expect the TEL property to be present in the originator VCARD (the properties N and FN will be included). See Section 3.1.3 in the *Message Access Profile* specification, version 1.0.
- Not provide a user interface for sending messages. Apple devices do not support sending messages using MAP.

All Apple devices running iOS 6.0 or later support MAP.

14.2.4 Audio/Video Remote Control Profile (AVRCP)

Every accessory that is compatible with an Apple product and supports the Audio/Video Remote Control Profile must meet the requirements of the Bluetooth *Audio/Video Remote Control Profile* specification, Version 1.4. Additional Apple requirements are specified in this section.

14.2.4.1 Supported Operations

Apple products support the following operation_IDs in Pass Through commands:

- Play
- Stop
- Pause
- Fast Forward
- Rewind
- Forward
- Backward

14.2.4.2 Repeat and Shuffle Modes

Every Apple device supports Repeat and Shuffle modes in the role of an AVRCP target. An AVRCP controller may use `SetPlayerApplicationSettingValue` to set a value on the Apple device and `GetPlayerApplicationSettingValue` to read a value, as described in Sections 6.5.4 and 6.4.3 of the Bluetooth *Audio/Video Remote Control Profile* specification version 1.4.

14.2.4.3 Notifications

Every accessory that is compatible with an Apple product and supports AVRCP must register for notifications and not perform repetitive polling to determine the status of the Apple product.

Party Wu
Digiland LTD
partywu@digiland.com.tw

Every Apple device supports registering for notifications in the role of an AVRCP Target, as described in Section 6.7 of the Bluetooth *Audio/Video Remote Control Profile* specification version 1.4. The commands `RegisterNotification` and `GetPlayStatus` are supported for these notifications:

- EVENT_PLAYBACK_STATUS_CHANGED
- EVENT_TRACK_CHANGED
- EVENT_NOW_PLAYING_CONTENT_CHANGED
- EVENT_AVAILABLE_PLAYERS_CHANGED
- EVENT_ADDRESSED_PLAYER_CHANGED
- EVENT_VOLUME_CHANGED

Accessories must use AVRCP notifications to present Apple device state to the user in accordance with the requirements in Presentation of Apple Device Updates (page 49).

14.2.4.4 Play/Pause Button

All accessories that support AVRCP and implement a Play/Pause button must confirm the playback status of the Apple device via AVRCP notifications (see Notifications (page 222) before sending a Play or Pause command (see Supported Operations (page 222)). Specifically:

- If the Apple device has notified the accessory that it is paused, pressing the accessory's Play/Pause button must send a Play command.
- If the Apple device has notified the accessory that it is playing, pressing the accessory's Play/Button button must send a Pause command.
- The accessory must not infer Apple device playback status based on the number of times the Play/Pause button has been pressed.

14.2.4.5 Volume Handling

Every accessory that is compatible with an Apple product and supports AVRCP must support Absolute Volume, as described in Section 6.13 of the Bluetooth *Audio/Video Remote Control Profile* specification version 1.4.

Every Apple device supports volume handling in the role of AVRCP Controller.

14.2.4.6 Browsing

Every accessory that is compatible with an Apple product and supports Browsing (in controller role) as part of AVRCP must:

- Not try to index or cache the entire library upon connection. The Apple product may contain tens of thousands of media items, each present multiple times in the hierarchy.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- When browsing a specific folder, do not fetch all its items. Only fetch those that are displayed to the user. It may prefetch a few items to improve the responsiveness of the user interface.
- Not reorder items (e.g. alphabetically).
- Not assume UIDs to be statically defined, especially in the root folder. The ordering and UIDs of folders and items may change at any point in future releases.
- Send the `SetBrowsedPlayer` command after receiving an EVENT_UIDS_CHANGED notification.
- Not assume that the UID passed to the PlayItem command will result in the media player playing that UID.

Currently only the built-in Music app supports browsing. When switching between players, an EVENT_AVAILABLE_PLAYERS_CHANGED notification and an EVENT_ADDRESSED_PLAYER_CHANGED notification will be generated. The UI then needs to look at the feature bit mask of the listed player to determine whether browsing is currently available.

All Apple devices running iOS 6.0 or later support AVRCP Browsing.

14.2.4.7 iOS App-Provided Metadata

An audio app running on an Apple device may use the iOS MediaPlayer Framework APIs to provide metadata about the current audio stream. The Apple device supplies this metadata to the accessory using AVRCP. For more information, see the `MPNowPlayingInfoCenter` class in Apple's MediaPlayer Framework documentation.

14.2.5 Advanced Audio Distribution Profile (A2DP)

Every accessory that is compatible with an Apple product and supports the Advanced Audio Distribution Profile must meet the requirements of the Bluetooth *Advanced Audio Distribution Profile* specification, Version 1.2. Additional Apple requirements are specified in this section.

14.2.5.1 SubBand Codec (SBC)

The SBC Codec Specific Information Elements, defined in Section 4.3.2 of the A2DP specification, that are applicable to Apple products are listed in Table 14-1 (page 224).

Table 14-1 SubBand Codec Information Elements for Apple products

Element	Value
Sampling Frequency	44,100 Hz
Channel Mode	Stereo
Block Length	16

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Element	Value
Subbands	8
Allocation Method	Loudness
Bitpool range	2 to 53. Accessories for Apple products must support 53.

14.2.5.2 MPEG 2/4 AAC Codecs

Apple devices support the non-mandatory codec MPEG-2/4 AAC, as defined in Section 4.5 of the *Advanced Audio Distribution Profile* specification, Version 1.2. Accessories must use the AAC codec in addition to SBC, because it provides higher audio quality for a given bit rate.

> **Note:** The following specifications provide details of Apple's implementation of the MPEG-2/4 AAC codec. In case of conflicts, the A2DP specification governs.

The MPEG 2/4 AAC Codec Specific Information Elements, defined in Section 4.5 of the A2DP specification, that are applicable to Apple devices are listed in Table 14-2 (page 225).

Table 14-2 MPEG-2/4 AAC Codec Information Elements for Apple devices

Element	Value
Object Type	MPEG-2 AAC LC
Sampling Frequency	44,100 Hz
Channels	2
Bit rate	264,630 bps
VBR	0

AAC audio stream packets in Apple devices have the structure shown in Table 14-3 (page 225).

Table 14-3 AAC audio packet for Apple devices

L2CAP	AVDTP	MPEG-4 LATM	MPEG-4 AAC
Header	Header	AudioMuxElement	Audio Payload

Paul Wu
Ubiquitous LTD
paul@ubiquitous.com.tw

The AAC Media Payload Format, as defined in Section 4.5.4 of the A2DP specification, is formatted using LATM, as defined in Section 4 of *IETF RFC 3016* . The following notes apply to the packet fields shown in Table 14-3 (page 225).

- The suggested L2CAP MTU value for each Apple device's AAC streaming channel is 885 bytes.
- The AVDTP Header is shown as the RTP header in Figure 4 of RFC 3016, and is the header defined in Section 7.2.1 of the Bluetooth *Audio/Video Distribution Transport Protocol* , Version 1.2.
- The `AudioMuxElement` is the same as the RTP payload in RFC 3016. It is defined in Section 1.7.3, Table 1.32 in ISO/IEC 13818-3:2005, subpart 1. The `muxConfigPresent` argument to the `AudioMuxElement` is set to 1 (in-band mode), as recommended in Section 4.1 of RFC 3016. As recommended in Section 4.3 of RFC 3016, only one `AudioMuxElement` is put into each AVDTP packet.
- The audio payload is encoded using MPEG-4, as recommended in Section 4.5.4 of the A2DP specification.
- For AAC-LC support, the accessory must support VBR capability. The Apple device will be varying AAC bit rate depending on the content and the accessory must be able to handle the variation without causing gap in the audio.

14.3 Audio Routing

This section describes how an accessory can differentiate between various audio contents coming from an Apple device and use this information to decide playback behavior.

An accessory can receive audio data from the Apple device via either of two Bluetooth profiles:

- HFP using eSCO channel
- A2DP using ACL channel

The Apple device picks which channel to use depending on how the audio content is used. An audio path created for two way communication (such as phone calls or FaceTime) always uses the HFP (eSCO) route for sending audio data. Music and similar content uses the A2DP route. In the absence of a defined route, audio playback will default to the Apple device.

If the accessory is already connected to the Apple device using the Lightning or 30-pin connector, connecting using Bluetooth must not pause any playing audio.

14.3.1 Audio Data Received via HFP Profile

Most of the audio content sent via HFP (eSCO) routes requires two way communication. Cases where HFP (eSCO) is used include (but are not limited to) cellular calls, FaceTime, and voice mail.

Tanya Cheng Pan Wu
tanya@planetband.com.tw

For any audio content that is being received via the HFP (eSCO) route, it is expected that both the speaker and the microphone of the accessory are dedicated to the Bluetooth link and must not handle any other audio content.

14.3.2 Audio Data Received via A2DP Profile

Audio content transferred via A2DP profiles can be broadly classified into two categories:

- Audio content from music, video, or game-like applications.
- System-generated sound for alerts and notifications.

14.3.2.1 Differentiating Audio Content from System Sounds

Music-like content can be differentiated from system sound by adding support for Audio/Video Remote Control Profile (AVRCP) version 1.3 or later. The AVRCP profile allows an accessory to be aware of the audio playback state in the Apple device, using notifications. See Audio/Video Remote Control Profile (AVRCP) (page 222).

When an Apple device initiates audio playback over an A2DP channel for playing music content, an AVRCP notification EVENT_PLAYBACK_STATUS_CHANGED is sent to indicate that playback status has changed to play state. See Section 6.7.2 of the *Audio/Video Remote Control Profile* specification, version 1.4. This indicates that audio data via the A2DP profile contains music. When an Apple device initiates audio playback over an A2DP channel for playing system sound, no AVRCP notification is sent.

Figure 14-1 (page 228) and Figure 14-2 (page 228) show the difference between the notifications for music playback and for system sounds.

Paul Chien@Pegatron LTD
Paul_Chien@pegatron.com.tw

Figure 14-1 Initiate Audio Playback (e.g. music)

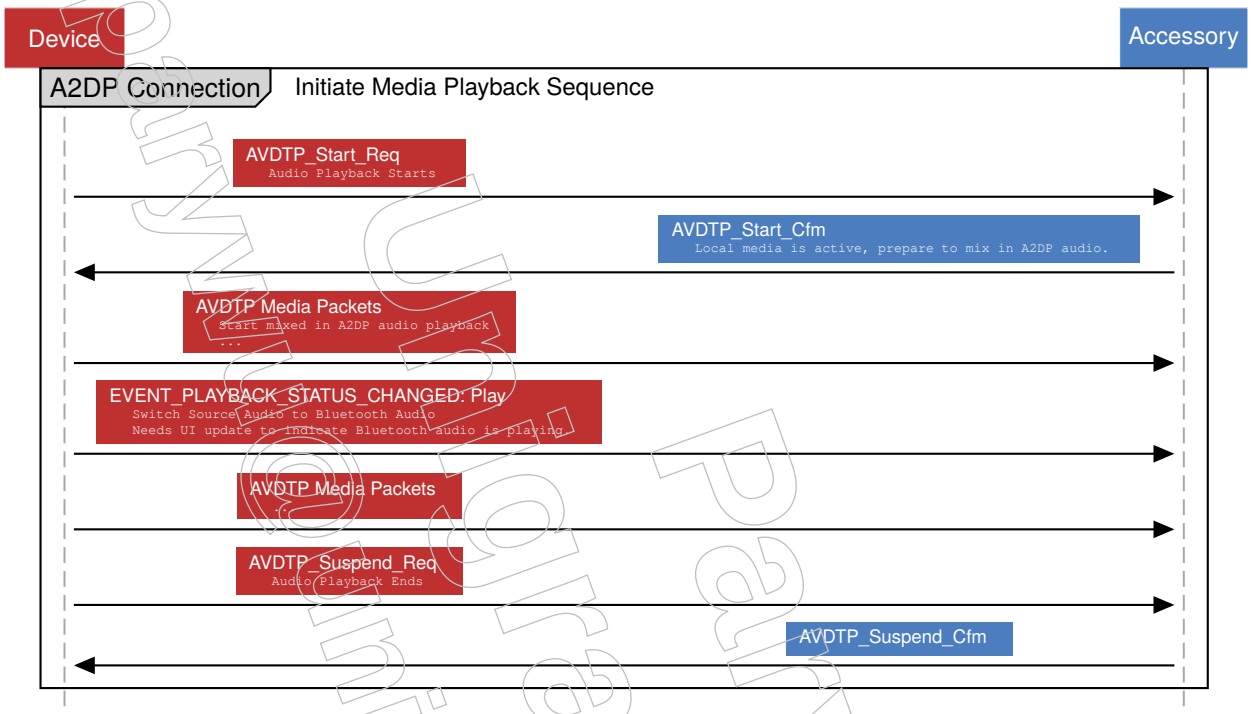
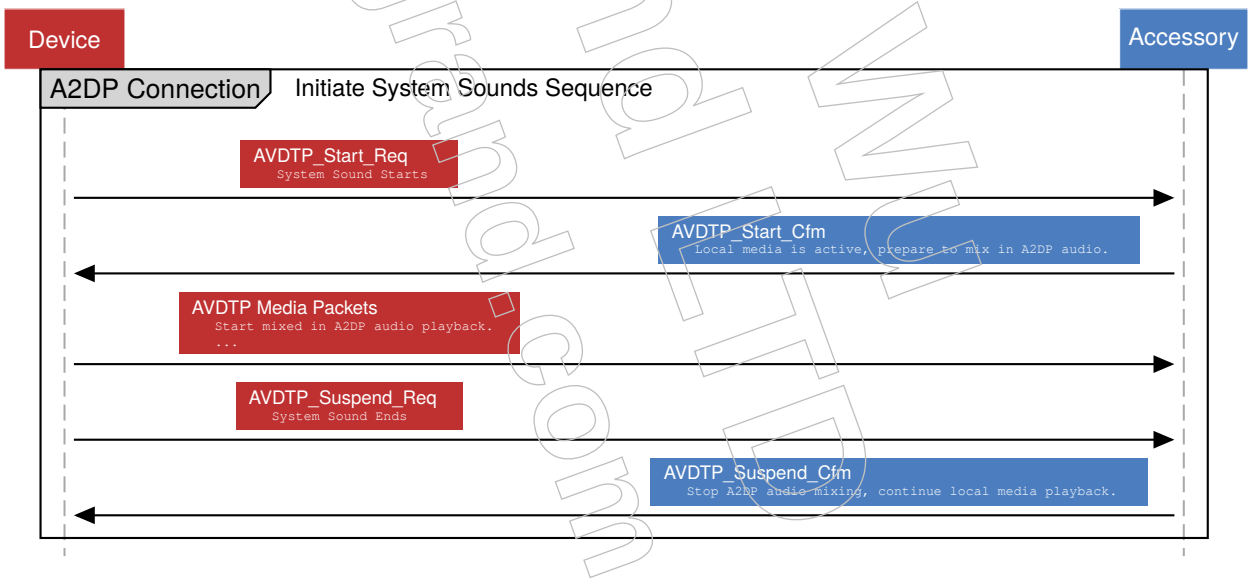


Figure 14-2 Initiate System Sound (e.g. turn-by-turn directions)



14.3.2.2 Expected Audio Routing Behavior for A2DP

The accessory must tune its audio routing behavior based on audio content over A2DP channel.

Apple Confidential
Apple Inc. Copyright (C) 2014 Apple Inc. All Rights Reserved.
Do Not Copy

If audio data contains music, then it is expected that the accessory speakers are dedicated to audio data coming via the Bluetooth link and any other audio playback is paused. If audio data contains system sound, then it is expected that the accessory can render audio as desired. If the accessory is playing audio from a different source, then system sound data can be mixed with the existing track for playback; it is not necessary to pause existing audio playback on the device.

14.4 iAP2

Accessories that implement iAP2 over the Bluetooth transport must additionally meet the following requirements:

- The Bluetooth Service Discovery Protocol (SDP) must be supported.
- The SDP data Maximum Transmission Unit (MTU) must be at least 672 bytes.
- SDP records must not be fragmented.
- Extended Inquiry Response (EIR) must be supported.
- A Service Class UUID of `0xFFCACADEAFDECADEDEFACADE00000000` must be declared in both SDP and EIR.
- The EIR Device Name must be the same as the `Name` parameter in the IdentificationInformation (page 579) message.

Unlike iAP1, there is no requirement for a specific Class of Device (CoD) or Major Service.

Apple recommends the following for a successful Bluetooth connection:

- Implement Bluetooth Sniff Mode, or Sniff Subrating if Bluetooth 2.1 is being used.
- Let the Apple device be the master device.
- For large packet transfers, stay within a Maximum Transmission Unit (MTU) size of 1000 bytes and a minimum of 658 bytes at the iAP2 layer.

Unlike wired transports, Apple devices will enter a hibernate state regardless of how many active Bluetooth connections are present. Bluetooth traffic from the accessory to the device will cause the device to exit the hibernate state. Therefore, accessories must generate Bluetooth traffic to an Apple device only in response to a direct user action. Autonomous generation of Bluetooth traffic for the purpose of keeping an Apple device out of hibernate is grounds for failure to pass self certification.

To re-establish a Bluetooth connection to the Apple device, the accessory must make a Bluetooth SDP query to find the RFCOMM channel associated with the Service Class UUID of `0xFECACADEAFDECADEDEFACADE00000000`, then connect to that channel. The accessory must not assume that the channel will remain the same between connections.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

The Apple device may refuse a Bluetooth connection under these circumstances:

- There are too many Bluetooth connections made to the Apple device. In this case, the accessory will receive an error indicating that a resource is unavailable.
- There are too many RFCOMM protocol connections to the Apple device. In this case, the accessory will receive a Resource Denied error.

The accessory must not expect that the Apple device will try to re-establish a broken Bluetooth connection.

Note: Service Class UUID and RFCOMM UUID are listed in little-endian order.

14.5 Test Procedures

14.5.1 Establishing a Connection

14.5.1.1 Device-initiated Connection

1. Go to the Settings app and select Bluetooth.
2. Tap on the accessory's name in the Devices list.

Expected result: A connection should be established.

14.5.1.2 Accessory-initiated Connection

1. Initiate a Bluetooth connection using the accessory.

Expected result: A connection should be established.

14.5.2 Role and Topology Management

1. Initiate a Bluetooth connection using the accessory.
2. Initiate a second Bluetooth connection using a different accessory.

Expected result: A connection should be established with the second accessory.

Pary Wu
LTD
Digiland
paywu@digiland.com.tw

14.5.3 Secure Simple Pairing (SSP)

1. Go to the Settings app and select Bluetooth.
2. Tap the blue triangle next to the accessory's name in the Devices list.
3. Tap the Forget this Device button.
4. Put the accessory in pairable mode.
5. Initiate the pairing from the device by tapping on the accessory's name in the Devices list.

Expected result: The accessory should successfully pair with the device.

14.5.4 Hands-Free Profile (HFP)

14.5.4.1 Voice Recognition

1. Establish a Bluetooth connection between the device and accessory.
2. Make a call using Voice Recognition from the device.
3. Adjust the call volume to 100% using the device buttons.
4. End the call.
5. Make a call using Voice Recognition from the accessory.
6. Maintain the call for 1 minute.
7. Adjust the call volume to 50% using the accessory.
8. End the call.

Expected result: Both calls should be successfully initiated using Voice Recognition. The volume should be adjustable from both the device and the accessory.

14.5.4.2 Accepting Call from Accessory

1. Establish a Bluetooth connection between the device and accessory.
2. Accept a call using the accessory.
3. Maintain the call for 1 minute.
4. End the call.

Expected result: The device's ringtone should have played through the accessory's speaker.

Party Wu
Universal LTD
partywu@universal-group.com.tw

14.5.5 Advanced Audio Distribution Profile (A2DP)

1. Establish a Bluetooth connection.
2. Play AAC-encoded audio through the accessory.
3. Allow the audio to play for at least 1 minute.
4. Press the Pause button on the accessory.
5. Press the Play button on the accessory.
6. Press the Pause button on the accessory.

Expected result: Audio should have played through the accessory's speaker with no audio skipping.

Pary Wu
Unigrand LTD
paryw@unigrand.com.tw

15. Bluetooth Accessory Identification

This chapter describes Apple-specific Bluetooth commands that extend accessory capabilities beyond those supported by standard Bluetooth profiles.

To enable Apple-specific features, the accessory must support HFP Command AT+XAPL (page 233), which provides accurate information about the accessory's supported features. The Apple device will use the information sent by this command to enable and disable custom commands.

The accessory must send the following AT+XAPL command after making a successful HFP Service Level Connection (SLC) to the Apple device. The accessory must send an AT+XAPL command first, before sending any of the additional Apple-specific commands described below.

15.1 HFP Command AT+XAPL

Description: Enables custom AT commands from an accessory.

Initiator: Bluetooth accessory

Format: AT+XAPL=*vendorID*-*productID*-*version*,*features*

Parameters:

- *vendorID*: A string representation of the hex value of the vendor ID from the manufacturer, without the `0x` prefix.
- *productID*: A string representation of the hex value of the product ID from the manufacturer, without the `0x` prefix.
- *version*: The revision of the software.
- *features*: A base-10 representation of a bit field. Available features are:
  - Bit 0 = reserved
  - Bit 1 = The accessory supports battery reporting (reserved only for battery operated accessories).
  - Bit 2 = The accessory is docked or powered (reserved only for battery operated accessories).
  - Bit 3 = The accessory supports Siri status reporting.
  - Bit 4 = the accessory supports noise reduction (NR) status reporting.

Apple Confidential Property of Apple Inc.

- All other values are reserved.

Example: AT+XAPL=ABCD-1234-0100,10 (Supports battery reporting and Siri status)

Response: +XAPL=iPhone,*features*

Pary Wu
Unigrand LTD
paywu@unigrand.com.tw

16. Bluetooth Connection

Accessories can automate the Bluetooth pairing process between an Apple device and any number of integrated Bluetooth accessory components if they also have a wired connection to the Apple device.

Additionally, some Apple devices can report changes in the Bluetooth connection status for an accessory's Bluetooth components. These notifications can be useful when the accessory wishes to determine whether it is connected to a particular Apple device via both Bluetooth and a Lightning connector. Accessories must implement this feature if they can maintain audio transport connections over both Bluetooth A2DP and other transports (see Multiple Audio Connections (page 366)).

16.1 Bluetooth Connection Requirements

The accessory must declare at least one Bluetooth component during identification to use this feature. The component(s) must comply with the requirements in Bluetooth (page 216).

The accessory must send an initial BluetoothComponentInformation (page 592) message within 2 seconds of receiving a IdentificationAccepted (page 587) message. Additionally, the accessory must also send a BluetoothComponentInformation (page 592) message to the device whenever the state of an identified accessory Bluetooth component changes. The accessory must report status for every identified Bluetooth component and not just status for the changed component. Conversely, after the initial BluetoothComponentInformation (page 592) message is sent, the accessory must not send additional ones unless there is a state change in one of its identified Bluetooth components.

In the case of Bluetooth components, the accessory must report the `ComponentIdentifier` of the identified component and `ComponentEnabled` parameter value of `true` if the Bluetooth component is ready for connections. Conversely, the accessory must send a BluetoothComponentInformation (page 592) message with `ComponentEnabled` set to `false` if the Bluetooth component is not ready for connections.

All accessories that support the Bluetooth Connection feature via iAP2 must send or receive the following iAP2 control session message(s):

- BluetoothComponentInformation (page 592)
- StartBluetoothConnectionUpdates (page 593)
- BluetoothConnectionUpdate (page 593)
- StopBluetoothConnectionUpdates (page 594)

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

16.2 Bluetooth Connection Usage

Pairing of Bluetooth components with an Apple device will automatically initiate if the accessory has successfully authenticated and the Apple device has not previously paired with the component. The accessory's Bluetooth component must be prepared to complete the pairing process as soon as the accessory has started identification.

The accessory must send a StartBluetoothConnectionUpdates (page 593) message to start the generation of component updates from the device. More than one component identifier may be specified if the accessory has multiple Bluetooth components with different MAC addresses. If the accessory later wishes to stop receiving component status notifications it may send a StopBluetoothConnectionUpdates (page 594) message to the device.

When updates are first started, one or more initial BluetoothConnectionUpdate (page 593) messages will be sent to the accessory to inform it of the starting connection state of all requested components. Subsequent notifications for a particular component always completely override prior ones. It is possible for the device to connect to the same component profile multiple times in a row. For example, this may occur when the device temporarily moves out of range for a period of time that is not long enough to trigger an automatic device disconnect and reconnects upon rediscovering the accessory component. Therefore, accessories must handle duplicate BluetoothConnectionUpdate (page 593) messages gracefully.

Accessories must send StopBluetoothConnectionUpdates (page 594) when they no longer have a need for them. For example, if an accessory makes use of Bluetooth connection updates purely to automate pairing of the accessory's Bluetooth component, it must stop asking for updates and not start them again upon subsequent connections to the Apple device.

Tomtop Technology LTD
Tomtop@tomtop-brand.com.tw

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

17. Bluetooth Headset Battery Level Indication

Any Hands-Free Bluetooth headset accessory can show its battery level to the user as an indicator icon in the Apple device status bar. This feature is supported on all Apple devices that support the Hands-Free Profile, including iPhone, iPod touch, and iPad.

Headset battery indication is implemented by two Apple-specific Bluetooth HFP AT commands, HFP Command AT+XAPL (page 233) and HFP Command AT+IPHONEACCEV (page 237)

17.1 HFP Command AT+IPHONEACCEV

Description: Reports a headset state change.

Initiator: Headset accessory

Format: `AT+IPHONEACCEV=`*Number of key/value pairs* `,` *key1* `,` *val1* `,` *key2* `,` *val2* `,...`

Parameters:

- *Number of key/value pairs*: The number of parameters coming next.
- *key*: the type of change being reported:
  - 1 = Battery Level
  - 2 = Dock State
- *val*: the value of the change:
  - Battery Level: string value between '0' and '9'
  - Dock State: `0` = undocked, `1` = docked

Example: `AT+IPHONEACCEV=1,1,3`

Parry Wu Bluetooth LTD parry@bluetooth.com.tw

18. Bluetooth Low Energy

The *Bluetooth 4.0* specification introduces Bluetooth Low Energy, a new wireless technology targeted for accessories with limited battery resources. If Bluetooth Low Energy is supported, the accessory must follow the guidelines in this section.

18.1 Role

The accessory must implement either the Peripheral role as defined in the *Bluetooth 4.0* specification, Volume 3, Part C, Section 2.2.2.3 or the Broadcaster role, as defined in Section 2.2.2.1.

18.2 Advertising Channels

The accessory must advertise on all three advertising channels (37, 38, and 39) at each advertising event. See the *Bluetooth 4.0* specification, Volume 6, Part B, Section 4.4.2.1.

18.3 Advertising PDU

The accessory must use one of the following advertising PDUs:

- ADV_IND
- ADV_NOCONN_IND
- ADV_SCAN_IND

ADV_DIRECT_IND must not be used. See the *Bluetooth 4.0* specification, Volume 6, Part B, Section 2.3.1.

18.4 Advertising Data

The advertising data sent by the accessory must contain at least the following information as described in the *Bluetooth Core Specification Supplement*, Part A:

- Flags

Parvinder Singh Wui
Universal Ltd
parvwui@universal.com.tw

- TX Power Level
- Local Name
- Services

The accessory may put the Local Name and the TX Power Level data in the SCAN_RSP PDU if, for example, it needs to reduce power consumption or not all of the advertising data fit into the advertising PDU. Note that, depending on its state, the Apple product may not always perform active scanning.

The primary services must always be advertised in the advertising PDU. Secondary services must not be advertised. Services not significant to the primary use case of the accessory may be omitted if space is limited in the Advertising PDU.

The advertising data and the scan response data in the SCAN_RSP PDU must comply with the formatting guidelines in the *Bluetooth 4.0* specification, Volume 3, Part C, Section 18: it starts with a length field, followed by AD Type and AD Data.

18.5 Advertising Interval

The advertising interval of the accessory must be carefully considered, because it affects the time to discovery and connect performance. For a battery-powered accessory, its battery resources must also be considered.

To be discovered by the Apple product, the accessory must first use the recommended advertising interval of 20 ms for at least 30 seconds. If it is not discovered within the initial 30 seconds, Apple recommends using one of the following longer intervals to increase chances of discovery by the Apple product:

- 152.5 ms
- 211.25 ms
- 318.75 ms
- 417.5 ms
- 546.25 ms
- 760 ms
- 852.5 ms
- 1022.5 ms
- 1285 ms

Note: Longer advertising intervals usually result in longer discovery and connect times.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

18.6 Connection Parameters

The accessory is responsible for the connection parameters used for the Low Energy connection. The accessory must request connection parameters appropriate for its use case by sending an L2CAP Connection Parameter Update Request at the appropriate time. See the *Bluetooth 4.0* specification, Volume 3, Part A, Section 4.20 for details. The connection parameter request may be rejected if it does not comply with all of these rules:

- *Interval Max * (Slave Latency + 1) ≤ 2 seconds*
- *Interval Min ≥ 20 ms*
- *Interval Min + 20 ms ≤ Interval Max*
- *Slave Latency ≤ 4*
- *connSupervisionTimeout ≤ 6 seconds*
- *Interval Max * (Slave Latency + 1) * 3 < connSupervisionTimeout*

If Bluetooth Low Energy HID is one of the connected services of an accessory, connection interval down to 11.25 ms may be accepted by the Apple product.

The Apple product will not read or use the parameters in the Peripheral Preferred Connection Parameters characteristic. See the *Bluetooth 4.0* specification, Volume 3, Part C, Section 12.5.

18.7 Privacy

The accessory must be able to resolve a Resolvable Private Address in all situations. Due to privacy concerns, the Apple product will use a Random Device Address as defined in the *Bluetooth 4.0* specification, Volume 3, Part C, Section 10.8.

18.8 Permissions

The accessory must not require special permissions, such as pairing, authentication, or encryption to discover services and characteristics. It may require special permissions only for access to a characteristic value or a descriptor value. See the *Bluetooth 4.0* specification, Volume 3, Part G, Section 8.1, fifth paragraph.

Paywall Unlocked FTD
paywallunlocked.com

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

18.9 Pairing

The accessory must not request pairing until an ATT request is rejected using the Insufficient Authentication error code. See the *Bluetooth 4.0* specification, Volume 3, Part F, Section 4 for details.

If, for security reasons, the accessory requires a bonded relationship with the Central, the Peripheral must reject the ATT request using the Insufficient Authentication error code, as appropriate. As a result, the Apple product may proceed with the necessary security procedures.

Similarly, if the Apple device acts as a Central and a GATT server, it may reject an ATT request using the Insufficient Authentication error code. The accessory must initiate the security procedure for pairing in response.

Pairing may require user authorization depending on Apple product. Once an accessory is paired with an Apple product, it must retain the distributed keys of both central and peripheral for future use. If the pairing is no longer required, the accessory must delete both sets of keys.

To incorporate authentication in an accessory design, the accessory must support the Authentication Specification of MFi program. For further information about MFi, see developer.apple.com/programs/mfi.

18.10 MTU Size

The Apple device supports and requests MTU size larger than default MTUs during the Exchange MTU Request handshake. See the *Bluetooth 4.0* specification, Volume 3, Part F, Section 3.2.8. The accessory may use this mechanism to negotiate larger MTU sizes.

18.11 Services

18.11.1 Generic Access Profile Service

The accessory must implement the Device Name characteristic per the *Bluetooth 4.0* specification, Volume 3, Part C, Section 12.1. The Device Name characteristic must be writeable.

18.11.2 Generic Attribute Profile Service

The accessory must implement the Service Changed characteristic only if the accessory has the ability to change its services during its lifetime.

The Apple product may use the Service Changed characteristic to determine if it can rely on previously read (cached) information from the device. See the *Bluetooth 4.0* specification, Volume 3, Part G, Section 7.1.

Property of Bluetooth SIG, Inc. and Bluetooth LTD
Unauthorized Copy

18.11.3 Device Information Service

The accessory must implement the Device Information Service. The service UUID for this service must not be advertised in the Advertising Data. The following characteristics must be supported:

- Manufacturer Name String
- Model Number String
- Firmware Revision String
- Software Revision String

18.11.4 Available Services

With iOS 7.0, any Apple device makes Battery Service, Time Service and Apple Notification Center Service (ANCS) available to an accessory. The Time Service supports the current time and local time information characteristics. The service does not provide an "Adjust Reason" when the current time changes. ANCS uses 7905F431-B5CE-4E99-A40F-4B1E122D00D0 as its UUID.

These services are not guaranteed to be available immediately after connection and the accessory must support Characteristic Value Indication of the Service Changed characteristic (see *Bluetooth 4.0* specification, Volume 3, Part G, Section 7.1) to be notified when the services become available. The Apple device will maintain a connection to an accessory as long as it is paired and uses one of the available services.

18.12 GATT Server

With iOS 6.0, applications may contribute services and characteristics to the GATT server that the Apple device makes available to the accessory. The recommendations in this section apply to the accessory in this case.

The following services are implemented internally by iOS and must not be published by third party iOS applications:

- Generic Attribute Profile Service
- Generic Access Profile Service
- Bluetooth Low Energy HID Service
- Battery Service
- Time Service
- Apple Notification Center Service

Unlicensed Apple Web HID
tsaiwu@unlicensed.com.tw

The Apple device implements the GAP Service Changed characteristic, because the database contents can change at any time. The accessory must therefore support the Characteristic Value Indication of this characteristic and, upon receiving indications, invalidate its database cache accordingly. See the *Bluetooth 4.0* specification, Volume 3, Part G, Section 7.1.

The accessory must minimize the use of ATT/GATT requests and commands and only send what is necessary. For example, do not use GATT Discover All Services when the accessory is looking for specific services. Use Discover Primary Service By Service UUID instead. Less airtime equals less power consumption and better performance for both the accessory and the Apple device.

When third party iOS applications discover services on the accessory, the following services are used internally by iOS and are filtered out from the list of discovered services:

- Generic Attribute Profile Service
- Generic Access Profile Service
- Bluetooth Low Energy HID Service
- Apple Notification Center Service

The accessory must be robust enough to handle any error gracefully. Pairing and Characteristic Value reads/writes may fail if the application that owns the service is not in the foreground and is not entitled to run in the background.

If an ATT Prepare Write Request is used, all queued attributes are contained within the same GATT Service.

Paul Wu
Paul@Digrand.com.tw
Digrand LTD

19. CarPlay

This chapter defines how a vehicle infotainment system (head unit) may receive and interact with digital content (audio, video, and metadata) streamed from an Apple device running iOS. It covers accessories that can receive streamed digital content from Apple devices.

To work with Apple CarPlay™, accessories must support the CarPlay feature.

CarPlay is only allowed for dashboard-integrated vehicle infotainment systems. It is not intended for rear-seat or non-vehicle integrations. The accessory must be connected to the vehicle audio system and must use an external microphone.

CarPlay accessories must only be marketed in countries where the CarPlay feature is available. For a list of countries where the CarPlay feature is available, see http://www.apple.com/ios/feature-availability/#applecarplay-applecarplay.

In the context of CarPlay, the term accessory is used to refer to either a vehicle and its head unit or an after-market head unit.

After-market head units must be integrated systems with a built-in display with a minimum 6 inch diagonal screen size. After-market head units that rely on the customer, an installer, or another third party to ensure spec compliance should make those requirements clear in their installation manual, documentation, etc. Examples of this include instructions on connecting vehicle speed sensors and external GNSS antennas, microphone placement, and placement of the Apple device for thermal considerations.

19.1 Additional Specifications

For clarifications and detailed specifications, this chapter cites portions of the following external specifications:

- USB 2.0 Specification (http://www.usb.org/developers/docs/usb20/)
- Apple Bonjour Support Site (https://www.apple.com/support/bonjour/)
- IETF RFCs
  - 793 (http://www.ietf.org/rfc/rfc793.txt)
  - 768 (http://www.ietf.org/rfc/rfc768.txt)
  - 2460 (http://www.ietf.org/rfc/rfc2460.txt)

Parkmate@Digital-Drive.com.tw

- 2474 (http://www.ietf.org/rfc/rfc2474.txt)
- 3966 (http://www.ietf.org/rfc/rfc3966.txt)
- 5905 (http://www.ietf.org/rfc/rfc5905.txt)

- IEEE Standard 802.11e-2005 (http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=10328)
- ITU H.264 Specification (http://www.itu.int/rec/T-REC-H.264)
- Universal Serial Bus Communications Class Subclass Specifications for Network Control Model Devices (http://www.usb.org/developers/docs/devclass_docs/)
- Universal Serial Bus Class Definitions for Communication Devices (http://www.usb.org/developers/docs/devclass_docs/)
- ITU-T P.1100: Narrow-band hands-free communication in motor vehicles (http://www.itu.int/rec/T-REC-P.1100-201103-I/)
- ITU-T P.1110: Wideband hands-free communication in motor vehicles (http://www.itu.int/rec/T-REC-P.1110-200912-I/)

19.2 General Requirements

19.2.1 Overview

Apple CarPlay lets an Apple device stream interactive digital content to an accessory with one or more audio/video endpoints and controls. A typical embodiment is shown in Figure 19-1 (page 246).

CarPlay is intended for use with a single front seat display. An accessory must not replicate digital video content to multiple displays.

For more information on how to integrate the Apple CarPlay identity in vehicle hardware and software and in marketing communications, refer to the *Apple CarPlay Identity Guidelines*.

Third Party Use Only Licensed to Yu Jiang (yujiang@unigrand.com.tw)

Figure 19-1 Topology of an automotive head unit using CarPlay

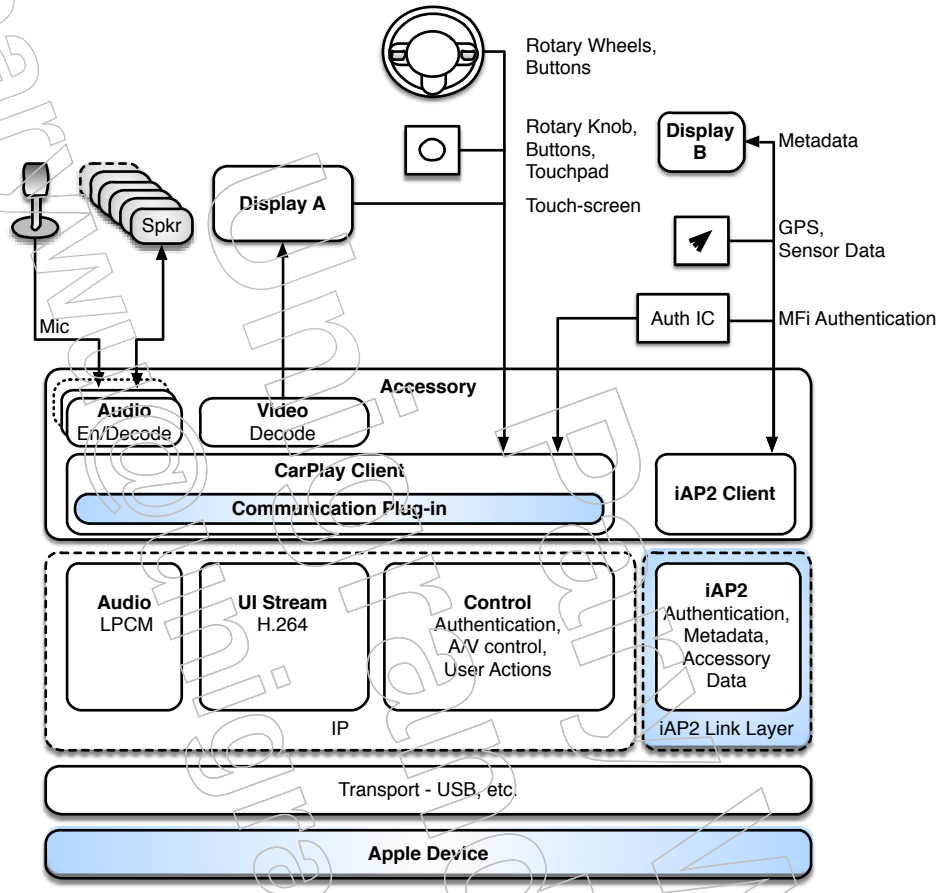


Figure 19-1 (page 246) shows an automotive head unit and its associated instrument cluster and displays with:

- Digital content that is independently driven by an Apple device
- User interactions that are controlled by one or more discrete controllers
- Multichannel audio, both input and output

19.2.2 Hardware Requirements

This section contains physical requirements for an accessory that supports CarPlay.

Partially Confidential
Licensed to [illegible]@gmail.com.tw

19.2.2.1 High Resolution Display

The accessory must provide a high-resolution display with a minimum resolution of 800 x 480, 24 bits of RGB color per pixel, 30 Hz refresh rate (60 Hz recommended) and all pixels must be visible to the end user. Note that future specifications may require at least 60Hz refresh rate. Displays with square pixels are recommended, but the Apple device can accommodate for non-square pixel displays as well. See User Interface (UI) Stream (page 258).

The minimum resolution of the display area used for CarPlay must be 800 x 480.

19.2.2.2 Processing

The accessory must be capable of:

- Hardware decoding of H.264 video, see User Interface (UI) Stream (page 258).
- Running the Communication Plug-in and other software clients, see Software Clients (page 256).
- Providing different audio processing based on the audio mode, see Audio (page 259).
- Synchronizing the audio output and input sample clocks, see Synchronization (page 286).

19.2.2.3 User Input Devices

The accessory must provide a touchscreen, or a combination of a rotary knob and at least select and back buttons. See User Input (page 295).

19.2.2.4 Speakers and Microphone

The accessory must provide audio output and input via the vehicle's speakers and a microphone in the vehicle cabin. See Audio (page 259).

19.2.2.5 Siri Button

Siri is an integral part of the CarPlay experience. Siri can be used in conjunction with the display to call people, select and play music, hear and compose text messages, and get directions. Siri can also be used to access Apple device features that do not appear on the display, including notifications, calendar information, reminders and more. Siri is one of the most important ways that users interact with CarPlay so it's critical that the behavior of Siri is consistent with what users are familiar with on Apple devices.

The accessory must have a tactile button for Siri activation. The Siri button cannot be a soft button that appears on a vehicle display. The Siri button is used to activate a Siri session and to continue a Siri session when there is an on-going conversation. The Siri button must be accessible at all times. Immediately after the Apple device has been connected to the accessory, users expect to be able to activate Siri at any time, regardless of what

Confidential and Proprietary Information

state the vehicle is in and even if CarPlay is not showing on the display. There are some exceptions, such as when the vehicle is performing a safety critical function or if there is already an active voice recognition session using the accessory's built-in system.

The Siri button must be easily accessible. It is expected that the Siri button is located on the steering wheel, although in some exceptional cases the Siri button may be located elsewhere. In many cases the Siri button will be the same button that is used to activate the accessory's built-in voice recognition system. In this case, a short press on the button should trigger the accessory's built-in voice recognition system and a long press on the button should trigger Siri. The accessory is responsible for defining what constitutes a long press on the button, but Apple recommends a period of 600 ms since it closely mimics the behavior of the Apple device home button. Once the user activates Siri by holding down the button, a Siri session is initiated. The accessory knows that a Siri session is active by observing the application state which will be "Speech". While the Siri session is active, the accessory must send all Siri button events (every time the Siri button is either pressed or released) to the Apple device. The Apple device monitors both button press events and button release events and manages the Siri session accordingly. The accessory must send raw events and should not attempt to interpret the button press and button release events. When the Siri session is terminated, the accessory may stop sending button events and return to a default state.

If the Siri button is a dedicated hardware button used exclusively for Siri and is never used for built-in voice recognition activation or for other features using other devices or systems, then you may choose to label the button with Siri branding. In this case, please consult with Apple for guidelines on using the Siri brand. Otherwise Siri should not appear on the button and it should be labeled with standard iconography, such as the icon used for the built-in voice recognition system.

See requestSiri (page 322).

19.2.2.6 "Apple CarPlay" Button

To show the CarPlay UI, the accessory must provide one of the following:

- A physical button.
- A soft button in a top-level menu.
- A menu item in a top-level menu.

The Apple CarPlay icon may be used on physical buttons that show the CarPlay UI. The label "Apple CarPlay" should appear in text below the icon. If the icon is not used, the text-only label "Apple CarPlay" must be used.

The Apple CarPlay logo may be used on soft buttons and menu items that show the CarPlay UI. If color graphics are featured in the accessory screen interface, the accessory should use the color version of the Apple CarPlay logo. If black-and-white graphics or other monochromatic themes are featured in the accessory screen interface,

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

use either the white version or the black version of the Apple CarPlay logo, matching the visual style of the interface. The logo must be accompanied by the label "Apple CarPlay". If the logo is not used, the text-only label "Apple CarPlay" must be used.

When using the label "Apple CarPlay", the accessory should match the font, size, style, and placement of other text used in the interface; do not imitate Apple typography. For example, if all capital letters are used on other items, then Apple CarPlay can also appear in all capital letters: "APPLE CARPLAY".

It is preferred that "Apple CarPlay" is placed on one line. If space is limited and other titles are stacked, "Apple CarPlay" can be stacked, with "Apple" on one line and "CarPlay" on the next line.

Any use of the Apple CarPlay icon, logo, or trademark within native UI requires Apple review and approval.

See requestUI (page 322) to show the CarPlay UI.

19.2.2.7 Sensors

The accessory must provide location information to the Apple device as described in Location Information (page 460).

19.2.2.8 Connection to the Apple device

To connect to the Apple device, the accessory must provide a USB Type A receptacle or a Lightning connector.

Refer to the *Apple CarPlay Identity Guidelines* for labeling any receptacles or connectors.

19.2.2.9 Device Mount Location

The Apple device should be mounted in an open holder that is away from direct sunlight or any heat source. Air ventilation is recommended but not required, however an enclosed compartment with no air circulation, such as the glove box, should be avoided.

19.2.3 Architecture

To support CarPlay, an accessory system must support USB as a USB device.

The accessory communicates with the Apple device using protocols defined by the Internet Protocol (IP) suite. An IP model is employed to abstract data transport away from the chosen physical layer where possible, although some aspects of the overall implementation will differ per transport. In addition, the accessory must establish an iAP2 link with the Apple device for authentication, identification, and exchange of metadata.

The major building blocks of this architecture are illustrated in Figure 19-1 (page 246), with Apple-provided software components highlighted in blue.

Paywall@uniplan.com Paywall Unplan LTD

19.2.3.1 CarPlay over USB

At a minimum, the accessory must support Hi-Speed USB; see the *USB 2.0 Specification*. Because of the bandwidth and low-latency requirements for transferring interactive audio/visual content, the use of a dedicated USB controller for the interface between the Apple device and accessory is recommended.

The accessory must also support USB Host Mode (where the Apple device is the host), as specified in USB Host Mode (page 527).

USB Network Control Model (NCM) is defined in the following specifications:

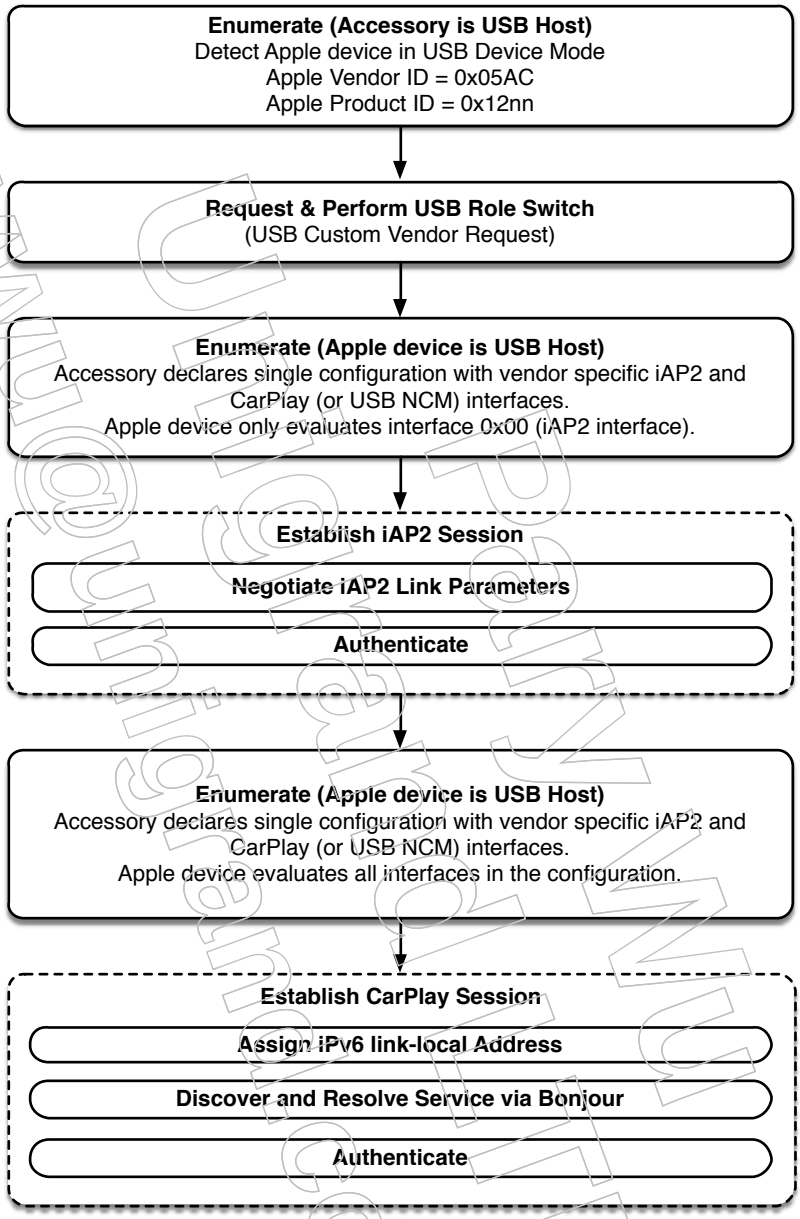
- *Universal Serial Bus Communications Class Subclass Specifications for Network Control Model Devices, Revision 1.0*
- *Universal Serial Bus Class Definitions for Communication Devices, Version 1.2*

The accessory must establish the CarPlay session automatically upon physical connection to the Apple device.

For an accessory that normally acts as a USB host, the steps shown in Figure 19-2 (page 251) are required.

Paly Wu
Unigrand LTD
paly.wu@unigrand.com.tw

Figure 19-2 Process for establishing a CarPlay session



19.2.3.1.1 Role Switch

If the accessory normally acts as a USB host, it must perform a host-to-device role switch as specified in USB Role Switch (page 529).

An accessory that presents directly as a USB device without undergoing the host-to-device role switch, will require the use of a custom USB cable or dock solution, incorporating the Lightning (C10A) or Lightning (C11A) connector (USB Host Mode). The standard Apple-provided white USB-to-Lightning cable is incompatible with a USB accessory presenting directly as a USB device.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

19.2.3.1.2 iAP2 / NCM Interface Configuration

Once role switch is completed, the accessory must continue by enumerating the supported USB interfaces. At a minimum the accessory must present a single configuration which includes:

- An iAP2 interface (0x00) (see USB Host Mode (page 527)).
- A USB NCM control interface (0x01 or higher) (see Table 19-1 (page 252) and Table 19-2 (page 253)).
- A USB NCM data interface (0x01 or higher) (see Table 19-3 (page 253)).

The accessory must proceed to establish an authenticated iAP2 Session as described in Accessory Authentication (page 201).

> **Note:** Until authentication is completed successfully, the Apple device will only enumerate the interface with descriptor 0x00 (the iAP2 interface) unless it is using control session version 2 (see Control Session (page 562)). With version 2, the Apple device may enumerate other USB interfaces once device identification is complete whether or not authentication is complete.

After iAP2 authentication is complete, the accessory must again present a single configuration which includes the same 3 interfaces.

Table 19-1 USB NCM Control Interface Descriptor

USB Descriptor	Value	Description
Interface Number	0x01-0xnn	Must not be 0x00 (the iAP interface). Note: The interface number must match the USBHostTransportCarPlayInterfaceNumber (see Table 52-14 (page 583)).
Interface Class	0x02	USB Communication Interface Class
Interface Subclass	0x0D	Network Control Model
Interface Protocol	0x00	No encapsulated commands / responses
Number of Endpoints	1	1 Interrupt IN (optional): This is typically used to convey changes in link status. Since link is expected to be maintained at all times, we will synthesize link up if there is a read completion via the data interface.

The accessory must also publish the additional descriptors outlined in Table 19-2 (page 253).

Apple Confidential Information

Table 19-2 USB NCM Communication Interface Descriptor Requirements

USB Descriptor	Description
Header	CDC Header functional descriptor
Union	CDC Union functional descriptor
Ethernet	CDC Ethernet Networking functional descriptor
NCM	NCM functional descriptor

Head unit MAC addresses must be properly assigned by the manufacturer, see http://www.iana.org/assignments/ethernet-numbers/ethernet-numbers.xhtml. The head unit must provide this MAC address for the Apple device to use via the CDC Ethernet Networking functional descriptor. The head unit may modify the address for its own use by setting the locally administered bit of the Organizationally Unique Identifier (OUI) portion of the MAC address, the second-least-significant bit of the most significant byte of the address, see http://standards.ieee.org/develop/regauth/tut/macgrp.pdf. The head unit must not use the un-modified MAC address. This will ensure that there are no collisions between the Apple device and the head unit.

Table 19-3 USB NCM Data Interface Descriptor

USB Descriptor	Value	Description
Interface Class	0x0A	USB Data Interface Class
Interface SubClass	0x00	
Interface Protocol	0x01	NCM Data Class
Number of Endpoints	0	(for Alternate Setting 0)
Number of Endpoints	2	(for Alternate Setting 1) 1 Bulk IN; and 1 Bulk OUT

The accessory must implement USB Hi-Speed NCM on this interface. The interface must support transfers of encapsulated datagrams up to 64 KB (i.e., up to 40 1514-byte Ethernet frames) and 16-bit NCM Transfer Blocks (i.e., NTB-16).

Accessories using the USB NCM interface for CarPlay must support at least 100 Mbps of bandwidth with a latency less than 5ms over both TCP and UDP protocols and a packet loss of less than 1% (as measured with iperf) over UDP.

When the Apple device is connected or disconnected, the accessory must reflect this change in the connection state of the NCM interface. The network stack on the head unit must mark the NCM interface as available only while the Apple device is connected.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

An example configuration for an accessory implementing the USB Communications Class NCM Subclass descriptors is as follows:

```
Device Release                                          :   r2.00
USB Spec Release                                        :   v2.00
Serial Number                                          :   ABC-0123456799
Class                                                   :   0xff
(Vendor-specific)
Subclass                                                :   0x00
Protocol                                                :   0x00
Configurations                                          :   1
    Configuration                                       :   1 (Default
Configuration)
        Attributes                                      :   0xc0
(Self-powered)
        Max Power                                       :   0 mA
        Interfaces                                      :   3
            Interface                                   :   0 / 0 (iAP2
Interface)
                Class                                   :   0xff
                Subclass                                :   0xf0
                Protocol                                :   0x00
                Endpoints                               :   2
                    Endpoint                            :   0x81
                        Attributes                      :   Bulk/IN
                        Max Packet Size                 :   512
                    Endpoint                            :   0x1
                        Attributes                      :   Bulk/OUT
                        Max Packet Size                 :   512
            Interface                                   :   1 / 0 (NCM
Communication Interface)
                Class                                   :   0x02
                Subclass                                :   0x0d
                Protocol                                :   0x00
                Endpoints                               :   1
                    Endpoint                            :   0x82
                        Attributes                      :   Interrupt/IN
```

Pary Wu LTD Unigrand
paryvu@unigrand.com.tw

```
                    Max Packet Size                 :   64
                    Interval                        :   1 mframe
                CDC Header Functional Descriptor    :
                    bcdCDC                          :   v1.10
                CDC Union Functional Descriptor     :
                    Control Interface               :   1
                    Subordinate Interface           :   2
                CDC Ethernet Networking Functional Descriptor   :
                    iMacAddress                     :   7
                    Ethernet Statistics             :   0
                    Max Segment Size                :   1514
                    Number MC Filters               :   0
                    Number Power Filters            :   0
                NCM Functional Descriptor           :
                    NCM Version                     :   v1.00
                    Network Capabilities            :   0x3A
            Interface                               :   2 / 0 (NCM
Data Interface Alternate 0)
                Class                               :   0x0a
                Subclass                            :   0x00
                Protocol                            :   0x01
                Endpoints                           :   0
            Interface                               :   2 / 1 (NCM
Data Interface Alternate 1)
                Class                               :   0x0a
                Subclass                            :   0x00
                Protocol                            :   0x01
                Endpoints                           :   2
                    Endpoint                        :   0x83
                        Attributes                  :   Bulk/IN
                        Max Packet Size             :   512
                    Endpoint                        :   0x3
                        Attributes                  :   Bulk/OUT
                        Max Packet Size             :   512
```

Pary Wu
pary.wu@eGrand.com.tw
eGrand LTD

19.2.3.1.3 Authentication

CarPlay is an authenticated solution, requiring the use of an MFi authentication coprocessor (2.0 B or later) obtained through Apple. Apple devices will stream only to authorized accessories.

Communication over both the iAP2 and CarPlay interfaces requires authentication and each interface provides a discrete authentication API. To expedite these dual authentication steps, local caching of the X.509 Certificate provided by the Apple Authentication Coprocessor is permitted.

All authenticated accessories are required to be certified under the Apple MFi program. The CarPlay accessory must successfully pass compliance tests to assure that all digital content from an Apple device will be correctly decoded and displayed and that all electrical requirements described in this specification are met.

19.2.3.1.4 Networking and Service Discovery

To establish a CarPlay session, the accessory must be networked with the Apple device by using the communication protocols defined by the Internet Protocol (IP) documentation; see *IETF RFC 2460, Internet Protocol, Version 6 (Ipv6) Specification* , December 1998. IP connectivity must be provided by IPv6 link-local addressing and the accessory must support Apple Bonjour zero-configuration networking over this interface; see the *Apple Bonjour Support Site* . The accessory will be identified uniquely by its MAC address and by a TXT record with additional data needed to resolve or use the service.

For more details, see Discovery (page 269).

19.2.3.1.5 Session Establishment

Once a connection has been established, setup and content transfer will start after the accessory completes authentication over the CarPlay interface.

The accessory must not send a 'Play' command automatically upon connection of a CarPlay enabled device. See Media Library Playback Requirements (page 473).

For more information on setting up a CarPlay session refer to Setup and Control (page 270).

The accessory must be able to establish a CarPlay session within 3 seconds of Apple device connection.

19.2.3.1.6 Session Termination

The accessory must be able to detect a disconnect and terminate the session within 1 sec of a physical detachment of the Apple device.

19.2.3.2 Software Clients

The accessory must implement iAP2 and CarPlay clients to support the message protocols and states specific to each interface.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Portions of both clients are provided as reference source code by Apple, namely:

- An iAP2 link layer
- A CarPlay Communication Plug-in, see CarPlay Communication Plug-in (page 324)

19.2.3.2.1 iAP2 Client

At a minimum, the iAP2 client must support:

- Accessory Authentication (page 201)
- Accessory Identification (page 205)
- Power (page 480)
- Location Information (page 460) and provide location information from GPS and other sensor data.

The location information will be used to augment the Apple device's built-in location services and help conserve device power.

Additionally, the iAP2 interface enables transfer of metadata and state information to the accessory from select iOS apps running in CarPlay, such as Music and Telephony. This information may be used to complement the CarPlay experience on an additional accessory screen that is not the primary CarPlay screen; for instance, to display call information on a vehicle instrument cluster display.

Accessories may also provide sensor data and other vehicle status to the Apple device, see Vehicle Status (page 532).

19.2.3.2.2 CarPlay Client

The CarPlay client must implement the logic described in Resource Management (page 287) for resource sharing between the accessory and Apple device. It must also manage user events and the transfer of digital UI and audio between the accessory and the Apple device.

Payload data for the CarPlay client may be encoded according to a variety of standards dependent on the content type, as described in Media Types and Formats (page 258), but the underlying transport layer employed by the Communication Plug-in will consist of one or more channels implementing either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). For TCP, see *IETF RFC 793, Transmission Control Protocol*, September 1981. For UDP, see *IETF RFC 768, User Datagram Protocol*, August 1980.

The Communication Plug-in will open a number of additional UDP and TCP channels in parallel to handle synchronization, retransmission requests, usage reports for user actions and other custom commands and controls.

The accessory must provide a networking stack that supports multiple concurrent UDP and TCP channels for:

Confidential Apple Inc.
Apple Inc. Confidential

- audio/video transfer
- command and control
- clock synchronization and retransmission

19.2.4 Media Types and Formats

Video content (a User Interface stream) will stream from the Apple device on a dedicated channel. Two additional dedicated channels are required for audio: main audio (input and output) and alternate audio.

All control and video data packets are encrypted (AES-128). The task of encryption/decryption is abstracted for the accessory maker within the Apple-provided Communication Plug-in code.

Apple devices will not stream content subject to DRM (Digital Rights Management) over CarPlay.

19.2.4.1 User Interface (UI) Stream

The UI is streamed from the Apple device as individual, H.264 NAL units configured with AVCC header data and an additional presentation timestamp for scheduling and synchronization.

The AVCC Format is specified in *ISO/IEC STANDARD 14496-15 (AVC file format)* section 5.2.4.1.1. If the accessory decoder does not natively support AVCC format, but instead requires Annex-B format, a set of utility functions are provided in the Communication Plug-in to assist with format conversion. Note however, that this conversion, if required, is software based, not hardware accelerated, and may impact performance and latency.

The H.264 Sequence Parameter Set (SPS) and Picture Parameter Set (PPS) are only sent once, at the beginning of the stream.

In order to minimize video encoding and decoding latency, no frame reordering is used (i.e. no B-frames). The SPS includes a Video Usability Information (VUI) header with the bitstream_restriction_flag set, max_num_reorder_frames of 0, and max_dec_frame_buffering set to the maximum Decoded Picture Buffer (DPB) size.

Video is encoded using full range Y'CbCr, black is 0, white is 255. The H.264 SPS has the video_full_range_flag set to reflect this. See Annex E of the *H.264* specification for more information.

The accessory's target resolution, max frame rate, and physical dimensions for the CarPlay content must be reported via the Info Message (page 273). Resolutions natively supported by the Apple device and its set of CarPlay compatible applications are listed in Table 19-4 (page 259).

Pearl Digital Private Limited
pearlpune@pearlpune.com

The accessory must implement an H.264 decoder capable of supporting at least High Profile 3.1 and content with a 800 x 480 resolution at 30 fps using YCbCr 4:2:0 chroma subsampling, but higher frame rates are strongly recommended. The decoder must be capable of supporting the H.264 level corresponding to the resolution and frame rate reported to CarPlay. See Table 19-5 (page 259).

Table 19-4 CarPlay H.264 Video Stream Standard Resolutions

Aspect Ratio	Pixel Dimensions	Frame Rate
16:9	960 x 540	60 fps (recommended), 30 fps
15:9	800 x 480	60 fps (recommended), 30 fps

Table 19-5 CarPlay H.264 Levels

H.264 Level	Max Bit Rate	Examples
High Profile 3.1	17.5 Mbps	800x480@30fps 800x480@60fps 960x540@30fps
High Profile 3.2	25 Mbps	960x540@60fps

The accessory's digital video display must natively support one of the standard resolutions encoded by the Apple device in either a full-screen or windowed configuration without resorting to stretching. When the CarPlay content is displayed as a window within a larger accessory display, the accessory must advertise the resolution and physical dimensions of this window when setting up the stream with the Apple device, not the actual (larger) display resolution and display dimensions. See High Resolution Display (page 247) for display requirements.

Accessory makers wishing to offer different target resolutions for the UI stream and/or nonsquare pixel displays must first consult with Apple for compatibility.

The CarPlay UI must appear on the accessory display within 500 ms of either:

- The accessory receiving a screen stream setup request.
- The accessory receiving a session configuration that includes a screen stream.

19.2.4.2 Audio

The accessory must support audio input and output streams encoded in the form of uncompressed LPCM. Volume control and display of volume control UI are the responsibility of the accessory because the Apple device provides only unattenuated line-level audio output.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

See Volume Management (page 267) for specific requirements and recommendations.

The accessory must support one main (input and output) and one alternate audio stream from the Apple device as described in Resources (page 287). Alternate audio must be mixed with main audio regardless of whether the source of main audio is the Apple device (for example, the iOS Music app) or the accessory (for example, the FM tuner on a vehicle head unit). In some use cases output on alternate audio might be accompanied by a request to duck the main audio channel (see Ducking (page 268)).

The audio sample rates supported by the accessory must be reported using the Info message, see Info Message (page 273).

Accessories that implement the CarPlay feature must respect the requirements from Multiple Audio Connections (page 366). In addition, they must use the CarPlay audio stream when using the feature and cannot use USB Host Mode Audio Transport in parallel.

The Apple device will at times simultaneously employ different sample rates for the main (input and output) and alternate audio streams, or may require sample rate transitions, for instance when transitioning main audio from music to a phone call. Main and alternate audio must be capable of maintaining independent sample rates. For example, if the main audio channel is operating at 16 kHz for telephony, alternate audio should continue to operate at a higher sample rate, e.g. 44.1 kHz, to preserve the fidelity of alerts and other alternate audio that would mix with phone call audio. All sample rate transitions will be communicated to the accessory by the Apple device using the Setup Message (page 280).

Also, the Apple device will communicate the intended use of the audio stream to allow the accessory to: (i) adjust its microphone audio processing (such as noise reduction, echo cancellation, frequency response, and gain), and (ii) suppress or eliminate additional noise sources (for example, temporarily turning down the defroster in a vehicle to reduce ambient noise).

After the Apple device sends a Setup Message (page 280) to request main (input and output) or alternate audio streams, the accessory must reply to the setup message within 100 ms and only after:

- It is ready to begin receiving and outputting audio samples (if main audio with input and output was requested). For output, the accessory must output all samples given to it after an acknowledgement of the setup message. For input the accessory must be ready to send audio samples to the Apple device after the acknowledgment of the setup message.
- It is ready to begin to continuously receive audio from the Apple device (if main or alternate audio output was requested). For output, the accessory must output all samples given to it after an acknowledgement of the setup message.

The accessory must support full-duplex audio on the main audio stream at the same sample rate in both input (microphone) and output directions. In addition, input and output must be driven from the same clock to avoid the need for asynchronous sample rate conversion.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

At no time during interaction with the Apple device must any popping, clicking, or other audible artifacts occur on the accessory, whether it be during transitions between shared resource ownership or mode changes, starting and stopping IO, initiation or cessation of ducking, or any other circumstance. Additionally, volume changes and ducking must be free any zippering or other audible artifacts. The output to the DAC for an unencoded LPCM audio signal sent from the Apple device should be bit-identical to that of an audio signal sent from other sources, assuming those sources allow direct digital transfer.

19.2.4.2.1 Main Audio - Entertainment

Sample Rate

Table 19-6 (page 261) details the audio stream requirements. The accessory must support both 44.1 kHz 16-bit and 48 kHz 16-bit audio. At least one of these two sample rates must be natively supported, that is, a sample rate at which the accessory can perform its audio processing without sample rate conversion. See Table 19-18 (page 277) for a full list of possible audio configurations.

Table 19-6 Entertainment Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
44.1 kHz	16 bit	stereo	n/a	Required
44.1 kHz	24 bit	stereo	n/a	Recommended
48 kHz	16 bit	stereo	n/a	Required
48 kHz	24 bit	stereo	n/a	Recommended

Latency

Latency from the receipt of a buffer of LPCM audio by the accessory from the CarPlay Communication Plug-In to the time it is emitted by the accessory's speaker must not exceed 35 ms.

19.2.4.2.2 Main Audio - Telephony

Unless otherwise specified, the accessory must meet the recommendations and test criteria in International Telecommunication Union Recommendations *ITU-T P.1100: Narrowband hands-free communication in motor vehicles* and *ITU-T P.1110: Wideband hands-free communication in motor vehicles*, whichever applies to the current use case.

Proof of compliance with the *ITU-T P.1100* and *ITU-T P.1110* specifications must be submitted to Apple as part of the self-certification audit process.

Sample Rate

Apple Confidential Information
Unauthorized Distribution Prohibited

Table 19-7 (page 262) details the audio stream requirements. The accessory must accept and provide the following sample rates with equal or higher bit depth.

Table 19-7 Telephony Audio Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
8 kHz	16 bit	mono	full-duplex	Required
16 kHz	16 bit	mono	full-duplex	Required

In the future, voice communication standards may incorporate even higher sample rates such as 32KHz and 48KHz, and also stereo.

Audio Processing

The accessory must perform echo cancellation and noise processing tuned for telephony and provide full-duplex audio that is free of echoes, noise, or other artifacts.

Uplink Output Level

In the nominal test arrangement, the accessory must provide a constant average uplink output level to the device. The use of automatic gain control (AGC) targeting the specified level is recommended. The nominal level measured at the uplink output of the accessory must be A-weighted -26 +/- 2 dB root-mean-square (RMS), expressed in units relative to full-scale (dBFS(A)).

Signal-to-Noise Ratio

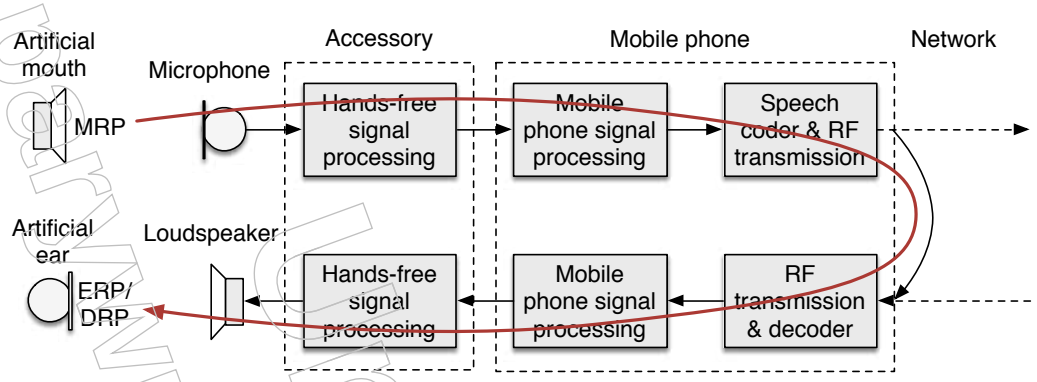
The accessory must maintain adequate signal-to-noise ratio (SNR) in its uplink output for voice communication purposes. The absolute SNR must be over 12 dB in all typical driving conditions as specified in Annex D of *ITU-T P.1100* /*ITU-T P.1110* . Higher values are preferred and the SNR should be much higher in a stationary vehicle and at slow driving speeds.

Round-Trip Delay

The definition of the combined round-trip delay of the accessory and the device follows *3GPP TS 26.132* , consisting of the sum of uplink and downlink processing latency of all acoustic, analog, and digital components of the accessory and the device, including any digital buffering. The round-trip path is illustrated in Figure 19-3 (page 263).

Partwu@unionchip.com

Figure 19-3 Telephony Audio Round-Trip Path



The combined round-trip delay of the accessory and the device must not exceed 275 ms. Smaller values are preferred and ideally the delay should be under 200 ms.

Send Frequency Response

The send frequency response must meet the requirements in *ITU-T P.1100*, Clause 11.4.1 for narrowband telephony and *ITU-T P.1110*, Clause 11.4.1 for wideband telephony.

19.2.4.2.3 Main Audio - FaceTime Audio

Unless otherwise specified, the accessory must meet the recommendations and test criteria specified in *ITU-T P.1110*.

FaceTime Audio operates at 24KHz and uses high-quality audio codecs.

Sample Rate

Table 19-8 (page 263) details the audio stream requirements. The accessory must accept and provide the following sample rate with equal or higher bit depth.

Table 19-8 FaceTime Audio Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
24 kHz	16 bit	mono	full-duplex	Required

In the future, even higher sample rates such as 32KHz and 48KHz and stereo may be used.

Audio Processing

Tairyu Wu (tairyu@ubiquitandi.com) Ubiquitandi LTD

The accessory must perform echo cancellation and noise processing tuned for telephony and provide full-duplex audio that is free of echoes, noise, or other artifacts. Due to high-quality audio coding, it is important to preserve the quality of the uplink signal. Less aggressive noise suppression than in telephony should be used if it results in better speech quality in the send direction (see Clause 12.5.1 in *ITU-T P.1110*).

Uplink Output Level

In the nominal test arrangement, the accessory must provide a constant average uplink output level to the device. The use of an AGC targeting the specified level is recommended, but the device should preserve linear operation for any constant speech level. The nominal level measured at the uplink output of the accessory must be A-weighted -26 +/- 2 dB root-mean-square (RMS), expressed in units relative to full-scale (dBFS(A)).

Signal-to-Noise Ratio

The accessory must maintain adequate signal-to-noise ratio (SNR) in its uplink output for voice communication purposes. The absolute SNR must be over 12 dB in all typical driving conditions as specified in Annex D of *ITU-T P.1110*. Higher values are preferred and the SNR should be much higher in a stationary vehicle and at slow driving speeds.

Round-Trip Delay

The round-trip delay must meet the same requirements as defined for Telephony (see Main Audio - Telephony (page 261)).

Send Frequency Response

The normalized send frequency response is measured from the mouth reference point (MRP) to the uplink output of the accessory (input to the device). The normalized send frequency response must be within the limits of the tolerance mask shown in Table 19-9 (page 264). The mask (see Figure 19-4 (page 265)) is drawn by straight lines between the breaking points in the table on a linear (dB sensitivity) - logarithmic (frequency) scale. Ideally, the response characteristics of the accessory should be flat in the frequency range of 100 Hz - 11 kHz. Extending the audio bandwidth to above 8 kHz is strongly recommended.

Table 19-9 Audio Tolerance Mask Limits For Send Frequency Response

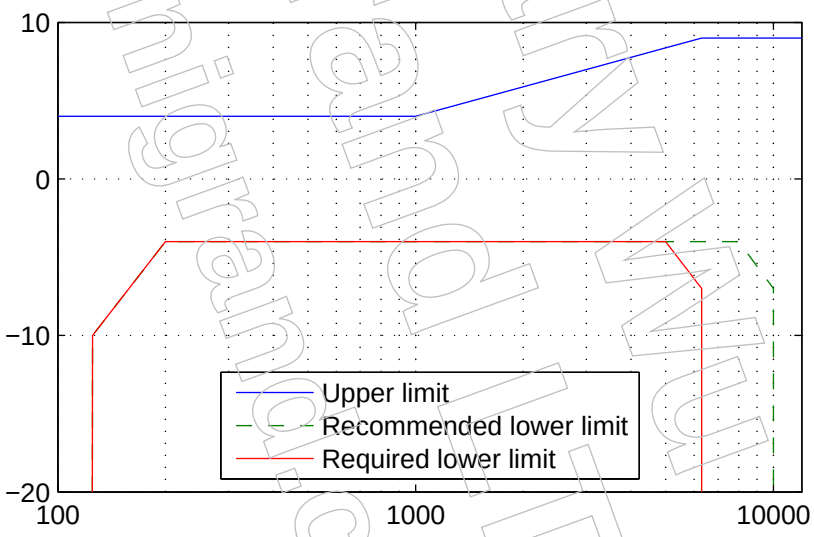
Frequency	Upper Limit	Recommended Lower Limit	Required Lower Limit
100 Hz	4 dB	-∞ dB	-∞ dB
125 Hz	4 dB	-10 dB	-10 dB
200 Hz	4 dB	-4 dB	-4 dB
1,000 Hz	4 dB	-4 dB	-4 dB

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Frequency	Upper Limit	Recommended Lower Limit	Required Lower Limit
5,000 Hz	(* See note.)	-4 dB	-4 dB
6,300 Hz	9 dB	-4 dB	-7 dB
8,000 Hz	9 dB	-4 dB	-∞ dB
10,000 Hz	9 dB	-7 dB	-∞ dB
12,000 Hz	9 dB	-∞ dB	-∞ dB

* Note: The limits for intermediate frequencies lie on a straight line drawn between the given values on a linear (dB) - logarithmic (Hz) scale. See Figure 19-4 (page 265)

Figure 19-4 Audio Tolerance Mask For Send Frequency Response (Sensitivity (dB) vs. Frequency (Hz))



19.2.4.2.4 Main Audio - Siri

Unless otherwise specified, the accessory must meet the recommendations and test criteria specified in *ITU-T P.1110*.

Sample Rate

Table 19-10 (page 266) details the audio stream requirements. The accessory must accept and provide the following sample rate with equal or higher bit depth.

Apple CarPlay Wireless Interface ID

Table 19-10 Siri Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
24 kHz	16 bit	mono	full-duplex	Required

Audio Processing

Directional microphones and linear beamforming with microphone arrays giving improved SNR are recommended. Linear echo cancellation for reducing unwanted audio sources (such as audio output from the system) without having any other effect on the speech signal are also recommended. However, single channel noise reduction methods (such as spectrum subtraction) must not be applied, as they will be detrimental to the speech recognition accuracy. Similarly, automatic gain control, residual echo suppression and attempts to blank out non-speech periods in the waveform must not be applied.

Uplink Output Level

In the nominal test arrangement, the accessory must provide a constant uplink output level to the device. The use of an AGC is not recommended. If the accessory adjusts signal gain, the gain must be held constant across each spoken utterance. The nominal level measured at the uplink output of the accessory must be A-weighted -26 +/- 2 dB root-mean-square (RMS), expressed in units relative to full-scale (dBFS(A)).

Signal-to-Noise Ratio

The accessory must maintain adequate signal-to-noise ratio (SNR) in its uplink output for speech recognition purposes. To maintain high speech recognition accuracy, the absolute SNR should be over 20 dB and all reasonable efforts should be made to maintain high SNR in all typical driving conditions as specified in Annex D of *ITU-T P.1110*. Values above 20 dB are recommended, and the SNR should be at least 25 dB in a stationary vehicle and at slow driving speeds.

Latency

Latency from the reception of sound at the accessory's microphone and the receipt by the CarPlay Communication Plug-In of a buffer of LPCM audio representing that sound must not exceed 60 ms. Likewise, latency from the receipt of a buffer of LPCM audio by the accessory from the CarPlay Communication Plug-In to the time it is emitted by the accessory's speaker must not exceed 35 ms.

Send Frequency Response

The normalized send frequency response for Siri must meet the same requirements as defined for FaceTime. See Main Audio - FaceTime Audio (page 263)

CarPlay Communication Plug-In R47 Interface Specification (Preliminary)

19.2.4.2.5 Alternate Audio - Alerts, Notifications, etc.

Sample Rate

Table 19-11 (page 267) details the audio stream requirements. The accessory must support both 44.1 kHz 16-bit and 48 kHz 16-bit audio. At least one of these two sample rates must be natively supported, that is, a sample rate at which the accessory can perform its audio processing without sample rate conversion. See Table 19-18 (page 277) for a full list of possible audio configurations.

Table 19-11 Alternate Audio Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
44.1 kHz	16 bit	mono or stereo	n/a	Required
44.1 kHz	24 bit	mono or stereo	n/a	Recommended
48 kHz	16 bit	mono or stereo	n/a	Required
48 kHz	24 bit	mono or stereo	n/a	Recommended

Latency

Latency from the receipt of a buffer of LPCM audio by the accessory from the CarPlay Communication Plug-In to the time it is emitted by the accessory's speaker must not exceed 35 ms.

19.2.4.2.6 Volume Management

The accessory's volume controls (physical or on-screen) must control the volume of audio from the Apple device.

If the accessory supports maintaining separate volumes for different types of audio, it is recommended that it does so using the following volume categories, with corresponding mappings to main audio stream audioTypes (see Table 19-27 (page 282)), and alternate audio:

- Entertainment - main audio "default" and main audio "media"
- Ringtone - main audio "alert"
- Phone Calls - main audio "telephony"
- Voice Prompts and Notifications - main audio "speech recognition" and alternate audio

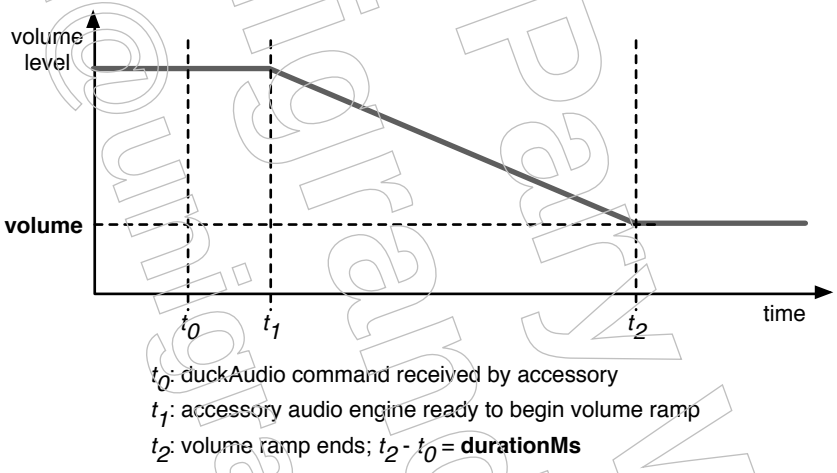
For example, were the user to change the volume while listening to music from the device, the accessory would remember that volume for the Entertainment volume category, and subsequently restore that volume the next time the main audio audioType was configured for "default" or "media".

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

19.2.4.2.7 Ducking

The accessory must support ducking, the process of ramping the main audio volume down so the user can more clearly hear an announcement or alert from the alternate audio channel. This ramping must be free of any zippering or other audible artifacts. For use cases where ducking is required, the device will send a "duckAudio" command to the accessory. For example, if the user is listening to music and they are getting close to their destination, the music may be ducked (attenuated) so that another stream can say "Arriving at your destination." Ducking fades the main audio volume down to a lower level so it is not an abrupt change. When the announcement is done, the original stream is unducked by fading back to the original level.

Figure 19-5 Event timeline for audio ducking/unducking



The duck/unduck ramp (t1 - t0) must start within a very short time after the message has been received. Typical and recommended times are 10-20 ms; response within 75 ms is required.

19.3 CarPlay Communication Protocol

The CarPlay communication protocol provides a mechanism for an accessory and an Apple device to:

- Discover and connect to one another on a network
- Setup and control audio and video (UI) streaming
- Arbitrate ownership and use of the accessory's shared resources
- Communicate the Apple device's app state to avoid conflicts that would lead to undesirable user experiences.
- Exchange user input events

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Implementation of this protocol must incorporate the reference implementation provided in the CarPlay Communication Plug-in.

The CarPlay communication protocol is flexible in its parsing of parameter values. CarPlay message parameters that are numbers, enum values, or booleans may be provided as strings that parse as those data types.

19.3.1 Discovery

CarPlay accessories advertise themselves on the network via Bonjour. Apple devices browse for them and if any are found and determined to be compatible, they are displayed to the user for playing to.

Table 19-12 (page 269) defines the CarPlay Bonjour service and its TXT record keys. The CarPlay communication plug-in will interface to the accessory's Bonjour implementation to configure and broadcast these values.

Table 19-12 CarPlay Bonjour Service Keys

Key	Required?	Description
deviceid	Y	Globally unique ID for the accessory; e.g., the primary MAC address, such as "00:11:22:33:44:55".
features	Y	Internally defined for Apple CarPlay. Value must not be modified.
model	Y	Model name of the accessory; must be globally unique and uniquely identify the product. This should be the model name of the vehicle for OEM integrations or an aftermarket unit's model name.
protovers	N	Protocol version string <major>.<minor> (e.g., "1.0"). Missing key defaults to "1.0".
srcvers	Y	Apple CarPlay source version assigned by Apple, in the form x.y.z .
flags	Y	Status of an operation. Value must not be modified. See Table 19-13 (page 269).

Table 19-13 Status flags bitfield enum values

Value	Bit	Description
0x01	0	Problem has been detected.
0x02	1	Device is not configured.
0x04	2	Audio cable is attached.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

The Bonjour service type for CarPlay is _airplay._tcp. Additional data needed for discovery-time metadata is advertised via a TXT record containing fields for feature detection, model information, versions, etc.

The accessory must enable the Bonjour DirectLink optimization. This will optimize the Bonjour discovery process and speed up the connection establishment time over the NCM interface.

19.3.2 Setup and Control

The Communication Plug-in in the CarPlay Client establishes a control channel for the primary tasks of:

- Configuring the UI stream to one of the target screen resolutions listed in Media Types and Formats (page 258).
- Scheduling the video signal to a synchronized clock between both accessory and Apple device (a microsecond resolution clock or better is required on the accessory).
- Configuring and managing the main and alternate audio streams, including callbacks to schedule audio delivery to the accessory and notifications from the Apple device whenever sample rate or format changes are required.
- Synchronizing shared resources such as the accessory's screen and speakers, plus determining which media source currently has audio and/or video focus on those resources.

HTTP/RTSP is used to set up, control, and monitor a CarPlay session. When an Apple device discovers a compatible accessory that is either already known by the Apple device or is user selected, the Apple device initiates a connection to the accessory and starts a CarPlay session. The following summarizes a typical session lifetime:

1. The Apple device establishes a TCP control connection to the accessory.
   - This connects to the TCP port advertised via Bonjour.
2. The Apple device authenticates the accessory and sets up a secure channel.
   - This process requires the accessory to perform the message exchange described in MFI-SAP (page 271).
   - After this step, all subsequent control requests and responses are encrypted.
3. The Apple device sends an info request to provide its info and get the accessory's info.
   - The request message is a plist providing the Apple device's model, version, etc.
   - The response message is a plist returning the accessory's audio formats, display info, model, current modes, etc.
   - See Info Message (page 273).
4. The Apple device sends a setup request to set up streams for control and, if needed, audio, screen, etc.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- This describes each stream to set up, including any TCP/UDP port numbers, configuration options, etc.
- The accessory responds with details about the streams it set up, such as listening TCP/UDP port numbers.
- For input or output TCP streams, the accessory will start listening for TCP connections.
- For input UDP streams, the Apple device will start listening for UDP packets.
- For output UDP streams, the accessory will start listening for UDP packets.
- See Setup Message (page 280).

5. The Apple device makes a TCP connection to the accessory for each TCP-based stream.
   - For both input and output streams, the Apple device initiates the connection.
   - For output streams, the Apple device sends data to the accessory (e.g., music to play on the accessory).
   - For input streams, the accessory will send data to the Apple device (e.g., HID reports). See Table 19-27 (page 282) and Table 19-29 (page 283).
6. The Apple device sends a record request to start a session on the accessory.
   - The accessory performs time sync negotiation with the Apple device. See Synchronization (page 286).
   - The accessory starts corresponding streams, creating threads to process audio/video data, timers, etc.
7. The Apple device sends setup requests to set up streams for audio and/or screen, and makes TCP connections as needed.
8. The Apple device and accessory exchange data.
   - The Apple device sends audio and video output data to the accessory for playback.
   - The accessory sends audio input data (if any) to the Apple device for input.
9. The Apple device sends a TEARDOWN requests to end one or more streams on the accessory.
   - The accessory stops processing for specified streams and releases corresponding resources.
10. The Apple device sends a TEARDOWN request to end the session on the accessory.
    - The accessory stops all audio and video processing and releases all session-specific resources.
11. The Apple device closes all TCP connections to the accessory.
    - This resets the Apple device back to the point before it started the session.

19.3.2.1 MFI-SAP

Authentication over the CarPlay interface requires use of the MFi Authentication Coprocessor for authenticating an encrypted channel. The protocol uses Curve25519 Elliptic-Curve Diffie-Hellman technology for key exchange, as described in cr.yp.to/ecdh.html. It also uses RSA for signing and verifying and AES-128 in counter mode for encryption.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Authentication starts when the Apple device generates a new, random Curve25519 key pair and sends its Curve25519 public key X to the accessory. The accessory responds by sending its public key Y to the Apple device, followed by its MFi Certificate and an encryption of Signature(Y,X).

Messages are exchanged using using RTSP/HTTP POST requests and responses to the `/auth-setup` URL. The content of the message is delivered via the body of the RTSP/HTTP message.

When the accessory receives the authentication request, it must:

1. Generate a new, random Curve25519 key pair.
2. Generate the shared secret using its Curve25519 private key and the streamer's Curve25519 public key.
3. Sign the two public keys with its RSA private key and encrypt the result with the AES master key derived from the Curve25519 shared secret.
4. Send its Curve25519 public key, its MFi certificate, and its AES-encrypted signature to the Apple device.

When the Apple device receives the response, it verifies that the accessory's MFi certificate is valid, decrypts the accessory response and verifies that the signature was signed by that certificate. If the signature verifies, the accessory is considered authenticated. The AES key derived from the Curve25519 shared secret is then used to encrypt all future content.

All Curve25519 keys must be generated from cryptographically strong random numbers. New Curve25519 keys must be generated for each authentication attempt; reuse of keys by the accessory is not permitted.

Each media stream must use a unique key and IV for encrypting content to avoid reusing the same key/IV for different pieces of data. The following are used to derive stream keys and IVs:

- ekey = Bytes 0-15 of SHA-512("AirPlayStreamKey" + streamConnectionID + session key (master))
- eiv = Bytes 0-15 of SHA-512("AirPlayStreamIV" + streamConnectionID + session key (master))

The steam connection ID is contained in Table 19-29 (page 283), and the ekey, eiv, and encryption type are contained in Table 19-23 (page 280). Table 19-14 (page 272) details the values used to specify the encryption type.

Table 19-14 Encryption type enum values

Value	Description
0x00	Invalid.
0x01	Unencrypted.
0x10	MFi-SAP-encrypted AES key.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

19.3.2.2 Info Message

The info message provides Apple device information to the accessory and returns accessory information to the Apple device. It is an HTTP GET to the `/info` URL (see Table 19-36 (page 285)) with binary plist request and response payloads. The Apple device may include any information it wants to provide to the accessory in the request plist. The request (see Table 19-15 (page 273)) may contain the "qualifier" key to limit the properties being requested from the accessory; for example, the Apple device may only want to get the accessory's model. The response (see Table 19-16 (page 273)) contains the properties requested by the Apple device.

Table 19-15 Info Message request keys

Key	Type	Required?	Description
qualifier	array	N	Array of property strings from Table 19-16 (page 273) to request from the accessory. If this key is missing, the accessory must return all of its properties. If this key is empty, the accessory must not return any properties.

Table 19-16 Info Message response keys

Key	Type	Required?	Description
audioFormats	array	Y	Array of dictionaries for audio formats supported by the accessory. See Table 19-17 (page 276).
audioLatencies	array	Y	Array of dictionaries specifying the fixed audio latencies within the accessory. See Table 19-19 (page 278). Accessory must specify default latencies for all audio streams, audio types and audio formats (some entries can be omitted based on the optional specifiers). A latter entry in the array overrides an earlier one if they both apply for a given stream type/audio type/format (last one wins).
bluetoothIDs	array	N (*)	Array of MAC address strings for the Bluetooth modules in the accessory. See disableBluetooth (page 318). (*) Required when accessory supports Bluetooth.

Confidential
Property of Universal Electronics Inc.
Unauthorized copying prohibited
Universal Electronics Inc.

Key	Type	Required?	Description
deviceID	string	Y	Globally unique device ID: e.g., "11:22:33:44:55". See Table 19-12 (page 269).
displays	array	Y	An array of dictionaries for display information supported by the accessory. See Table 19-20 (page 278).
features	bitfield	Y	Internally defined for Apple CarPlay. Value must not be modified. See Table 19-12 (page 269).
firmwareRevision	string	N	Firmware revision of the accessory, e.g. "FirmwareRevision0.1".
hardwareRevision	string	N	Hardware revision of the accessory, e.g. "HardwareRevision0.1".
hidDevices	array	Y	An array of dictionaries for HID device information. See User Input (page 295).
hidLanguages	array	N	List of BCP-47 language code strings for languages supported by the accessory's character recognizer.
keepAliveSendStatsAsBody	boolean	N	Indicates whether the accessory supports statistics as part of the keep alive message body.
limitedUIElements	array	N (*)	Array of UI elements affected by limited UI mode. [See Table 19-22 (page 279).] (*) Required when limitedUI.
limitedUI	boolean	N	Indicates whether or not certain UI elements are limited. See setLimitedUI (page 323).
macAddress	string	N	MAC address of the connected interface.
manufacturer	string	Y	Name of the accessory manufacturer (user-visible).

paywu@uniguard.com.tw Uniguard Payment Inc.

Key	Type	Required?	Description
model	string	Y	The model name of the accessory, which must be globally unique and must uniquely identify the product. This should be the model name of the vehicle for OEM integrations or an aftermarket unit's model name. See Table 19-12 (page 269).
modes	dictionary	Y	The current modes of the accessory. The modes dictionary uses the same keys as the "changeModes" command. See Table 19-53 (page 319).
name	string	Y	This must be set to "CarPlay".
nightMode	boolean	N	Indicates if it is dark outside and appearance should change to suit night time use. If the accessory does not support night mode, then it should not respond to this property at all (return an unsupported error). The Apple device can then use other metrics (such as time of day) to decide on its own when to enter night mode. See setNightMode (page 323).
oemIcon	data	N (*)	PNG data for 104 x 104 pixel icon representing the accessory manufacturer's logo. The icon should be solid white with a transparent, alpha channel background. There should be no shading or attempts to represent 3-dimensional volume since iOS will present the icon in the appropriate style (applying backgrounds, gradients, etc). The icon should follow the design principles of the current version of iOS. For example, see the Mail envelope icon. (*) Required when oemIconVisible is True.
oemIconLabel	string	N (*)	Label shown underneath the oemIcon. (*) Required when oemIconVisible is True.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Key	Type	Required?	Description
oemIconVisible	boolean	N	Whether or not the oemIcon is visible on the home screen.
OSInfo	string	N	Operating system information including name, version, and architecture (e.g. "Darwin 13.0.0 x86_64").
protocolVersion	string	Y	Protocol version string major.minor ; e.g., "1.0". The default value is "1.0". See Table 19-12 (page 269).
rightHandDrive	boolean	N	True if the vehicle is right hand drive.
settingsIcon	data	N	PNG data for 44 x 30 pixel icon representing the accessory manufacturer's logo (user-visible).
sourceVersion	string	Y	CarPlay source version assigned by Apple, in the form x.y.z . See Table 19-12 (page 269).
statusFlags	bitfield	Y	Status of an operation. Value must not be modified. See Table 19-12 (page 269).

Table 19-17 Audio format keys

Key	Type	Required?	Description
type	enum	Y	The stream type to which the input and/or output formats apply. One of Table 19-31 (page 283).
audioInputFormats	bitfield	Y (*)	The audio input formats supported by the given type. See Audio (page 259) for requirements. (*) Not required if the stream type does not support input. Combination of Table 19-18 (page 277).
audioOutputFormats	bitfield	Y	The audio output formats supported by the given type. See Audio (page 259) for requirements. Combination of Table 19-18 (page 277).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

The audio input formats supported by the accessory must be a subset of the audio output formats supported. When using full-duplex audio, the Apple device will use the same format for both directions and will choose from the supported input formats.

Table 19-18 Audio formats bitfield enum values

Value	Bit	Description
0x00000004	2	PCM, 8000 Hz, 16-Bit, Mono
0x00000008	3	PCM, 8000 Hz, 16-Bit, Stereo
0x00000010	4	PCM, 16000 Hz, 16-Bit, Mono
0x00000020	5	PCM, 16000 Hz, 16-Bit, Stereo
0x00000040	6	PCM, 24000 Hz, 16-Bit, Mono
0x00000080	7	PCM, 24000 Hz, 16-Bit, Stereo
0x00000100	8	PCM, 32000 Hz, 16-Bit, Mono
0x00000200	9	PCM, 32000 Hz, 16-Bit, Stereo
0x00000400	10	PCM, 44100 Hz, 16-Bit, Mono
0x00000800	11	PCM, 44100 Hz, 16-Bit, Stereo
0x00001000	12	PCM, 44100 Hz, 24-Bit, Mono
0x00002000	13	PCM, 44100 Hz, 24-Bit, Stereo
0x00004000	14	PCM, 48000 Hz, 16-Bit, Mono
0x00008000	15	PCM, 48000 Hz, 16-Bit, Stereo
0x00010000	16	PCM, 48000 Hz, 24-Bit, Mono
0x00020000	17	PCM, 48000 Hz, 24-Bit, Stereo
0x00040000	18	Reserved
0x00080000	19	Reserved
0x00100000	20	Reserved
0x00200000	21	Reserved
0x00400000	22	Reserved

Larry Wu@Quintabio.com.tw Quintabio LTD Larry Wu

Value	Bit	Description
0x00800000	23	Reserved
0x01000000	24	Reserved
0x02000000	25	Reserved

Table 19-19 Audio latencies keys

Key	Type	Required?	Description
type	enum	N	The stream type to which the input and/or output formats apply. One of Table 19-31 (page 283).
audioType	string	N	Type of audio content (e.g. telephony, media, etc.). See Table 19-32 (page 284).
sr	number	N	Number of samples per second (e.g. 44100).
ss	number	N	Bit size of each audio sample (e.g. "16").
ch	number	N	Number of audio channels (e.g. 2 for stereo).
inputLatencyMicros	number	Y	Input latency in microseconds.
outputLatencyMicros	number	Y	Output latency in microseconds.

Table 19-20 Display keys

Key	Type	Required?	Description
edid	data	N	Raw EDID of the display, if available.
features	bitfield	Y	Features of the display as a bitmask. Combination of Table 19-21 (page 279).
maxFPS	number	N	Max frames per second the display supports.
heightPixels	number	Y	Height of the display in pixels. Must be non-zero.
widthPixels	number	Y	Width of the display in pixels. Must be non-zero.
heightPhysical	number	Y	Height of the display in millimeters. Must be non-zero.
widthPhysical	number	Y	Width of the display in millimeters. Must be non-zero.

Uploaded by Tapwing
Tapwing@uniquedevices.com

Key	Type	Required?	Description
uuid	string	Y	UUID of the display.

Table 19-21 Display features bitfield enum values

Value	Bit	Description
0x02	1	Supports interacting via knobs.
0x04	2	Supports interacting via low-fidelity touch.
0x08	3	Supports interacting via high-fidelity touch.

Note that the choice of display features supported by the accessory will directly impact the look and feel of the user interface presented by the Apple device on the accessory's screen.

- An accessory that reports only supporting knobs (bit 1) will be presented with a user interface that may be interacted with exclusively via the use of a knob, select, and back buttons.
- An accessory that reports only supporting low-fidelity touch (bit 2) will be presented with a limited touch interface. For instance, the user interface for a map will be augmented with soft arrow-keys as overlays to enable panning.
- An accessory that reports only supporting high-fidelity touch (bit 3) will be presented with a touch interface that allows for scrolling and panning gestures.

The display features value is a bitmask, so combinations are possible -- for instance: "supports interacting via knobs" and "supports interacting via high-fidelity touch." In this case, the high-fidelity touch user interface dominates, with knob use being primarily constrained to scrolling through lists.

To claim support for high-fidelity touch, the time between a user touch input to the time a frame updates on the display must be less than 140ms.

Table 19-22 Limited UI elements string values

Value	Description
"softKeyboard"	Touch keyboard that appears on screen.
"softPhoneKeypad"	Touch phone keypad that appears on screen.
"nonMusicLists"	Lists of non-music items.
"musicLists"	Lists of music items.

Apple Confidential
Licensed to [redacted]@gmail.com

The `/info` URL may also be used via an HTTP GET to read accessory information. The URL may contain a query string to limit the properties being requested (for example, `GET /info?deviceID&model` to get the accessory's device ID and model properties).

19.3.2.3 Setup Message

The setup message is sent by the device to configure and set up all streams between the accessory and the device. The CarPlay streams are logical flows of data, such as a control stream for sending commands and getting responses, a screen stream for sending H.264 frames, an audio stream for sending audio, etc. The setup request message includes descriptions of all the streams the controller wants to set up. The response includes details about the streams that were set up, i.e. control only, main audio and screen streams, alternate audio only, etc. Both the request and the response payloads are binary plists.

The initial setup message after the device connects to the accessory contains information to set up the control stream. If needed, it may also setup the audio and screen streams.

Table 19-23 (page 280) lists the specifics of the initial setup request keys sent by the Apple device to the accessory.

Table 19-23 Initial Setup request keys

Key	Type	Required?	Description
deviceID	string	Y	Globally unique device ID (e.g. "11:22:33:44:55").
eiv	data	Y	AIS Encryption Initialization Vector for the accessory. This is sent in the clear. See MFI-SAP (page 271)
ekey	data	Y	AIS Encryption Key for the accessory. Encrypted with master key from encryption setup. See MFI-SAP (page 271)
et	enum	Y	Encryption type. One of Table 19-14 (page 272).
macAddress	string	Y	MAC address of the iOS network interface used for the connection.
model	string	Y	Model name of the Apple device (e.g. "Device1,1").
name	string	Y	User-customizable name of the Apple device.
osBuildVersion	string	Y	Operating system build version string (e.g. "11A200").
sessionUUID	string	Y	UUID of the session.

Confidential and Proprietary
Apple Inc.

Key	Type	Required?	Description
sourceVersion	string	Y	Apple device side CarPlay source version string (e.g. "101.7").
streams	array	Y (*)	Array of stream descriptions. See Table 19-27 (page 282) and Table 19-29 (page 283). (*) The initial setup message may not include "streams" if it only sets up the control stream.
timingPort	number	Y	Port the Apple device is listening on for time sync requests.

Table 19-24 (page 281) lists the specifics of the initial setup response keys sent by the accessory to the Apple device.

Table 19-24 Initial Setup response keys

Key	Type	Required?	Description
eventPort	number	Y	TCP/UDP port number for events.
streams	array	Y (*)	Array of stream descriptions. See Table 19-27 (page 282) and Table 19-29 (page 283). (*) The initial setup message may not include "streams" if it only sets up the control stream.
timingPort	number	Y	Port the controller is listening on for time sync requests.

Once a session is established, the setup message is sent to set up and configure the audio and screen streams, as they are required by the device.

Table 19-25 (page 281) lists the specifics of the setup request keys sent by the Apple device to the accessory after a session has been started.

Table 19-25 Setup request keys

Key	Type	Required?	Description
streams	array	Y	Array of stream descriptions. See Table 19-27 (page 282) and Table 19-29 (page 283).

Unigarden Parkway LTD
parkway@unigarden.com

Table 19-26 (page 282) lists the specifics of the setup response keys sent by the accessory to the Apple device after a session has been started.

Table 19-26 Setup response keys

Key	Type	Required?	Description
streams	array	Y	Array of stream descriptions. See Table 19-27 (page 282) and Table 19-29 (page 283).

19.3.2.3.1 Streams

Each stream is set up using a stream descriptor with a type, parameters, and behaviors associated with it, as listed in Table 19-27 (page 282) for main/alternate audio stream and Table 19-29 (page 283) for screen stream.

The wallTime and mediaTime response parameters are used for media clock synchronization of the main and alternate audio streams as described in Synchronization (page 286).

Table 19-27 Main/alternate audio stream descriptors request keys

Key	Type	Required?	Description
audioFormat	bitfield	Y	Format of the audio data (e.g. 44100 Hz, Stereo PCM). For main audio applies to both input and output. Combination of Table 19-18 (page 277).
audioLatencyMs	number	Y	Desired milliseconds of audio latency.
audioLoopback	boolean	N	True if audio output must be looped back and sent as audio input. Debug use only.
audioType	string	Y (*)	Type of audio content (e.g. telephony, media, etc.). See Table 19-32 (page 284). (*) Main audio only.
dataPort	number	Y (*)	UDP port the controller is listening on for data. (*) Only required when audio input is enabled (main audio only).
input	boolean	N	True if input must be enabled for the stream (main audio only).
type	enum	Y	Type of stream. One of Table 19-31 (page 283).

Do Not Redistribute
Confidential
Apple Inc.

Table 19-28 Main/alternate audio stream descriptors response keys

Key	Type	Required?	Description
dataPort	number	Y	UDP port the accessory is listening on for data.
type	enum	Y	Type of stream. One of Table 19-31 (page 283).
mediaTime	number	N	Media timestamp at the same moment as "wallTime".
wallTime	number	N	Wall clock timestamp at the same moment as "mediaTime" using synchronized wall time.

Table 19-29 Screen stream descriptors request keys

Key	Type	Required?	Description
latencyMs	number	Y	Desired milliseconds of screen latency.
type	enum	Y	Type of stream. One of Table 19-31 (page 283).
streamConnectionID	number	Y	Unique stream connection ID. Used for deriving encryption key and IV. See MFI-SAP (page 271)

Table 19-30 Screen stream descriptors response keys

Key	Type	Required?	Description
dataPort	number	Y	TCP or UDP port the accessory is listening on for data.
type	enum	Y	Type of stream. One of Table 19-31 (page 283).

Table 19-31 Stream ID enum values

Name	Value	Protocol	Description
Invalid	0	n/a	Reserved for an invalid stream ID.
Main Audio	100	UDP	Low-latency audio input/output. Value is also the RTP payload type.
Alt Audio	101	UDP	Low-latency UI sounds, alerts, etc., output. Value is also the RTP payload type.
Screen	110	TCP/UDP	H.264 screen output. Value is also the RTP payload type.

Paid version @ unlockiphone.com

Table 19-32 Main Audio type string values

Value	Description
"default"	Unspecified or unknown audio type
"alert"	Ringtones, alarms, and other high-priority sounds
"media"	Entertainment (e.g.: music playback)
"telephony"	Telephony
"speechRecognition"	Speech recognition

Table 19-33 Channels

Name	Transport	Port	QoS	Direction	Description
RTSP Controller control	TCP	5000	BE	device to accessory	iOS control channel
RTSP Accessory control	TCP	dynamic	BE	accessory to device	Accessory control channel
RTP Main Audio Output	UDP	dynamic	VO	device to accessory	Audio data
RTP Alternate Audio Output	UDP	dynamic	VO	device to accessory	Audio data
RTP Main Audio Input	UDP	dynamic	VO	accessory to device	Audio data
Time Sync Client	UDP	dynamic	CTL	accessory to device	Audio time sync requests
Time Sync Server	UDP	dynamic	CTL	device to accessory	Audio time sync responses
RTP Screen	TCP	dynamic	VI	device to accessory	User interface (H.264 frames)

taywu@uniguardian.com.tw Taywu Uniguardian LTD

19.3.2.4 Feedback Message

The feedback message exchanges timing information and statistics to and from an accessory. The controller does an HTTP POST to the `/feedback` URL and the accessory responds with its feedback info. The request and response payloads are binary plists. These plists may be omitted or empty if they don't have feedback to report in that direction.

This message is used for media clock synchronization of the main and alternate audio streams as described in Synchronization (page 286).

Table 19-34 Feedback response keys

Key	Type	Required?	Description
streams	array	N	Streams reporting feedback. Contains dictionaries of Table 19-35 (page 285).

Table 19-35 Feedback stream keys

Key	Type	Required?	Description
sr	number	N (*)	Estimated consumption rate in samples per second of the stream. This is calculated based on mediaTime and wallTime. (*) Required when type is an audio stream.
type	enum	Y	Type of stream. One of Table 19-31 (page 283).
mediaTime	number	N	Media timestamp at the same moment as "wallTime".
wallTime	number	N	Wall clock timestamp at the same moment as "mediaTime" using synchronized wall time.

19.3.2.5 URLs

Table 19-36 (page 285) lists the URLs used with CarPlay. All URLs are accessed via HTTP/RTSP.

Table 19-36 URLs for CarPlay

URL	Method	Description
n/a	OPTIONS	Standard HTTP options message to return the supported methods, etc.
n/a	RECORD	RTSP message to start playback.

Apple Confidential
Unauthorized Distribution
jawad@gurbanigranar.com

URL	Method	Description
n/a	SETUP	RTSP message to set up a session and streams.
n/a	TEARDOWN	RTSP message to tear down a session.
`/auth-setup`	POST	Performs MFi-SAP authentication; see MFI-SAP (page 271). Request and response are opaque security data.
`/command`	POST	For performing command requests and their responses; see Commands (page 316). Requests and responses are binary plists.
`/feedback`	POST	Exchanges information with the accessory; see Feedback Message (page 285). Requests and responses are binary plists.
`/info`	GET	Exchanges information with the accessory; see Info Message (page 273). Requests and responses are binary plists.
`/perf`	POST	Performs network performance tests. Requests and responses are binary plists.

19.3.2.6 Synchronization

The clocks between the Apple device and the accessory must be synchronized to present media data at the correct time. The synchronization process is managed by the Communication Plug-in in the CarPlay Client and includes:

- Synchronizing wall clocks so both entities have an accurate notion of absolute presentation times (for example, when the user should hear an audio sample). Note that the synchronized clocks are maintained internally and do not affect the system time of either the Apple device or the accessory.
- Mapping media clocks to wall clocks so that a media timestamp from one device can be mapped to a media timestamp on another device. This allows the media production or consumption rate of one device's media to be clocked off the other device.

The accessory must drive all audio streams, regardless of type (input or output) from the same media clock that is used for synchronization with the Apple device.

19.3.2.6.1 Wall Clock Synchronization

Synchronizing wall clocks is done by exchanging Network Time Protocol (NTP) packets. The Apple device acts as the NTP server, providing the reference clock, and the accessory acts as the NTP client, syncing its clock to the reference. There are two parts to this process: rate and offset synchronization. Rate synchronization tracks how fast the reference clock is running relative to the local clock. This is done by measuring the difference

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

between timestamps over an increasingly large interval to minimize measurement noise (such as network jitter). The interval is capped to allow it to adapt to real changes in the reference clock rate (for example, from temperature changes affecting the rate). Offset synchronization tracks the absolute difference from the reference clock. This is measured with each packet exchange and smoothed to minimize measurement noise.

19.3.2.6.2 Media Clock Synchronization

Synchronizing media clocks is done by periodically exchanging mapping relationships between media sample time and wall clock time. When the accessory is providing the clock, it must sample its media clock and synchronized wall clock at the same moment. These samples are used to calculate media clock rate. The synchronized wall clock is used so the calculated rate is based on time relative to the Apple device. This allows the Apple device to adapt its media production rate to match the accessory's consumption rate and avoid the need for sample rate conversion. The media clock and wall clock tuple also allows the Apple device to determine the absolute media clock offset between devices. Media clock timing relationships are exchanged via the Feedback Message (page 285).

The parameters exchanged in the Feedback message are calculated by the Communication Plug-in on behalf of the accessory. To do so, the accessory must provide local and media clocks using a high resolution tick counter to measure time intervals (such as a processor cycle counter). These counters must be stable (less than 50 PPM of variance) and precise (10 MHz or higher) with minimal and consistent overhead (less than 1 microsecond to sample each clock).

19.3.3 Resource Management

19.3.3.1 Resources

In its typical embodiment in an automobile, the CarPlay architecture includes three resources that must be managed:

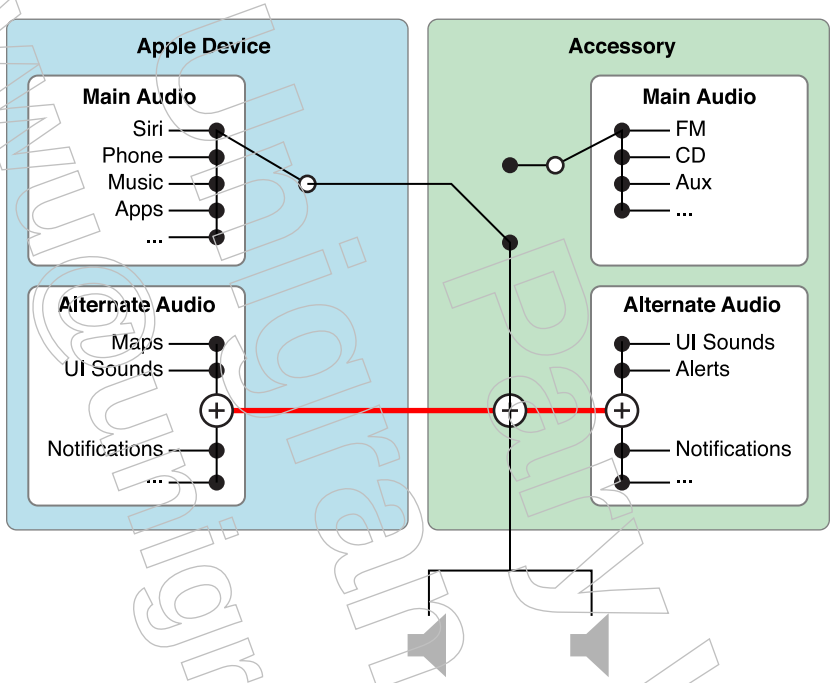
- **Screen**: The automotive head unit screen.
- **Main Audio**: The full-duplex (input and output) audio stream. There can be only one source (either accessory or Apple device) for main audio. Voice control, Siri, telephony, and media are examples of audio that uses this stream.
- **Alternate Audio**: An output-only audio stream. There can be multiple sources of alternate audio, which must be mixed together (possibly with Ducking (page 268)). UI sounds, alerts, and navigation prompts are examples of audio that uses this stream.

Screen and main audio are managed as a shared resource and both the accessory and the device must request an ownership (see Resource Ownership (page 289)), where as the alternate audio stream must be available to the device at all times.

Licensed by Wu Jianfeng (jianfeng.wu@tcl.com)

Both accessory and Apple device must distinguish between main and alternate audio, and route audio accordingly. Figure 19-6 (page 288) illustrates conceptually the various audio sources and shows how they must be mixed.

Figure 19-6 Mixing CarPlay audio



19.3.3.2 App States

Three app-specific states must be communicated between the accessory and the Apple device:

- **Speech**: An entity (accessory or Apple device) is performing a speech operation (such as, speaking or recognizing speech). All audio output from the accessory must be disabled while speech is being recorded to prevent interference with recognition algorithms. When an entity is speaking to the user, other entities must avoid playing speech. For example, during this time navigation prompts can be replaced with tones or other indicators.
- **Phone Call**: An entity is on a phone call. Only a single entity must be in this state at a time.
- **Turn-by-turn Navigation**: An entity is performing turn-by-turn navigation. Only a single entity must be in this state at a time.

Note that additional app states may be added in future versions of CarPlay. Also note that the values of the app state properties are purely informative, to allow both the accessory and Apple device to make policy decisions that improve the user experience. For example:

Early Draft Confidential Property
Apple Inc.

- While one entity is performing speech recognition, all audio from the accessory must be silenced to avoid interfering with the speech recognition algorithms.
- While one entity is speaking to the user, other speech must be suppressed or conveyed as tones rather than speech; multiple, simultaneous speakers can be confusing to the user.
- While one entity is engaged in a phone call, the other entity cannot initiate a phone call (first caller wins).
- If one entity starts performing turn-by-turn navigation, the other entity must stop doing its turn-by-turn navigation, if applicable (last system wins).

The accessory must respect the current values of the app state properties. These properties must not be ignored unless it is essential to give critical safety or emergency information to the user.

19.3.3.3 Resource Ownership

Only one entity (accessory or Apple device) can claim ownership of a resource at any given time. When one entity claims ownership of a resource, it also constrains the circumstances under which ownership can be transferred. The three constraint modifiers are:

- **Anytime**: The other entity may take/borrow the resource at any time. If the screen is showing unimportant information, it would use this value.
- **User-Initiated**: The other entity may take or borrow this resource, but doing so would disrupt the current user experience, and must only happen in response to a user-initiated action, and only if acquiring the resource is crucial to the user experience of the new mode. While music is playing, for example, the main audio channel must not be taken or borrowed unless the user has switched to a different audio source. A user-initiated action is defined as one where the user taps a user interface element, presses a button, turns a knob, and so forth, in order to instruct an entity to set a new context, change source, and so forth. The accessory must not initiate any 'user-initiated' actions on the user's behalf unless these are required to give critical safety or emergency information to the user.
- **Never**: The other entity may not take or borrow this resource under any circumstances. The owner/borrower of the resource would set this during a phone call or while a safety alert is being shown.

Ownership of a resource can be permanently transferred (taken) or temporarily transferred (borrowed).

Take: Ownership of the resource is permanently transferred to the other entity. If the user switches from the accessory's FM radio to the Apple device's Music app, for example, the Apple device would take the main audio channel from the accessory.

Borrow: Ownership is temporarily transferred. When ownership is returned, the owner should resume whatever it was doing with the resource. If the user is listening to the accessory's FM radio and starts a Siri session, for example, the Apple device would borrow the main audio channel. When the Siri session ends, ownership returns to the accessory and the FM radio must continue.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

When a resource is borrowed, the other entity still retains ownership and must maintain its state. Maintaining the original ownership state (with take constraints) is important to support the following three possible ways for a borrow state to end:

- The entity borrowing the resource has finished using it and returns it to the owner. This is the typical case. For example, when a phone call ends the main audio channel may return to the accessory, which continues playing FM radio.
- The owner requires the resource and prematurely ends the borrowing state, using the take constraint of the borrow state. For example, the Apple device may borrow the main audio channel to start a Siri session, but the accessory prematurely ends the session because the main audio channel is needed for an incoming call from a phone in the vehicle that does not support CarPlay.
- The entity borrowing the resource wishes to convert from borrowing to ownership, using the original owner's take constraints. For example, the user may ask Siri to switch from FM radio to the Apple device's music player. The Siri session initially borrows the main audio channel, but when the music player starts it will want to own it.

19.3.4 Modes

Modes provide a mechanism to manage shared resources (e.g. the main screen), and state (e.g. on a phone call) during a CarPlay session. They are represented by the values in the following tables:

Table 19-37 Entity enum values

Name	Value	Description
None	0	Only used for appStates to indicate that no entity is in one of the App States. May not be used to describe ownership for a given resource (Main Screen or Main Audio).
Controller	1	The Apple device owns the given resource (Main Screen or Main Audio) or is in the given app state.
Accessory	2	The accessory owns the given resource (Main Screen or Main Audio) or is in the given app state.

Table 19-38 Resource ID enum values

Name	Value	Description
MainScreen	1	The main display for the system.

Confidential Property of Uniview Technology Co., Ltd

Name	Value	Description
MainAudio	2	The main audio stream used for input, output, or full-duplex audio, used by telephony and media. This mode requires exclusive access; either the Apple device or the accessory may use it but not both at once.

Table 19-39 Resource constraint enum values

Name	Value	Description
Anytime	100	Resource may be taken or borrowed at any time.
User Initiated	500	Resource may be taken or borrowed if user requests.
Never	1000	Resource may never be taken or borrowed.

Table 19-40 Resource ownership transfer type enum values

Name	Value	Description
Take	1	Acquire ownership of a resource permanently. For example, if the user switches from the accessory's FM radio to the Apple device's music app, the Apple device would "take" main audio.
Untake	2	Indicate that a resource is no longer needed. Ownership of a resource does not change immediately, it changes the next time the controller requests it. The accessory is notified using a "modesChanged" command (see modesChanged (page 320)).
Borrow	3	Transfer ownership temporarily. For example, if the user is listening to the accessory's FM radio and then starts a Siri session, the Apple device would "borrow" main audio. When the Siri session ends, the Apple device would "unborrow" main audio.
Unborrow	4	Release ownership of a resource that was acquired temporarily.

Table 19-41 Resource transfer priority enum values

Name	Value	Description
Nice-to-have	100	Transfer succeeds only if constraint is Anytime.
User Initiated	500	Transfer succeeds only if constraint is UserInitiated or Anytime.

Confidential
Property of Apple Inc.
Do not distribute
Licensed to ...@...com.tw

Table 19-42 App state enum values

Name	Value	Description
Speech	1	An entity is performing a speech operation. All audio output must be disabled while speech is being recorded to prevent interference with recognition algorithms. When an entity is speaking to the user, other entities must avoid playing speech. During this time, navigation prompts, etc. can be replaced with tones or other indicators.
PhoneCall	2	An entity is engaged in a phone call. Only a single entity must be in this state at a time.
TurnByTurn	3	An entity is performing turn-by-turn navigation. Only a single entity must be in this state at a time.

Table 19-43 Speech mode enum values

Name	Value	Description
None	-1	No speech-related states are active.
Speaking	1	An entity is speaking to the user.
Recognizing Speech	2	An entity is recording audio to recognize speech from the user.

19.3.4.1 Initial Mode

When a session starts (on connection, for example), the Apple device queries the accessory for its current mode to synchronize the states of the two devices. This is done via the Info Message (page 273). The accessory must respond with its current mode, including the type, priority and constraints for each resource, as well as current app states and speech mode. The Apple device interprets this initial state as a resource transfer to the accessory and will only then evaluate whether any mode changes are required and permissible. For example, if music was already playing on the Apple device prior to connection to the accessory, the Apple device would then "take" main audio, provided that this action was compatible with the accessory's initial mode constraints.

Synchronization of the Initial Mode may be required beyond just the start of a CarPlay session: the accessory must be ready to respond to an Info message requesting an update of its current mode at any time.

19.3.4.2 Mode Changes

To modify the current mode, the accessory sends the Apple device a mode change request communicating the intent of the mode change. This is done via the `/command` URL (See Commands (page 316)). The Apple device takes the request and determine a new mode based on it. The Apple device will reject badly-formed

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

requests or requests that are not compatible with the current resource constraints; in either case an error will be returned to the accessory. Once the Apple device has evaluated a new mode, it communicates it to the accessory (see Current Mode (page 294)). Mode changes originating from the Apple device are communicated directly to the accessory as an update to the Current Mode; they are not preceded by a mode change request from the Apple device to the accessory.

The accessory must honor any updates to the Current Mode within 100 ms of receiving them from the Apple device. For example, it must relinquish control of the screen within 100 ms if screen ownership is transferred to the Apple device.

A mode change request consists of the following information:

Main Screen:

- Transfer Type: Take, Untake, Borrow, or Unborrow
- Priority: Nice to Have or User-Initiated
- Constraints: (1 or 2 constraints, depending on the Transfer Type)
- For Transfer Type Take, Apple device may Take back Main Screen: Anytime or User Initiated or Never
- For Transfer Type Take, Apple device may Borrow back Main Screen: Anytime or User Initiated or Never
- For Transfer Type Borrow, Apple device may Unborrow Main Screen: Anytime or User Initiated or Never

Main Audio:

- Transfer Type: Take, Untake, Borrow, or Unborrow
- Priority: Nice to Have or User-Initiated
- Constraints: (1 or 2 constraints, depending on the Transfer Type)
- For Transfer Type Take, Apple device may Take back Main Audio: Anytime or User Initiated or Never
- For Transfer Type Take, Apple device may Borrow back Main Audio: Anytime or User Initiated or Never
- For Transfer Type Borrow, Apple device may Unborrow Main Audio: Anytime or User Initiated or Never

App States:

- Speech: Speaking, Speech Recognizing, or Neither
- Phone Call: True or False
- Turn-by-turn navigation: True or False

A change request must include at least one of the foregoing properties. If an action requires changes to multiple properties of a mode, they should be sent in a single transaction.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Not all mode change requests will require changes to all properties, in which case unchanged properties can be left out.

It is important that the accessory always update the Apple device with its current state, regardless of the current mode. For example, if the user switches from FM radio to the CD player, ownership of the main audio will not change. But the accessory should still send a mode change request indicating that it wants to take ownership of main audio.

The accessory must specify whether it wishes to own or borrow a resource, as described in the Resource Ownership (page 289). When the accessory has finished borrowing a resource, it must send a mode change request to release it. The release transfer type indicates to the Apple device that the accessory no longer needs the resource, and the resource constraints will be reset to "Other may take anytime". A successful taking or borrowing of a resource currently borrowed by the other entity will terminate the borrow, as described in Resource Ownership (page 289).

The accessory can use a priority value to indicate to the Apple device how badly it wants a resource. A priority of "nice to have" will succeed if the take/borrow constraint is "anytime." A priority of "user-initiated" will succeed if the take/borrow constraint is "user-initiated" or "anytime."

In the case of a non-routine safety or emergency situation (for example, a flat tire), the accessory should still send a changeModes request, but it does not have to wait for a modesChanged notification before taking over video or audio. In all other routine cases, the accessory must wait for the notification.

The accessory must be ready to handle setup or record events from the Apple device within 200 ms of the time that the accessory issues the changeModes command (see changeModes (page 318)) or receives a modesChanged command (see modesChanged (page 320)).

19.3.4.3 Current Mode

The current mode that the Apple device communicates to the accessory includes these parameters:

- **Accessory** or **Apple device** has Screen
- **Accessory** or **Apple device** has Main Audio
- **Accessory** or **Apple device** or **Neither** is performing speech recognition or speech
- **Accessory** or **Apple device** or **Neither** is engaged in a phone call
- **Accessory** or **Apple device** or **Neither** is performing turn-by-turn navigation

Note that if ownership of a resource is returned to the accessory when the accessory didn't explicitly ask for it, the accessory should resume whatever it was last doing with that resource, if possible.

For example modes and mode changes, see Examples of Modes and Mode Changes (page 329).

Confidential Property of Apple Inc. Unauthorized Distribution Prohibited

19.3.5 User Input

This section covers the delivery of user input events for CarPlay and assumes familiarity with the Device Class for Human Interface Devices (HID) specification, the HID Usage Tables and addendum that are published by the USB Implementors Forum. For further information please visit http://www.usb.org/developers/hidpage/.

See additional requirements in HID Requirements (page 407).

Each HID device dictionary contains a HID descriptor (see USB HID specification), device UUID, and a display UUID directing the HID events to a specific display. Table 19-44 (page 295) lists the possible HID device description keys stored in an HID device dictionary.

Table 19-44 HID device description keys

Key	Type	Required?	Description
displayUUID	string	Y	UUID of a display associated with the HID device or a null UUID if not associated to a display.
hidCountryCode	number	Y	USB-style HID country code.
hidDescriptor	data	Y	USB-formatted HID descriptor.
hidProductID	number	Y	USB-style HID product ID. Must be non-zero.
hidVendorID	number	Y	USB-style HID vendor ID. Must be non-zero.
name	string	Y	User-friendly name of the HID device.
uuid	string	Y	UUID to uniquely identify the HID device.

19.3.5.1 Digitizer Support (Touchscreen and Touchpad)

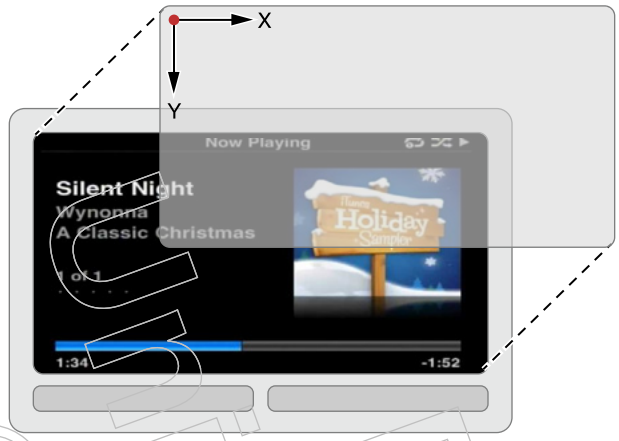
A digitizer is a device that measures absolute spatial position, typically in two or more dimensions. CarPlay supports the following types of digitizers:

19.3.5.1.1 Touchscreen

A touchscreen is a digitizer with an integrated display that allows the use of a finger for direct interaction with a presented interface.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 19-7 Touchscreen

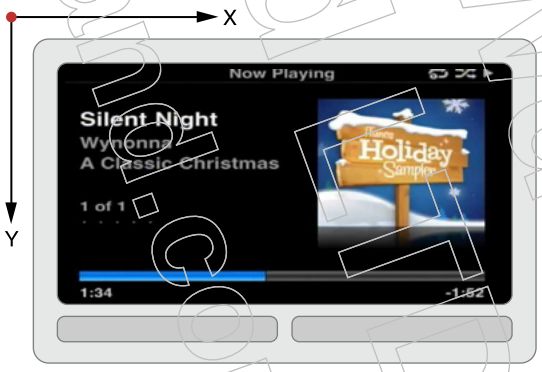


The touch surface must be sampled at the same rate as the refresh rate of the video stream on the integrated display. A sampling and refresh rate of 60 Hz is recommended. See High Resolution Display (page 247) and User Interface (UI) Stream (page 258).

19.3.5.1.2 Touchpad

A touchpad is a digitizer without an integrated display that allows the use of a finger for indirect interaction with a presented interface.

Figure 19-8 Touchpad



Generally, reports originating from this class of device would entailed relative movements synonymous with a mouse. These devices must only convey absolute transducer movement.

19.3.5.1.3 HID Descriptor Support

The accessory may support the following HID usages:

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 19-45 Digitizer Support HID Usages

Page ID	Page Name	Usage ID	Usage Name	Usage Type
0x01	Generic	0x30	X	Dynamic Value
0x01	Generic	0x31	Y	Dynamic Value
0x0D	Digitizer	0x04	Touch Screen	Application Collection
0x0D	Digitizer	0x05	Touch Pad	Application Collection
0x0D	Digitizer	0x22	Finger	Logical Collection
0x0D	Digitizer	0x32	In Range	Momentary Control
0x0D	Digitizer	0x33	Touch	Momentary Control
0x0D	Digitizer	0x37	Data Valid	Momentary Control
0x0D	Digitizer	0x38	Transducer Index	Dynamic Value
0x0D	Digitizer	0x42	Tip Switch	Momentary Control
0x0D	Digitizer	0x51	Contact Identifier	Dynamic Value

When the Apple device does not own the screen, the accessory must not send these HID usages.

19.3.5.1.4 Single Touch

Single touch implies the ability to convey the movement of only one finger over a digitizer surface.

In order to ensure proper detection, an accessory must declare an application collection using either one of the following usages:

- Touch Screen
- Touch Pad

The accessory must declare a logical collection using the following usage:

- Finger

Each logical collection must also contain absolute X and Y axes:

- X
- Y

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

And either one of the following to convey a touch down:

- Touch
- Tip Switch

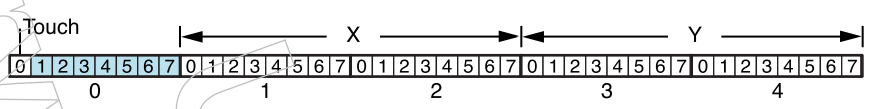
The following is an example report descriptor and format for a single-touch screen:

```
0x05, 0x0D,         // Usage Page (Digitizer)
0x09, 0x04,         // Usage (Touch Screen)
0xA1, 0x01,         // Collection (Application)
0x05, 0x0D,         //   Usage Page (Digitizer)
0x09, 0x22,         //   Usage (Finger)
0xA1, 0x02,         //   Collection (Logical)
0x05, 0x0D,         //     Usage Page (Digitizer)
0x09, 0x33,         //     Usage (Touch)
0x15, 0x00,         //     Logical Minimum......... (0)
0x25, 0x01,         //     Logical Maximum......... (1)
0x75, 0x01,         //     Report Size............. (1)
0x95, 0x01,         //     Report Count............ (1)
0x81, 0x02,         //     Input...................(Data, Variable, Absolute)
0x75, 0x07,         //     Report Size............. (7)
0x95, 0x01,         //     Report Count............ (1)
0x81, 0x01,         //     Input...................(Constant)
0x05, 0x01,         //     Usage Page (Generic Desktop)
0x09, 0x30,         //     Usage (X)
0x15, 0x00,         //     Logical Minimum......... (0)
0x26, 0x20, 0x03,   //     Logical Maximum......... (800)
0x75, 0x10,         //     Report Size............. (16)
0x95, 0x01,         //     Report Count............ (1)
0x81, 0x02,         //     Input...................(Data, Variable, Absolute)
0x09, 0x31,         //     Usage (Y)
0x15, 0x00,         //     Logical Minimum......... (0)
0x26, 0xE0, 0x01,   //     Logical Maximum......... (480)
0x75, 0x10,         //     Report Size............. (16)
0x95, 0x01,         //     Report Count............ (1)
0x81, 0x02,         //     Input...................(Data, Variable, Absolute)
```

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

```
0xC0,                   //   End Collection
0xC0,                   // End Collection
```

Figure 19-9 Example Input Report Layout for Single-Touch Screen



19.3.5.1.5 Multi Touch

Multi touch implies the ability to convey the movement of more than one finger over a digitizer surface.

In order to ensure proper detection, an accessory must follow the requirements of single touch. Additionally, an accessory must be able to identify the finger producing the movement using either one of the following usages:

- Transducer Index
- Contact Identifier

The following is an example report descriptor and format for a multi-touch screen:

```
0x05, 0x0D,             // Usage Page (Digitizer)
0x09, 0x04,             // Usage (Touch Screen)
0xA1, 0x01,             // Collection (Application)
0x05, 0x0D,             //   Usage Page (Digitizer)
0x09, 0x22,             //   Usage (Finger)
0xA1, 0x02,             //   Collection (Logical)
0x05, 0x0D,             //     Usage Page (Digitizer)
0x09, 0x38,             //     Usage (Transducer Index)
0x75, 0x08,             //     Report Size..... (8)
0x95, 0x01,             //     Report Count..... (1)
0x81, 0x02,             //     Input.....(Data, Variable, Absolute)
0x09, 0x33,             //     Usage (Touch)
0x15, 0x00,             //     Logical Minimum..... (0)
0x25, 0x01,             //     Logical Maximum..... (1)
0x75, 0x01,             //     Report Size..... (1)
0x95, 0x01,             //     Report Count..... (1)
0x81, 0x02,             //     Input.....(Data, Variable, Absolute)
```

Confidential Property of Danny Wu
Unauthorized Distribution Prohibited

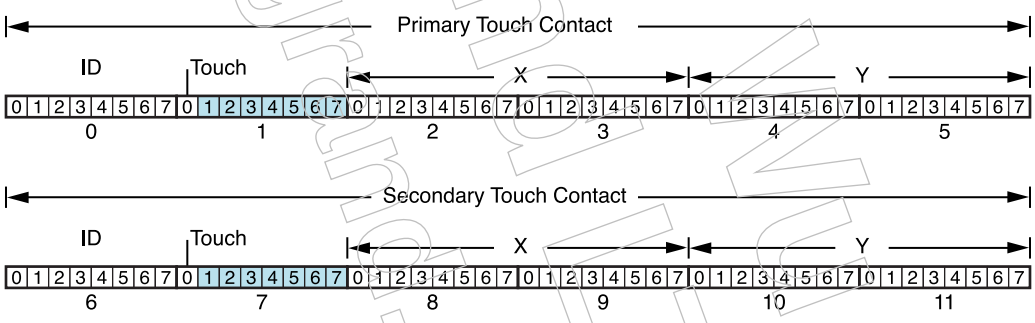
```
0x75, 0x07,          //     Report Size..... (7)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x01,          //     Input.................(Constant)
0x05, 0x01,          //     Usage Page (Generic Desktop)
0x09, 0x30,          //     Usage (X)
0x15, 0x00,          //     Logical Minimum..... (0)
0x26, 0x20, 0x03,    //     Logical Maximum..... (800)
0x75, 0x10,          //     Report Size..... (16)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0x09, 0x31,          //     Usage (Y)
0x15, 0x00,          //     Logical Minimum..... (0)
0x26, 0xE0, 0x01,    //     Logical Maximum..... (480)
0x75, 0x10,          //     Report Size..... (16)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0xC0,                //   End Collection
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x22,          //   Usage (Finger)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x38,          //     Usage (Transducer Index)
0x75, 0x08,          //     Report Size..... (8)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0x09, 0x33,          //     Usage (Touch)
0x15, 0x00,          //     Logical Minimum..... (0)
0x25, 0x01,          //     Logical Maximum..... (1)
0x75, 0x01,          //     Report Size..... (1)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0x75, 0x07,          //     Report Size..... (7)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x01,          //     Input.................(Constant)
```

Carvuio@outlook.com Carvuio Wu Confidential

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

```
0x05, 0x01,          //     Usage Page (Generic Desktop)
0x09, 0x30,          //     Usage (X)
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0x20, 0x03,    //     Logical Maximum......... (800)
0x75, 0x10,          //     Report Size.............. (16)
0x95, 0x01,          //     Report Count............. (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0x09, 0x31,          //     Usage (Y)
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0xE0, 0x01,    //     Logical Maximum......... (480)
0x75, 0x10,          //     Report Size.............. (16)
0x95, 0x01,          //     Report Count............. (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0xC0,                //   End Collection
0xC0,                // End Collection
```

Figure 19-10 Example Input Report Layout for Multi-Touch Screen



19.3.5.2 Character Input Gesture Support

Note: This section contains provisional content pending ratification and subject to change.

We are seeking to add functional enhancements to the HID Digitizer Page (0x0D) to convey the accessory's ability to process character generating gestures. Though there already exists a HID unicode page, it is limited to USC-2 (UTF16-LE). As a result, this prevents accessories from fully supporting certain Asian character sets.

We are proposing the addition of support for the transmitting of character strings with alternate encodings such as UTF8, UTF16 and UTF32.

Confidential Property of Apple Inc.

The accessory may support the following HID usages:

Table 19-46 Character Input Gesture Support HID Usages

Page ID	Page Name	Usage ID	Usage Name	Usage Type
0x0D	Digitizer	0x23	Device Settings	Logical Collection
0x0D	Digitizer	0x24	Character Gesture	Logical Collection
0x0D	Digitizer	0x60	Character Gesture Enable	Dynamic Flag
0x0D	Digitizer	0x61	Character Gesture Quality	Dynamic Value
0x0D	Digitizer	0x62	Character Gesture Data Length	Buffered Bytes
0x0D	Digitizer	0x63	Character Gesture Data	Dynamic Value
0x0D	Digitizer	0x64	Character Gesture Encoding	Named Array
0x0D	Digitizer	0x65	UTF8 Character Gesture Encoding	Selector
0x0D	Digitizer	0x66	UTF16 Little Endian Character Gesture Encoding	Selector
0x0D	Digitizer	0x67	UTF16 Big Endian Character Gesture Encoding	Selector
0x0D	Digitizer	0x68	UTF32 Little Endian Character Gesture Encoding	Selector
0x0D	Digitizer	0x69	UTF32 Big Endian Character Gesture Encoding	Selector

When the Apple device does not own the screen, the accessory must not send these HID usages.

19.3.5.2.1 Basic Character Gesture Recognition

Basic character gesture recognition allows an accessory to convey a single character string as a result of interpreting transducer movement on a digitizer surface.

In order for the Apple device to properly detect support for these gestures, the accessory must declare a logical collection with the following usages:

- Character Gesture
- Character Gesture Data Length

Unlicensed Apple Product
Confidential
Do not distribute

- Character Gesture Data

Additionally, the accessory must also include string encoding information. If more than one encoding type is supported, they must be placed in a selector array. Otherwise, the accessory may declare individual encoding support via a static item. Use the following usages:

- Character Gesture Encoding
- UTF8 Character Gesture Encoding
- UTF16 Little Endian Character Gesture Encoding
- UTF16 Big Endian Character Gesture Encoding
- UTF32 Little Endian Character Gesture Encoding
- UTF32 Big Endian Character Gesture Encoding

The Apple device will notify the accessory of the input mode using the hidSetInputMode as described in hidSetInputMode (page 321).

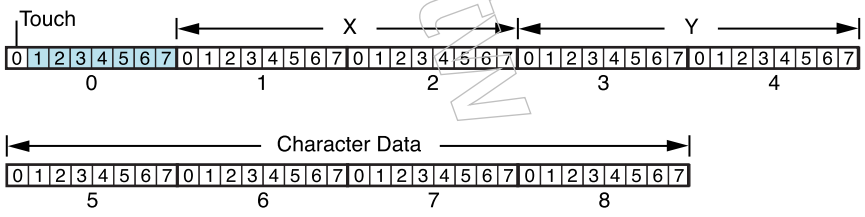
The following is an example report descriptor and format for a single-touch touchpad with basic character gesture recognition:

```
0x05, 0x0D,          // Usage Page (Digitizer)
0x09, 0x05,          // Usage (Touch Pad)
0xA1, 0x01,          // Collection (Application)
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x22,          //   Usage (Finger)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x33,          //     Usage (Touch)
0x15, 0x00,          //     Logical Minimum......... (0)
0x25, 0x01,          //     Logical Maximum......... (1)
0x75, 0x01,          //     Report Size............. (1)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0x75, 0x07,          //     Report Size............. (7)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x01,          //     Input...................(Constant)
0x05, 0x01,          //     Usage Page (Generic Desktop)
0x09, 0x30,          //     Usage (X)
```

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

```
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0x00, 0x04,    //     Logical Maximum......... (1024)
0x75, 0x10,          //     Report Size............. (16)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0x09, 0x31,          //     Usage (Y)
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0x00, 0x04,    //     Logical Maximum......... (1024)
0x75, 0x10,          //     Report Size............. (16)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0xC0,                //   End Collection
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x24,          //   Usage (Gesture Character)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x63,          //     Usage (Gesture Character Data)
0x75, 0x20,          //     Report Size............. (32)
0x95, 0x01,          //     Report Count............ (1)
0x82, 0x02, 0x01,    //     Input...................(Data, Variable, Absolute,
Buffered bytes)
0x09, 0x65,          //     Usage (Gesture Character Encoding UTF8)
0x09, 0x62,          //     Usage (Gesture Character Data Length)
0x75, 0x08,          //     Report Size............. (8)
0x95, 0x02,          //     Report Count............ (2)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0xC0,                //   End Collection
0xC0,                // End Collection
```

Figure 19-11 Example Input Report Layout for Single-Touch Touchpad with Basic Character Gesture Recognition



Apple Confidential
Do Not Distribute

When reporting character data, care must taken to ensure proper processing of repeated characters. This must be accomplished by issuing a subsequent report that clears gesture data and data length for a given character collection. Using the report format from the example above, the following sequence illustrates how this is accomplished ignoring current finger touch and location states:

First HID report dispatched containing gesture event for the character 'A':

```
XX XX XX XX XX 41 00 00 00 01 65
```

Second HID report dispatched clearing the gesture:

```
XX XX XX XX XX 00 00 00 00 00 65
```

19.3.5.2.2 Character Gesture Recognition with Alternate Interpretations

There can be situations in which the recognition system generates more than one interpretation of a gesture motion. We are proposing the ability to convey alternate gesture interpretations from a single accessory which will allow the Apple device to select the appropriate string based on its current application context.

Each gesture item follows the requirements detailed in Basic Character Gesture Recognition, but must also include a declaration for Character Gesture Quality in each logical collection. This will give the Apple device additional qualitative information so that it can select the appropriate interpretation.

In addition, if no alternative interpretations are available, the recognition system must inform the Apple device by ensuring that only the first character is populated and all subsequent characters are cleared.

The following is an example report descriptor and format for a single-touch touchpad with character gesture recognition with alternate interpretations:

```
0x05, 0x0D,          // Usage Page (Digitizer)
0x09, 0x05,          // Usage (Touch Pad)
0xA1, 0x01,          // Collection (Application)
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x22,          //   Usage (Finger)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x33,          //     Usage (Touch)
0x15, 0x00,          //     Logical Minimum......... (0)
0x25, 0x01,          //     Logical Maximum......... (1)
0x75, 0x01,          //     Report Size............. (1)
```

Apple Inc. Copyright © 2014 Apple Inc. All Rights Reserved.

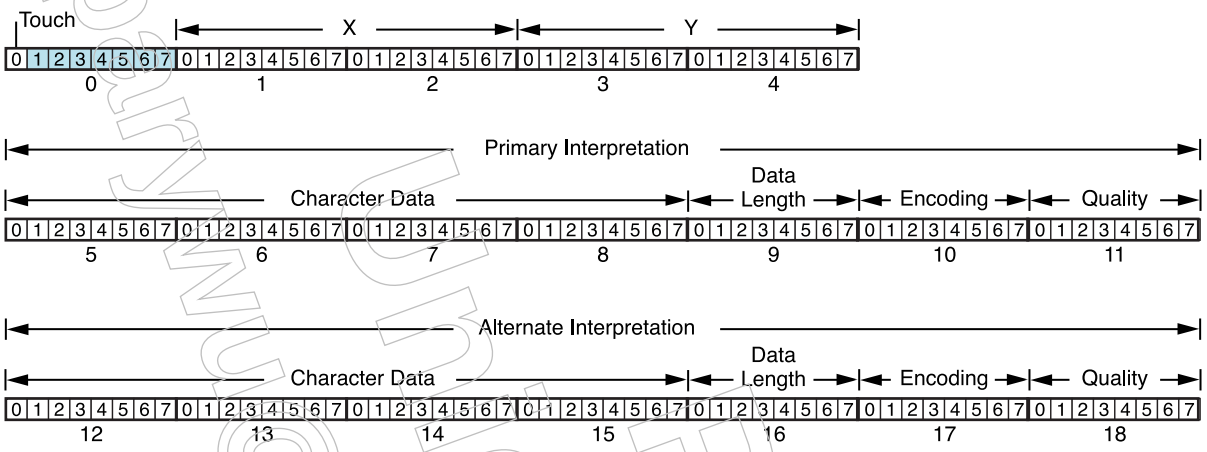
```
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0x75, 0x07,          //     Report Size..... (7)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x01,          //     Input.................(Constant)
0x05, 0x01,          //     Usage Page (Generic Desktop)
0x09, 0x30,          //     Usage (X)
0x15, 0x00,          //     Logical Minimum..... (0)
0x26, 0x00, 0x04,    //     Logical Maximum..... (1024)
0x75, 0x10,          //     Report Size..... (16)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0x09, 0x31,          //     Usage (Y)
0x15, 0x00,          //     Logical Minimum..... (0)
0x26, 0x00, 0x04,    //     Logical Maximum..... (1024)
0x75, 0x10,          //     Report Size..... (16)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0xC0,                //   End Collection
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x24,          //   Usage (Gesture Character)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x63,          //     Usage (Gesture Character Data)
0x75, 0x20,          //     Report Size..... (32)
0x95, 0x01,          //     Report Count..... (1)
0x82, 0x02, 0x01,    //     Input.................(Data, Variable, Absolute,
Buffered bytes)
0x09, 0x65,          //     Usage (Gesture Character Encoding UTF8)
0x09, 0x62,          //     Usage (Gesture Character Data Length)
0x75, 0x08,          //     Report Size..... (8)
0x95, 0x02,          //     Report Count..... (2)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0x09, 0x61,          //     Usage (Gesture Character Quality)
0x15, 0x00,          //     Logical Minimum..... (0)
```

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

```
0x25, 0x64,          //     Logical Maximum......... (100)
0x75, 0x08,          //     Report Size............. (8)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0xC0,                //   End Collection
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x24,          //   Usage (Gesture Character)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x63,          //     Usage (Gesture Character Data)
0x75, 0x20,          //     Report Size............. (32)
0x95, 0x01,          //     Report Count............ (1)
0x82, 0x02, 0x01,    //     Input...................(Data, Variable, Absolute,
Buffered bytes)
0x09, 0x65,          //     Usage (Gesture Character Encoding UTF8)
0x09, 0x62,          //     Usage (Gesture Character Data Length)
0x75, 0x08,          //     Report Size............. (8)
0x95, 0x02,          //     Report Count............ (2)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0x09, 0x61,          //     Usage (Gesture Character Quality)
0x15, 0x00,          //     Logical Minimum......... (0)
0x25, 0x64,          //     Logical Maximum......... (100)
0x75, 0x08,          //     Report Size............. (8)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0xC0,                //   End Collection
0xC0,                // End Collection
```

Barry Wu@quinetapply.com.tw

Figure 19-12 Example Input Report Layout for Single-Touch Touchpad with Character Gesture Recognition with Alternate Interpretations

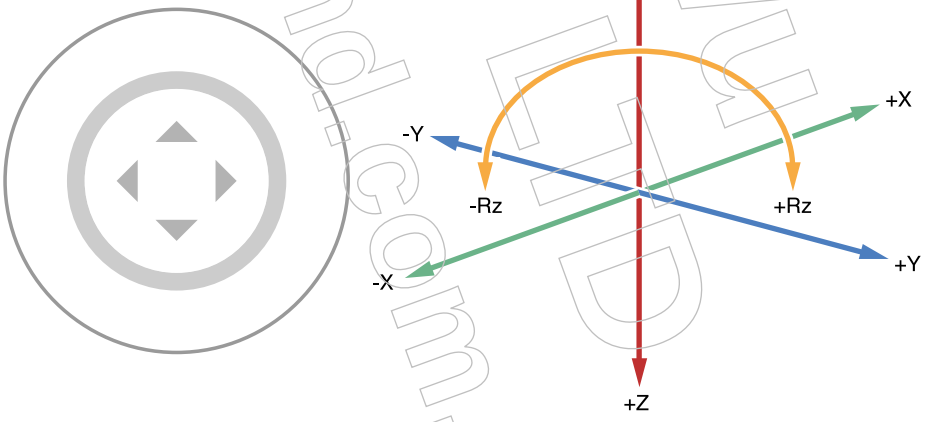


19.3.5.3 Knob Support (Multi-axis Controller)

Knobs refer to user input devices, commonly found on a vehicle center console, which allow for indirect interaction with a media display.

A multi-axis controller is similar to a joystick but minimally consists of three variable axes (X, Y and Z). Additionally, rotational axes for X, Y, and Z may also be included.

Figure 19-13 Multi-axis Controller



The accessory may support the following HID usages:

Copyright © 2014 Apple Inc. All Rights Reserved.

Table 19-47 Knob Support HID Usages

Page ID	Page Name	Usage ID	Usage Name	Usage Type
0x01	Generic	0x30	X	Dynamic Value
0x01	Generic	0x31	Y	Dynamic Value
0x01	Generic	0x32	Z	Dynamic Value
0x01	Generic	0x35	Rz	Dynamic Value
0x01	Generic	0x37	Dial	Dynamic Value
0x01	Generic	0x38	Wheel	Dynamic Value
0x09	Button	0x01	Button 1 (Primary Button)	(see USB HID)
0x0D	Consumer	0x224	AC Back	One Shot Control

When the Apple device does not own the screen, the accessory must not send these HID usages.

The preferred state of the axes should signify that the knob rests at the center when not in use.

The accessory must support the following to convey translational movement:

- X
- Y

The accessory must support either one of the following to convey selection:

- Z
- Button 1 (Primary Button)

When implementing the Z axis, the Apple device only requires movement in the +Z (down) direction and will translated it to a primary button.

The accessory must support either one of the following to allow scrolling:

- Rz
- Dial
- Wheel

The accessory must support the AC Back button.

Bowu@uniland.com.tw Uniland LTD

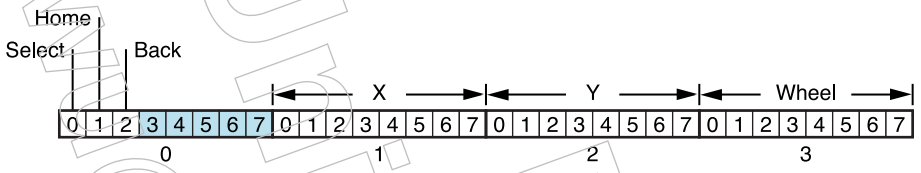
The following is an example report descriptor and format for a multi-axis controller:

```
0x05, 0x01,         // Usage Page (Generic Desktop)
0x09, 0x08,         // Usage (MultiAxisController)
0xA1, 0x01,         // Collection (Application)
0x05, 0x09,         //   Usage Page (Button)
0x09, 0x01,         //   Usage 1 (0x1)
0x15, 0x00,         //   Logical Minimum......... (0)
0x25, 0x01,         //   Logical Maximum......... (1)
0x75, 0x01,         //   Report Size............. (1)
0x95, 0x01,         //   Report Count............ (1)
0x81, 0x02,         //   Input...................(Data, Variable, Absolute)
0x05, 0x0C,         //   Usage Page (Consumer)
0x0A, 0x23, 0x02,   //   Usage 547 (AC Home)
0x0A, 0x24, 0x02,   //   Usage 548 (AC Back)
0x15, 0x00,         //   Logical Minimum......... (0)
0x25, 0x01,         //   Logical Maximum......... (1)
0x75, 0x01,         //   Report Size............. (1)
0x95, 0x02,         //   Report Count............ (2)
0x81, 0x02,         //   Input...................(Data, Variable, Absolute)
0x75, 0x05,         //   Report Size............. (5)
0x95, 0x01,         //   Report Count............ (1)
0x81, 0x01,         //   Input...................(Constant)
0x05, 0x01,         //   Usage Page (Generic Desktop)
0x09, 0x30,         //   Usage (X)
0x09, 0x31,         //   Usage (Y)
0x15, 0x81,         //   Logical Minimum......... (-127)
0x25, 0x7F,         //   Logical Maximum......... (127)
0x75, 0x08,         //   Report Size............. (8)
0x95, 0x02,         //   Report Count............ (2)
0x81, 0x02,         //   Input...................(Data, Variable, Absolute)
0x05, 0x01,         //   Usage Page (Generic Desktop)
0x09, 0x38,         //   Usage (Wheel)
0x15, 0x81,         //   Logical Minimum......... (-127)
0x25, 0x7F,         //   Logical Maximum......... (127)
```

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

```
0x75, 0x08,          //   Report Size..... (8)
0x95, 0x01,          //   Report Count..... (1)
0x81, 0x06,          //   Input.................(Data, Variable, Relative)
0xC0,                // End Collection
```

Figure 19-14 Example Input Report Layout for Multi-Axis Controller



19.3.5.4 Buttons

The CarPlay feature supports various buttons commonly found within the center console or steering wheel of a vehicle.

The accessory may support the following HID usages:

Table 19-48 Button Support HID Usages

Page ID	Page Name	Usage ID	Usage Name	Usage Type	Apple Function
0x0C	Consumer	0xB0	Play	On/Off Control	Play
0x0C	Consumer	0xB1	Pause	On/Off Control	Pause
0x0C	Consumer	0xB5	Scan Next Track	One Shot Control	Scan Next Track
0x0C	Consumer	0xB6	Scan Previous Track	One Shot Control	Scan Previous Track
0x0C	Consumer	0xCD	Play/Pause	One Shot Control	Toggle Play/Pause
0x0C	Consumer	0x223	AC Home	One Shot Control	CarPlay
0x0C	Consumer	0x224	AC Back	One Shot Control	Back
0x0B	Telephony	0x20	Hook Switch	On/Off Control	Accept call

Pary Wu
Universal
Uniqueband.com

Page ID	Page Name	Usage ID	Usage Name	Usage Type	Apple Function
0x0B	Telephony	0x21	Flash	Momentary Control	Toggle Accept or Reject/End call
0x0B	Telephony	0x26	Drop	One Shot Control	Reject/End call
0x0B	Telephony	0x2F	Phone Mute	One Shot Control	Mute the microphone
0x0B	Telephony	0xB0	Phone Key 0	Selector	Phone Key 0
0x0B	Telephony	0xB1	Phone Key 1	Selector	Phone Key 1
0x0B	Telephony	0xB2	Phone Key 2	Selector	Phone Key 2
0x0B	Telephony	0xB3	Phone Key 3	Selector	Phone Key 3
0x0B	Telephony	0xB4	Phone Key 4	Selector	Phone Key 4
0x0B	Telephony	0xB5	Phone Key 5	Selector	Phone Key 5
0x0B	Telephony	0xB6	Phone Key 6	Selector	Phone Key 6
0x0B	Telephony	0xB7	Phone Key 7	Selector	Phone Key 7
0x0B	Telephony	0xB8	Phone Key 8	Selector	Phone Key 8
0x0B	Telephony	0xB9	Phone Key 9	Selector	Phone Key 9
0x0B	Telephony	0xBA	Phone Key Star	Selector	Phone Key Star
0x0B	Telephony	0xBB	Phone Key Pound	Selector	Phone Key Pound

Accessories with non-touch UIs must implement AC Back.

Accessories must implement at least one of the following:

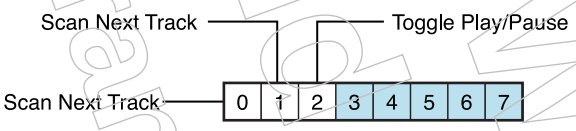
- Flash
- Hook Switch & Drop

The following is an example report descriptor and format for a simple media buttons accessory:

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

```
0x05, 0x0C,          // Usage Page (Consumer)
0x09, 0x01,          // Usage 1 (0x1)
0xA1, 0x01,          // Collection (Application)
0x05, 0x0C,          //   Usage Page (Consumer)
0x09, 0xB5,          //   Usage 181 (Scan Next Track)
0x09, 0xB6,          //   Usage 182 (Scan Previous Track)
0x09, 0xCD,          //   Usage 205 (Toggle Play / Pause)
0x15, 0x00,          //   Logical Minimum......... (0)
0x25, 0x01,          //   Logical Maximum......... (1)
0x75, 0x01,          //   Report Size............. (1)
0x95, 0x03,          //   Report Count............ (3)
0x81, 0x02,          //   Input...................(Data, Variable, Absolute)
0x75, 0x05,          //   Report Size............. (5)
0x95, 0x01,          //   Report Count............ (1)
0x81, 0x01,          //   Input...................(Constant)
0xC0,                // End Collection
```

Figure 19-15 Example Input Report Layout for Simple Media Buttons



The following is an example report descriptor and format for a simple telephone buttons accessory with flash and numeric keys:

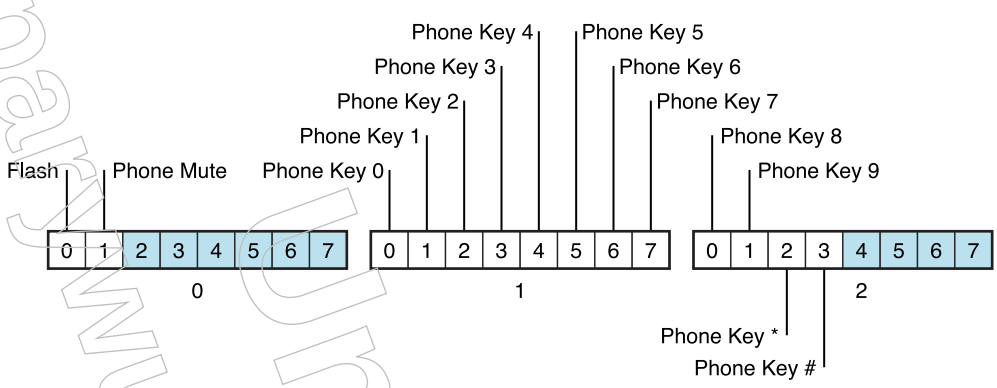
```
0x05, 0x0B,          // Usage Page (Telephony Device)
0x09, 0x01,          // Usage 1 (0x1)
0xA1, 0x01,          // Collection (Application)
0x05, 0x0B,          //   Usage Page (Telephony Device)
0x09, 0x21,          //   Usage 33 (Flash)
0x09, 0x2F,          //   Usage 47 (Phone Mute)
0x15, 0x00,          //   Logical Minimum......... (0)
0x25, 0x01,          //   Logical Maximum......... (1)
0x75, 0x01,          //   Report Size............. (1)
0x95, 0x02,          //   Report Count............ (2)
```

Early Draft Confidential

```
0x81, 0x02,         //   Input..................(Data, Variable, Absolute)
0x75, 0x06,         //   Report Size..... (6)
0x95, 0x01,         //   Report Count..... (1)
0x81, 0x01,         //   Input..................(Constant)
0x05, 0x0B,         //   Usage Page (Telephony Device)
0x09, 0xB0,         //   Usage 176 (Phone Key 0)
0x09, 0xB1,         //   Usage 177 (Phone Key 1)
0x09, 0xB2,         //   Usage 178 (Phone Key 2)
0x09, 0xB3,         //   Usage 179 (Phone Key 3)
0x09, 0xB4,         //   Usage 180 (Phone Key 4)
0x09, 0xB5,         //   Usage 181 (Phone Key 5)
0x09, 0xB6,         //   Usage 182 (Phone Key 6)
0x09, 0xB7,         //   Usage 183 (Phone Key 7)
0x09, 0xB8,         //   Usage 184 (Phone Key 8)
0x09, 0xB9,         //   Usage 185 (Phone Key 9)
0x09, 0xBA,         //   Usage 186 (Phone Key *)
0x09, 0xBB,         //   Usage 187 (Phone Key #)
0x15, 0x00,         //   Logical Minimum......... (0)
0x25, 0x01,         //   Logical Maximum......... (1)
0x75, 0x01,         //   Report Size..... (1)
0x95, 0x0C,         //   Report Count..... (12)
0x81, 0x02,         //   Input..................(Data, Variable, Absolute)
0x75, 0x04,         //   Report Size..... (4)
0x95, 0x01,         //   Report Count..... (1)
0x81, 0x01,         //   Input..................(Constant)
0xC0,               // End Collection
```

Pary Wu
parywu@ftdi.com.tw
FTDI

Figure 19-16 Example Input Report Layout for Simple Telephone Buttons



19.3.6 Siri User Input

Accessories may support the ability for a user to activate Siri. This is communicated to the controller via the requestSiri command (see requestSiri (page 322)), with the parameter being one of the following Siri actions:

Table 19-49 Siri actions enum values

Name	Value	Description
Prewarm	1	Indicate that the Apple device should begin preparing Siri. At this point, no audio or video resources will be taken.
ButtonDown	2	Indicate to the Apple device that the Siri button has been depressed.
ButtonUp	3	Indicate to the Apple device that the Siri button has been released.

Accessories may wish to overload their existing "speech recognition" button, where a short press activates the accessory's native speech recognition system, while a longer press activates Siri. The recommended value for the longer press timeout is 600 ms. It is important that the accessory prewarm Siri as early as possible (for example, when the physical button is pressed) to ensure the best possible response time from Siri in the event that it is activated. After the inital Siri start up sequence, the accessory must forward all physical button presses to the device without any additional delays. Once the Siri session is finished, the accessory must return to the button's default behavior.

Short press (accessory speech recognition):

```
Physical button depressed
|        Physical button released
|        |
+--------+-------------------------------------------------> time
```

Partwu@uniontech.com.tw Partwu@uniontech.com.tw

```
|
|
Prewarm action sent
```

For short presses, where the physical button is released before the accessory-determined timeout interval, only the Prewarm action is sent to the controller, and the accessory's native speech recognition system is activated.

Long press (Siri):

```
Physical button depressed
|                Long Press Timeout       Physical button released
|                |                        |
+----------------+------------------------+-----------------------> time
|                |                        |
|                ButtonDown action sent   ButtonUp action sent
Prewarm action sent
```

For long presses, where the physical button is held down past the accessory-determined timeout interval, the Prewarm and ButtonUp actions are sent to the controller, just like in the previous scenario, but when the timeout interval is reached, the accessory sends an additional ButtonDown action when the timeout interval has elapsed.

For accessories that have a dedicated Siri button, sending a Prewarm action is unnecessary and ButtonDown / ButtonUp actions are sent when the button is physically depressed and released.

19.3.7 Commands

Commands are messages exchanged between the Apple device and the accessory to perform an action. They are sent as HTTP requests using the `/command` URL on the control stream. The Apple device sends its commands over the Controller control channel while the accessory sends its commands over the Accessory control channel. Each command receives an HTTP response. The request and response payloads are binary plists. Each request contains a key, type, to indicate the command to perform. Additional keys may be included for command-specific parameters.

For example, an hidSendReport command would look like this (in XML for readability, the actual request is a binary plist):

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>
    <key>hidReport</key>
    <data>
    ABEiM0Q=
    </data>
    <key>type</key>
    <string>hidSendReport</string>
    <key>uuid</key>
    <string>cbb63f31-c020-4db9-ac16-2283dda2a079</string>
</dict>
</plist>
```

19.3.7.1 duckAudio

Sender: Apple device

Description: Ramps volume down (for example, to fade down music to play a voice prompt). See Ducking (page 268). Table 19-50 (page 317) lists the specifics of duckAudio request keys.

Table 19-50 duckAudio request keys

Key	Type	Required?	Description
durationMs	number	Y	Number of milliseconds the ramp down should last.
volume	number	Y	Suggested final dB attenuation of the audio at the end of the duck (-144 to 0 dB).

19.3.7.2 unduckAudio

Sender: Apple device

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Description: Ramps volume back to the pre-duck level. See Ducking (page 268). Table 19-51 (page 318) lists the specifics of unduckAudio request keys.

Table 19-51 unduckAudio request keys

Key	Type	Required?	Description
durationMs	number	Y	Number of milliseconds the ramp up should last.

19.3.7.3 disableBluetooth

Sender: Apple device

Description: Disable Bluetooth connectivity to specified device (i.e. the Apple device). If the accessory supports Bluetooth, it must provide its bluetoothIDs (see Info Message (page 273)) in order to receive a disableBluetooth command from the device. At a minimum, on receipt of this command, the accessory must immediately disable any existing connections with the specified Apple device for the following Bluetooth profiles:

- Hands-Free Profile
- Advanced Audio Distribution Profile
- A/V Remote Control Profile

In addition, the accessory must not attempt to connect any of the above-listed Bluetooth profiles with the same Apple device while the two are already connected over CarPlay (that is, a session is already in progress). Table 19-52 (page 318) lists the specifics of disableBluetooth request keys.

Table 19-52 disableBluetooth request keys

Key	Type	Required?	Description
deviceID	string	Y	MAC address of Bluetooth device to disable connectivity to (that is, the Apple device).

19.3.7.4 changeModes

Sender: Accessory

Description: Change resource states and/or app states. See Modes (page 290). Table 19-53 (page 319) lists the specifics of changeModes request keys and Table 19-56 (page 320) lists the specifics of changeModes response keys.

Third Party Wu
Wu@thirdpartywu.com
Unauthorized LTD

Table 19-53 changeModes request keys

Key	Type	Required?	Description
appStates	array	N	Application state(s) to change. Contains dictionaries of Table 19-54 (page 319).
resources	array	N	Resource ownership(s) to change. Contains dictionaries of Table 19-55 (page 319).
reasonStr	string	N	A textual description of the reason for the mode change.

Table 19-54 appState request dictionary request keys

Key	Type	Required?	Description
appStateID	enum	Y	One of Table 19-42 (page 292).
speechMode	enum	N (*)	(*) Required if appStateID is Speech. One of Table 19-43 (page 292).
state	boolean	N (*)	(*) Required if appStateID is PhoneCall or TurnByTurn.

Table 19-55 resource request dictionary request keys

Key	Type	Required?	Description
resourceID	enum	Y	One of Table 19-38 (page 290).
transferType	enum	Y	One of Table 19-40 (page 291).
transferPriority	enum	N (*)	(*) Required if transferType is take or borrow. One of Table 19-41 (page 291).
takeConstraint	enum	N (*)	(*) Required if transferType is take. One of Table 19-39 (page 291).
borrowConstraint	enum	N (*)	(*) Required if transferType is take. One of Table 19-39 (page 291).
unborrowConstraint	enum	N (*)	(*) Required if transferType is borrow. One of Table 19-39 (page 291).

Purchased by Unlimited Technology (unlimitedtechnology@gmail.com)

Table 19-56 changeModes response keys

Key	Type	Required?	Description
params	dictionary	N	Updated modes if the mode change was successful, otherwise absent. The modes dictionary uses the same keys as the "changeModes" command.
status	number	Y	Result of performing the mode change. If the value is nonzero, the change failed.

19.3.7.5 modesChanged

Sender: Apple device

Description: Updates the accessory's current mode after a mode change by the Apple device. Table 19-57 (page 320) lists the specifics of modesChanged request keys.

Table 19-57 modesChanged request keys

Key	Type	Required?	Description
appStates	array	N	Application state(s) to change. Contains dictionaries of Table 19-58 (page 320).
resources	array	N	Resource ownership(s) to change. Contains dictionaries of Table 19-59 (page 320).

Table 19-58 appState dictionary keys

Key	Type	Required?	Description
appStateID	enum	Y	One of Table 19-42 (page 292).
entity	enum	Y	One of Table 19-37 (page 290).
speechMode	enum	N (*)	(*) Required if appStateID is Speech. One of Table 19-43 (page 292).

Table 19-59 resource dictionary keys

Key	Type	Required?	Description
resourceID	enum	Y	One of Table 19-38 (page 290).
entity	enum	Y	Value must not be 'None'. One of Table 19-37 (page 290).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

19.3.7.6 forceKeyFrame

Sender: Accessory

Description: The accessory may force a key frame to be sent for the screen stream.

19.3.7.7 hidSendReport

Sender: Accessory

Description: When an HID event occurs on the accessory, such as the user turning a knob, it sends it to the Apple device over the control stream by sending a command with a type of hidSendReport to the `/command` URL. The command contains the USB HID report and UUID of the HID device. Table 19-60 (page 321) lists the HID event keys.

Table 19-60 hidSendReport request keys

Key	Type	Required?	Description
hidReport	data	Y	USB-formatted HID report.
timestamp	number	N	NTP timestamp when the event occurred (synchronized to the controller's clock).
uuid	string	Y	UUID to uniquely identify the HID device.

19.3.7.8 hidSetInputMode

Sender: Apple device

Description: Sets input mode on a HID.

Table 19-61 hidSetInputMode request keys

Key	Type	Required?	Description
hidInputMode	enum	Y	One of Table 19-62 (page 321).
uuid	string	Y	UUID to uniquely identify the HID device.

Table 19-62 HID input modes enum values

Name	Value	Description
Default	0	Default mode for non-character input (e.g. panning).

Paul Wu Unlimited LTD
paul@unlimited.com.tw

Name	Value	Description
Character	1	Optimized for entering characters.
Scrolling	2	Hint to interpret scroll gestures and send wheel HID events.
ScrollingWithCharacters	3	Send characters and scrolls as wheel HID events.
DialPad	4	Send 0-9, #, * + character events only.

19.3.7.9 requestSiri

Sender: Accessory

Description: Requests that Siri be invoked with a specified action.

The requestSiri command must only be sent as a result of direct user action on a physical or virtual control surface. The behavior should match that of hidSendReport (page 321), see HID Requirements (page 407).

Table 19-63 requestSiri request keys

Key	Type	Required?	Description
siriAction	enum	Y	One of Table 19-49 (page 315).

19.3.7.10 requestUI

Sender: Either Apple device or accessory

Description: For the Apple device to ask for the accessory UI to be shown, or for the accessory to ask for the CarPlay UI to be shown. The accessory must not send requestUI upon connection; the Apple device will send requestUI, if appropriate.

The requestUI command must only be sent as a result of direct user action on a physical or virtual control surface. The accessory must not send requestUI upon connection; the Apple device will request resources based on the modes provided in the Info message. See HID Requirements (page 407).

Where the sender is the accessory, a URL request key (see Table 19-64 (page 323)) is available to specify the desired CarPlay compatible app to be shown.

Note: Only URLs that do not require user interaction on the Apple device are allowed.

Party Wu (partywu@uniquevisionic.com)

Table 19-64 Accessory to Apple device requestUI request keys

Key	Type	Required?	Description
url	string	N	URL identifier of the desired CarPlay UI application to launch: • no url - the CarPlay screen will be shown. • "maps:" - the CarPlay Maps application will be shown. • "mobilephone:" - the CarPlay Phone application will be shown. • "music:" - the CarPlay Music application will be shown. • "nowplaying:" - the Now Playing screen will be shown. • "tel:xxx-xxx-xxxx" - the CarPlay Phone application will be shown and a phone call with the desired number will be placed (for support phone number formats see *IETF RFC 3966* .

19.3.7.11 setNightMode

Sender: Accessory

Description: The accessory may indicate whether it is dark outside.

Table 19-65 setNightMode request keys

Key	Type	Required?	Description
nightMode	boolean	Y	True if it is dark outside, false otherwise.

19.3.7.12 setLimitedUI

Sender: Accessory

Description: The accessory may indicate whether or not to limit certain UI elements. See Table 19-22 (page 279).

Table 19-66 setLimitedUI request keys

Key	Type	Required?	Description
limitedUI	boolean	Y	True if certain UI elements should be limited.

Party Wu (partywu@unibrand.com.tw) Unibrand LTD

19.3.8 CarPlay Communication Plug-in

The CarPlay Communication Plug-in, source code available to accessory developers with approved CarPlay product plans, implements the core functionality described in the preceding sections of CarPlay Communication Protocol (page 268). It interfaces with Bonjour to configure and advertise the CarPlay service and establishes the communication and data links between the accessory and Apple device.

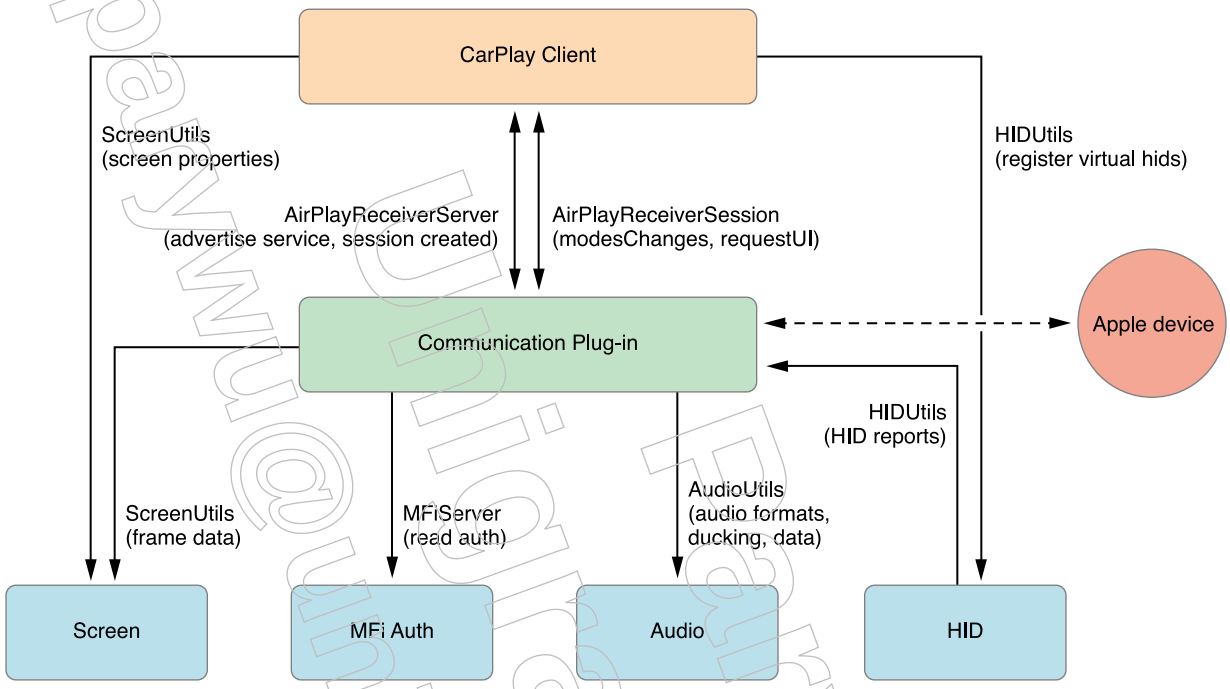
The accessory maker is not required to implement their own TCP and UDP channels for either setup and control or audio and UI transfer. Instead, these are automatically created by the Communication Plug-in which also manages authentication and encryption-decryption of all CarPlay channels.

For simplicity, the core info, setup and feedback messages are abstracted into a platform API, as they are any commands and user event. Figure 19-17 (page 325) shows an architectural overview of the CarPlay Communication Plug-in. To integrate the plug-in on a specific platform, a set of utility functions has been provided:

- **MFiServer** provides integration with platform specific installation of an MFi authentication chip, e.g. over I2C bus. The plug-in will automatically read out authentication data for encrypting the audio and video streams.
- **ScreenUtils** provides integration with the platform specific video decoding interfaces. During setup, a CarPlay client application can configure the specific screen properties. The plug-in will then directly send video frames through the provided custom APIs.
- **AudioUtils** provides integration with the platform specific audio interfaces. On each audio stream setup the plug-in will provide information about the audio formats and data, as well as forward requests to duck the current audio playback.
- **HIDUtils** provides integration with the platform specific HID devices (touchscreens, knob-based controls, etc.). A CarPlay client application can first register the available HID devices and then post HID reports to the plug-in.
- **AirPlayReceiverServer** / **AirPlayReceiverServerDelegate** - A CarPlay client application can use the delegate to send and receive server-level commands to the device, as well as obtain a reference to an active AirPlayReceiverServerSession.
- **AirPlayReceiverServerSession** / **AirPlayReceiverServerSessionDelegate** - A CarPlay client application can use the delegate to send and receive session-level commands to the device. Typical examples will be notifications for modeChanged, or commands as requestUI.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

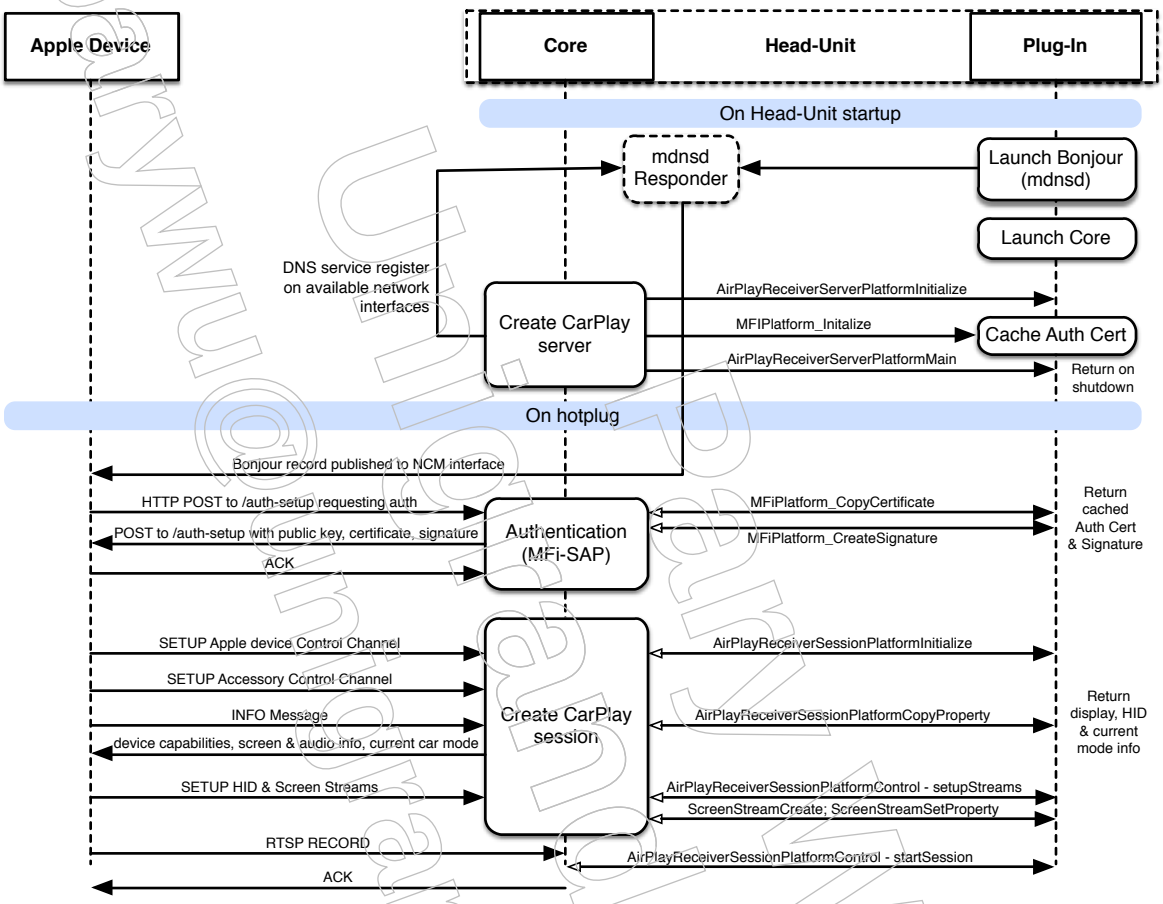
Figure 19-17 CarPlay Communication Plug-in Reference Architecture



An illustration of this platform abstraction is included in the following use case sequences, shown in Figure 19-18 (page 326), Figure 19-19 (page 327), Figure 19-20 (page 328), and Figure 19-21 (page 329).

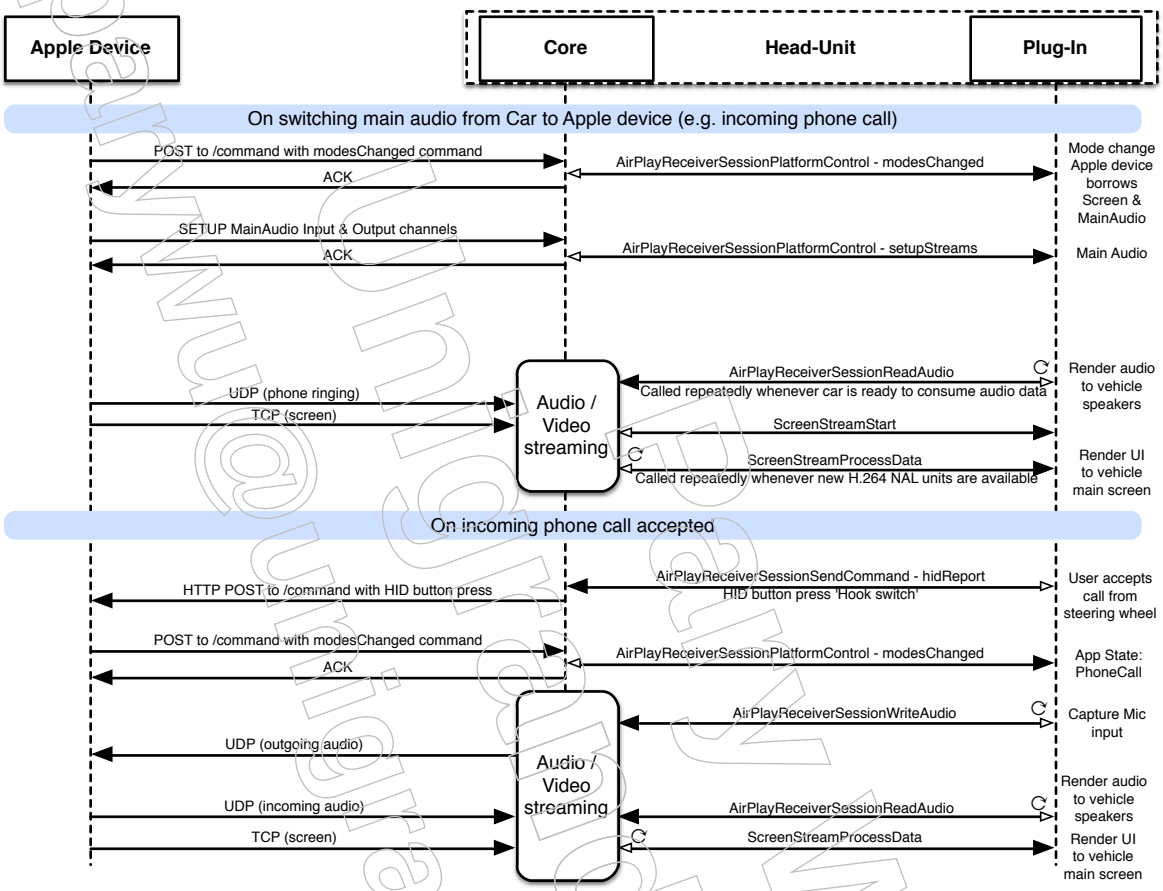
paywu@uniland.com.tw Uniland LTD Jay Wu

Figure 19-18 Control sequence to initiate a CarPlay session



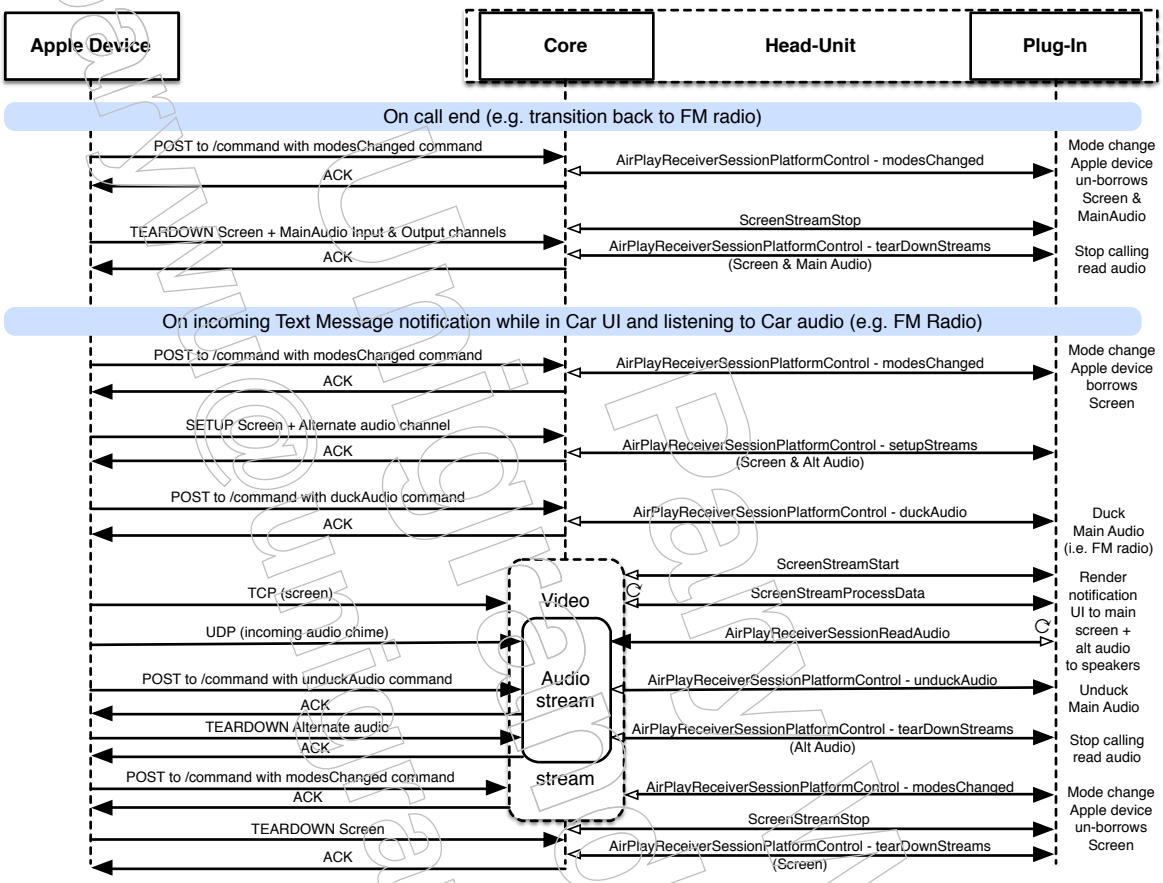
peiyu@eland.com.tw Eland Peiyu Wu LTD

Figure 19-19 Control sequence for switching the main audio channel from accessory to Apple device



peiwu@unionbrand.com.tw Unionbrand LTD

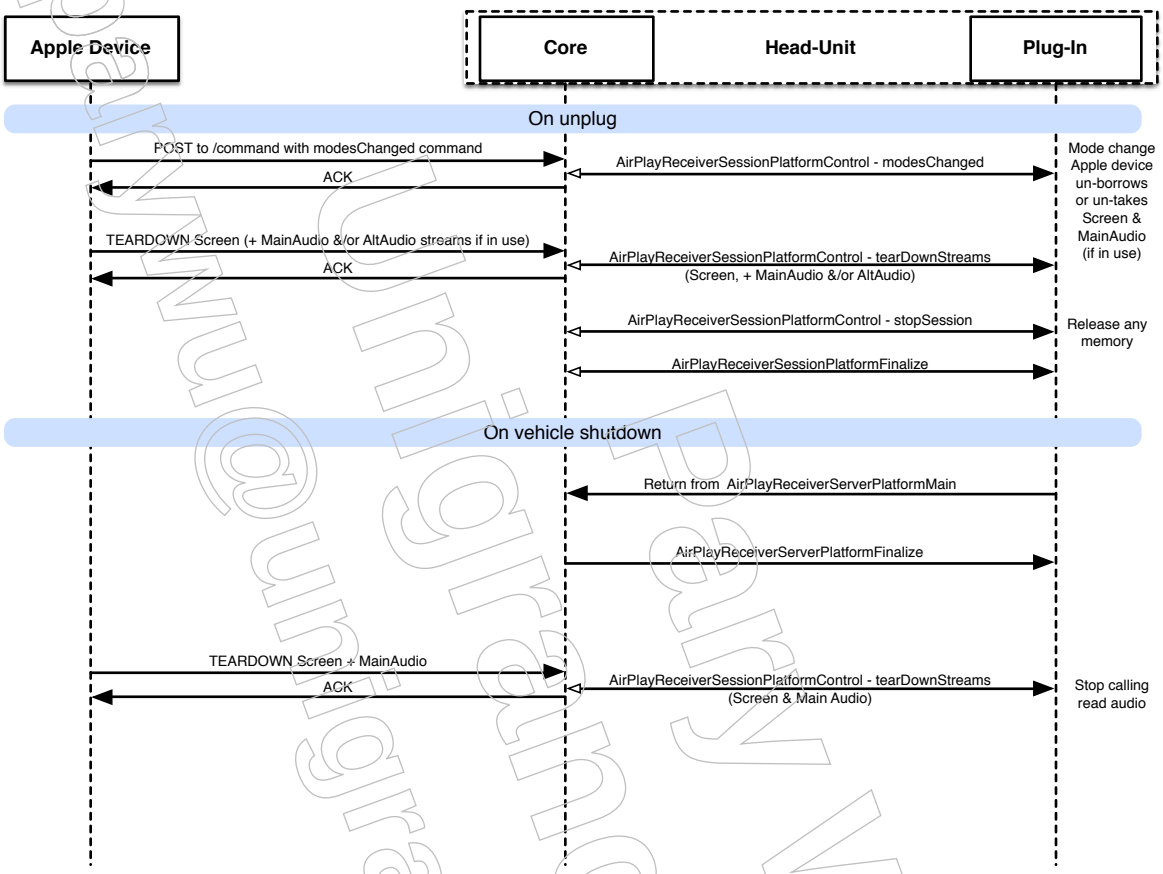
Figure 19-20 Control sequence for returning main audio from Apple device to accessory



Note: Figure 19-20 (page 328) provides an example for borrowing alternate audio and screen to display notifications.

Apple Confidential
Property of Pioneer Electronics (Taiwan) Corporation LTD
www.pioneer.com.tw

Figure 19-21 Control sequence for closing a CarPlay session



19.4 Examples of Modes and Mode Changes

The following examples illustrate sequences of modes and mode changes.

19.4.1 Accessory Showing Vehicle Info

In this initial state, the accessory is showing non-critical vehicle information. No audio is playing.

- Accessory owns Screen
- Accessory owns Main Audio
- Neither is performing speech recognition or speech
- Neither is engaged in a phone call
- Neither is performing turn-by-turn navigation

Tuan Ngia Wu
tuanwu@unigrand.com.tw
Unigrand LTD

The mode dictionary in this state is:

```
{
    resources: {
        {
            resourceID: 1
            entity: 2
        },
        {
            resourceID: 2
            entity: 2
        }
    }
}
```

19.4.2 User Turns on FM Radio

The user now turns on the accessory's FM radio. The accessory sends the following mode change request:

- Screen:
- Transfer Type: Take
- Priority: Nice to Have
- Constraints: Other may take anytime; other may borrow anytime

- Main Audio:
- Transfer Type: Take
- Priority: User-Initiated
- Constraints: Other may take user-initiated; other may borrow anytime

The changeModes command sent from the accessory to the Apple device is:

```
{
    resources: {
        {
            resourceID: 1
            transferType: 1
```

Parry Wu Uniqland LTD parywu@uniqland.com.tw

```
            transferPriority: 100
            takeConstraint: 100
            borrowConstraint: 100
        },
        {
            resourceID: 2
            transferType: 1
            transferPriority: 500
            takeConstraint: 500
            borrowConstraint: 100
        }
    }
    reasonStr: "Turn on FM Radio"
}

response:
{
    status: 0
    modes: {
        resources: {
            {
                resourceID: 1
                entity: 2
            },
            {
                resourceID: 2
                entity: 2
            }
        }
    }
}
```

Pary Wu
Unigrand LTD
pary.wu@unigrand.com.tw

The Apple device evaluates this request and updates the current mode that it sends to the accessory as follows:

- Accessory has Screen
- Accessory has Main Audio

- Neither is performing speech recognition or speech
- Neither is engaged in a phone call
- Neither is performing turn-by-turn navigation

No modesChanged command is sent from the Apple device to the accessory (the mode didn't change).

19.4.3 Case 1: User Switches to the Apple device Music

The user now presses the button to source audio from the Apple device instead of the accessory's FM radio. The Apple device claims ownership of the screen and main audio and the new state reported to the accessory is:

- Apple device has Screen
- Apple device has Main Audio
- Neither is performing speech recognition or speech
- Neither is engaged in a phone call
- Neither is performing turn-by-turn navigation

The modesChanged command sent from the Apple device to the accessory is:

```
{
    params: {
        resources: {
            {
                resourceID: 1
                entity: 1
            },
            {
                resourceID: 2
                entity: 1
            }
        }
    }
}
```

Pary Wu and Unigrand LTD
pary@unigrand.com.tw

19.4.4 Case 2: User Answers an Incoming Phone Call

In this alternate scenario, the user is listening to the accessory's FM radio when a phone call comes in. The Apple device will first borrow the screen and main audio to play the ring tone and show caller ID.

The mode change communicated to the accessory will be:

- Apple device has Screen
- Apple device has Main Audio
- Neither is performing speech recognition or speech
- Neither is engaged in a phone call
- Neither is performing turn-by-turn navigation

The modeChanged command sent from Apple device to the accessory is:

```
{
    params: {
        resources: {
            {
                resourceID: 1
                entity: 1
            },
            {
                resourceID: 2
                entity: 1
            }
        }
    }
}
```

Assuming the user chooses to answer the call, the mode transitions to an "in a phone call" mode. The mode change communicated to the accessory will be:

- Apple device has Screen
- Apple device has Main Audio
- Apple device is doing speech
- Apple device is engaged in a phone call

Party Wu
Unigrand LTD
party@unigrand.com.tw

- Neither is performing turn-by-turn navigation

The modesChanged command sent from the Apple device to the accessory is:

```
{
    params: {
        resources: {
            {
                resourceID: 1
                entity: 1
            },
            {
                resourceID: 2
                entity: 1
            }
        }
        appStates: {
            {
                appStateID: 1
                entity: 1
                speechMode: 1
            }
            {
                appStateID: 2
                entity: 1
            }
        }
    }
}
```

When the phone call ends, the Apple device returns the screen and main audio to the accessory and communicates the following mode change:

- Accessory has Screen
- Accessory has Main Audio
- Neither is performing speech recognition or speech

Parry Wu
Unigrand LTD
parrywu@unigrand.com.tw

- Neither is engaged in a phone call
- Neither is performing turn-by-turn navigation

The modesChanged command sent from the Apple device to the accessory is:

```
{
    params: {
        resources: {
            {
                resourceID: 1
                entity: 2
            },
            {
                resourceID: 2
                entity: 2
            }
        }
    }
}
```

The accessory sees that it has resumed ownership of the screen and main audio, and resumes playing the FM radio in response.

19.5 Test Procedures

Test procedures for self certification are available for accessory manufacturers with approved product plans.

Pary Wu
Uniqland LTD
paryWu@uniqland.com.tw

20. Cases

Accessories that substantially enclose Apple devices must comply with the requirements stated in this chapter unless the accessory supports other features in this specification whose requirements conflict with the requirements in this chapter. For example, the requirements listed in Apple Lightning Connector (page 114) may override some requirements listed in this chapter.

If the accessory has multiple user-detachable components that substantially enclose the Apple device, the requirements and/or overrides must be applied to each component separately.

20.1 Product Design

A well-designed case will securely house an Apple device while not interfering with the device's operation. Significant factors in mechanical design include access to the device's sensors, controls, and connectors.

20.1.1 Device Layouts and Dimensions for Apple Devices

Cases must be designed to accommodate the full range of Apple device sizes within each product's dimensional variation. Dimensional drawings with tolerances for all Apple devices can be found in Device Dimensional Drawings (page 636).

20.1.2 Access to Controls

The case must readily permit the user to access and operate the device's mechanical controls. This includes volume and ring/silent controls, sleep/wake control, and the home button.

20.1.3 Access to the Headphone Jack and 30-pin or Lightning Connector

The case must provide ready access to the headphone jack. The headphone jack opening must be at least 6.0 mm in diameter and at most 14.0 mm deep. At least 6.5 mm in diameter and at most 10.0 mm deep is recommended for best compatibility with a range of headphones.

The case must also provide unobstructed access to either the 30-pin connector or the Lightning connector.

Paryvu Chen (paryvu@uniqueidentity.com.cn) Unique Identity LLC

If the case is for an Apple device with the Lightning connector, the opening must be at least 12.05 mm by 6.30 mm with full radii rounded edges. 13.65 mm by 6.85 mm is recommended for best compatibility with a range of cables and docks.

In addition, the headphone jack and 30-pin or Lightning connector openings must be designed with enough margin to compensate for shifting or dimensional changes of the case material.

20.1.4 Device Protection

Cases must protect the Apple device from a 1 m drop onto a hard paved surface in any device orientation.

Specifically, exposed glass on the Apple device must not come within 1 mm of a flat surface, such as a table or floor, in any orientation when the case is attached. This may be achieved by either covering the exposed glass or creating features around it that will space the exposed glass at least 1 mm away from the flat surface.

20.1.5 Cover Glass Contact

Cases that claim compatibility with iPhone 6 or iPhone 6 Plus should not contact the cover glass as defined in their dimensional drawings. See iPhone 6 (page 638) and iPhone 6 Plus (page 639).

20.1.6 Dock Compatibility

For compatibility with docks, the distance from bottom of the Apple device to the outside of a case should not exceed 1.8 mm.

20.2 Acoustics

The case must not impair or degrade the acoustical performance of an Apple device.

20.2.1 Speaker and Microphone Openings

When Apple devices have speakers or microphones, their locations may vary from model to model. Refer to the dimensional drawings for various Apple devices cited in Device Layouts and Dimensions for Apple Devices (page 336). The case must not obstruct the speaker or microphone ports.

20.2.2 Speaker to Microphone Coupling

The case must not facilitate the conduction of sound from the speaker to any microphone. Such sound conduction can cause echoing in phone calls.

Tommy Wu (tommy@unigrand.com) Unigrand Ltd. Unigrand.com.tw

20.2.3 Call Quality

The case must not impair or degrade the user's experience making and receiving both audio calls over a cellular network or video calls using Apple's FaceTime software. User testing must be conducted in handset, speakerphone, and headset modes of operation, to confirm that the case does not change the loudness or frequency response of the speakers or microphones. In addition, the user must not be able to detect any sound distortion resulting from enclosing the Apple device in the case.

20.3 Sensors

Some Apple devices contain several sensors, including an ambient light sensor, a magnetic compass, a proximity sensor, an accelerometer, and a three-axis gyroscope. Cases must be designed so they do not interfere with the operation of these sensors.

20.3.1 Ambient Light and Proximity Sensor Interference

The ambient light sensor and proximity sensor locations for various Apple devices are shown in the dimensional drawings cited in Device Layouts and Dimensions for Apple Devices (page 336). Some of the dimensional drawings specify a recommended keep-out area around these sensors. No material must be allowed to cover these sensors or their keep-out areas, this includes films and privacy screens. Cases that allow the Apple device to slide around must not obstruct any sensors.

20.3.2 Magnetic Interference

Apple recommends avoiding the use of magnets and metal components in cases.

Cases for Apple devices must not affect the device's built-in magnetic compass (if present).

Additionally, the iPhone 6 Plus has an autofocus rear camera equipped with optical image stabilization that can be affected by magnets and metal components in cases and rear camera accessories. Cases and rear camera accessories that claim compatibility with iPhone 6 Plus must not affect operation of the autofocus rear camera.

20.3.3 Touch ID Sensor

Cases for Apple devices must not inhibit the use of the device's Touch ID sensor (if present). See the device's dimensional drawing for the keep-out area.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

20.4 Camera

This section applies to all Apple devices that are equipped with a camera (with or without a flash). The field of view (FOV) of the camera and the illumination provided by the flash is designed for each Apple product. It is imperative that manufacturers consult technical specifications released for each product and do not assume these parameters are shared between products.

Images from the camera may be affected by the geometry, color and surface finish of the camera opening in the case.

20.4.1 Geometry

The camera lens field of view must not be blocked. Making the case opening around the camera too small may block the field of view of the lens. This may cause vignetting in the image, where the corners of the image are darker than the center. Blocking marginal rays just outside the field of view of the lens may also reduce the sharpness of the image. See Device Layouts and Dimensions for Apple Devices (page 336) for the detailed mechanical keep-out.

The case opening must not be designed in a way that directs stray light into the camera. If the opening is too narrow or too steep, it may reflect light into the camera, washing out the image or adding an unwanted color cast. Adding a chamfer to the camera window and making the window larger can help to direct stray light away from the camera. Additionally, where the product is equipped with a flash, a narrow or steep opening may reflect light from the case opening back into the camera or scene. This may cause the image to appear washed out or contain unwanted artifacts. A wider opening and chamfer will help direct light reflected from the case edges away from the scene and camera.

20.4.2 Color

Note: A matte black material or coating around the camera hole opening is recommended.

Any light reflected from the case may pick up the color of the case. Black material or black coating can help avoid color bleeding into the camera from an external light source or the flash. The darker the color, the less light may be reflected from the source into the camera.

20.4.3 Surface Finish

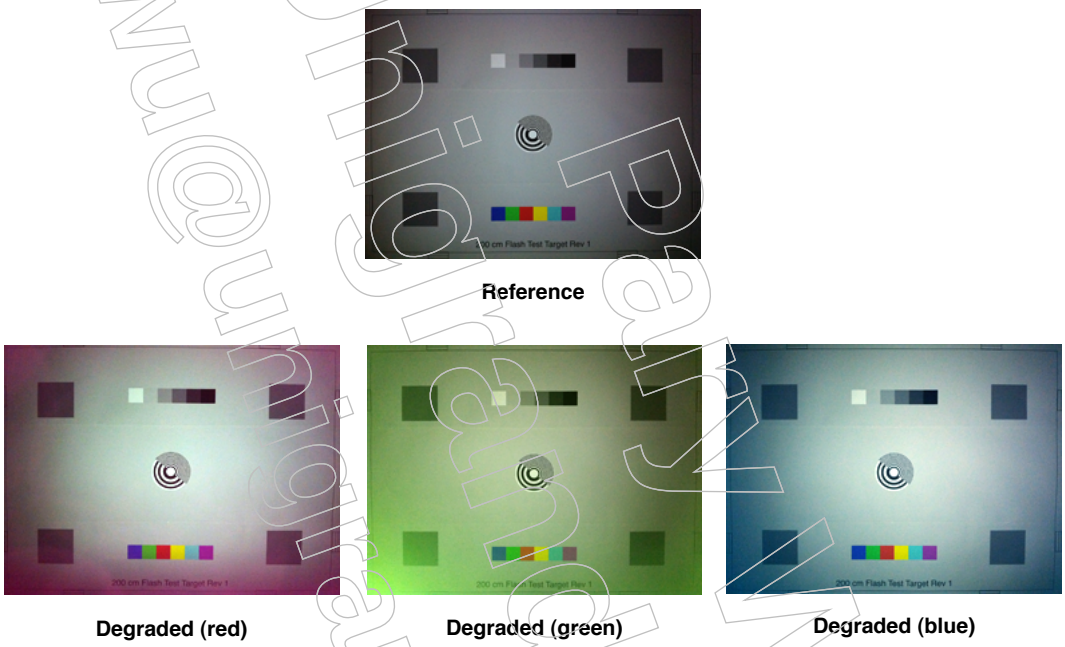
If the cone opening is < 120°, then a semi-gloss finish around the camera opening is recommended. The flash is a strong source of light and reflections from the camera case opening edge should be managed so they do not reflect back into the camera or the scene. Semi-gloss material can direct light away from the camera if

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

combined with a > 95° angle. If the angle is too narrow a high gloss material may reflect light back into the camera. Matte or diffuse materials scatter light in all directions and may increase the likelihood that light from the flash or strong sources in the scene is reflected into the camera.

20.4.4 Image Degradation Examples

Figure 20-1 Image degradation by color shifting



20.4.4.1 Contrast Decrease

Figure 20-2 Image degradation by decrease of contrast

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

20.4.4.2 Image Blocking

Figure 20-3 Image degradation by blocking

20.4.4.3 Flash Interference

Figure 20-4 Image degradation by flash interference

20.5 Reliability

Cases for Apple devices must be tested to verify that they will withstand long-term use under typical use conditions, and that they do not impair or degrade the functionality of the device, damage it or its immediate surroundings, or adversely affect the user.

20.5.1 Device Insertion and Removal

The case must hold the Apple device securely while permitting its easy insertion and removal. The case and the enclosed device must not be damaged by the repeated insertion and removal of the device from the case under conditions representative of long-term use in a variety of environments.

20.5.2 Colorfastness

Any dyes, inks, or coatings in or on the case must not bleed color onto either the device or its user, particularly while the case is in contact with common substances such as water or sunscreen.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

20.6 Environmental

Cases for Apple devices must comply with applicable environmental regulations in the regions in which such cases are to be sold, and any applicable substance or material restrictions, including applicable restrictions on the following substances:

- Lead (Pb)
- Cadmium (Cd)
- Mercury (Hg)
- Hexavalent Chromium (Cr(VI))
- Hexavalent Chromium (Cr(VI)) in leather
- Nickel (Ni) plating on user-accessible surfaces
- Polybrominated biphenyls (PBB)
- Polybrominated diphenyl ethers (PBDE)
- Azo-based dyes that release the aromatic amines specified in European Directive 2002/61/EC
- Natural rubber latex
- Formaldehyde
- Endangered species of flora and fauna in products or packaging (US Lacey Act)
- Perfluorooctanoic Acid (PFOA)
- Perfluorooctane Sulfonates (PFOS)
- Phthalates

20.7 RF

20.7.1 Materials and Coatings

Cases for Apple devices must not contain materials or coatings that absorb radio frequency energy. Such materials can impair or degrade the performance of cellular communication antennas or GPS, Wi-Fi, or Bluetooth antennas. Examples include (but are not limited to) the following:

- Metals (e.g. steel, aluminum, magnesium, titanium, etc.)
- Plastics with any carbon content
- Plastics with any glass content
- Plastics with metallic plating

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- Metallic paints
- Black paints with high carbon loading
- White paints with high titanium dioxide loading Metallic
- Physical Vapor Deposition (PVD) coatings

20.7.2 Near Field Communication (NFC)

Cases that claim compatibility with NFC enabled Apple devices (such as iPhone 6 and iPhone 6 Plus) must not degrade device NFC transaction performance.

20.8 Touchscreen

The touch interface in an Apple device senses the presence of one or more fingers on its surface. Any material between the surface and the user's hand, even a very thin sheet of plastic, can affect the performance of the touch interface.

20.8.1 Overlay

If a case design requires a touchscreen to be overlaid with another material, the material must be:

- Thinner than 0.3 mm
- Designed so that there are no air gaps between it and the touchscreen surface
- Not be electrically conductive

20.8.2 Edge Swipe Gestures

A case must allow the user to use edge swipe gestures that were introduced in iOS 7.0. These gestures include bringing up Control Center, Notification Center, and swiping back from apps that may use edge swipe gestures (such as the Messages app).

20.9 Test Procedures

20.9.1 Apple Device Compatibility

Case testing procedures vary depending on the Apple device they enclose. Cases must be tested for every Apple device that they claim compatibility with.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

20.9.1.1 iPhone 6

The following tests must be run for cases that claim compatibility with iPhone 6:

- Product Design (page 347) using an iPhone 6.
- RF (OTA) (page 349) using an iPhone 6 for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4GHz Wi-Fi 1, 7, 13
  - 5.0GHz Wi-Fi 36, 60, 100, 120, 165

20.9.1.2 iPhone 6 Plus

The following tests must be run for cases that claim compatibility with iPhone 6 Plus:

- Product Design (page 347) using an iPhone 6 Plus.
- RF (OTA) (page 349) using an iPhone 6 Plus for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4GHz Wi-Fi 1, 7, 13
  - 5.0GHz Wi-Fi 36, 60, 100, 120, 165

20.9.1.3 iPhone 5/iPhone 5s

The following tests must be run for cases that claim compatibility with iPhone 5/iPhone 5s:

- Product Design (page 347) using an iPhone 5s.
- RF (OTA) (page 349) using an iPhone 5s for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4GHz Wi-Fi 1, 7, 13
  - 5.0GHz Wi-Fi 36, 60, 100, 120, 165

It is not possible for a case to claim compatibility with only the iPhone 5 or only the iPhone 5s.

Party Wu
Unigrand LTD
tonywu@unigrand.com.tw

20.9.1.4 iPhone 5c

The following tests must be run for cases that claim compatibility with iPhone 5c:

- Product Design (page 347) using an iPhone 5c.
- RF (OTA) (page 349) using an iPhone 5c for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4GHz Wi-Fi 1, 7, 13
  - 5.0GHz Wi-Fi 36, 60, 100, 120, 165

20.9.1.5 iPad mini/iPad mini 2/iPad mini 3

The following tests must be run for cases that claim compatibility with iPad mini/iPad mini 2/iPad mini 3:

- Product Design (page 347) using an iPad mini 3.
- RF (OTA) (page 349) using an iPad mini 3 for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4GHz Wi-Fi 1, 7, 13
  - 5.0GHz Wi-Fi 36, 60, 100, 120, 165

It is not possible for a case to claim compatibility with only the iPad mini or only the iPad mini 2 or only the iPad mini 3.

20.9.1.6 iPad Air 2

The following tests must be run for cases that claim compatibility with iPad Air 2:

- Product Design (page 347) using an iPad Air 2.
- RF (OTA) (page 349) using an iPad Air 2 for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4GHz Wi-Fi 1, 7, 13
  - 5.0GHz Wi-Fi 36, 60, 100, 120, 165

Payment@unigrand.com.tw Unigrand LTD

20.9.1.7 iPad Air

The following tests must be run for cases that claim compatibility with iPad Air:

- Product Design (page 347) using an iPad Air.
- RF (OTA) (page 349) using an iPad Air for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4GHz Wi-Fi 1, 7, 13
  - 5.0GHz Wi-Fi 36, 60, 100, 120, 165

20.9.1.8 iPad (4th generation)

The following tests must be run for cases that claim compatibility with iPad (4th generation):

- Product Design (page 347) using an iPad (4th generation).
- RF (OTA) (page 349) using an iPad (4th generation) for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - LTE 17
  - 2.4GHz Wi-Fi 1, 7, 13
  - 5.0GHz Wi-Fi 36, 60, 100, 120, 165

20.9.2 Apple Device Configurations

Specific Apple device configurations must be used depending on the test being performed and the device being tested against.

Table 20-1 Required device configurations for accessory case test procedures

Test	Device	Model	Color
Product Design (page 347)	iPhone 6	Any	Any
Product Design (page 347)	iPhone 6 Plus	Any	Any
Product Design (page 347)	iPhone 5s	Any	Any
Product Design (page 347)	iPhone 5c	Any	Any
Product Design (page 347)	iPad Air 2	Any	Any

Party Wu
Unigrand LTD
wu@unigrand.com.tw

Test	Device	Model	Color
Product Design (page 347)	iPad mini 3	Any	Any
Product Design (page 347)	iPad Air	Any	Any
Product Design (page 347)	iPad (4th generation)	Any	Any
RF (OTA) (page 349)	iPhone 6	Unlocked A1586	Any
RF (OTA) (page 349)	iPhone 6 Plus	Unlocked A1524	Any
RF (OTA) (page 349)	iPhone 5s	Unlocked A1453 and Unlocked A1530	Any
RF (OTA) (page 349)	iPhone 5c	Unlocked A1456 and Unlocked A1529	Any
RF (OTA) (page 349)	iPad Air 2	Unlocked A1567	Any
RF (OTA) (page 349)	iPad mini 3	Unlocked A1600 and Unlocked A1601	Any
RF (OTA) (page 349)	iPad Air	Unlocked A1475 and Unlocked A1476	Any
RF (OTA) (page 349)	iPad (4th generation)	Unlocked A1459	Any

20.9.3 Product Design

20.9.3.1 Equipment

- Apple device. See Table 20-1 (page 346).
- Apple Lightning Digital AV Adapter
- Apple 30-pin to USB Cable
- Vernier calipers
- 0.85 mm plastic feeler gauge
- Apple EarPods with Remote and Mic

20.9.3.2 Procedure

1. Insert the Apple device into the case.
2. Ensure that the Apple device completely fits inside the case. The Apple device must not be loose.

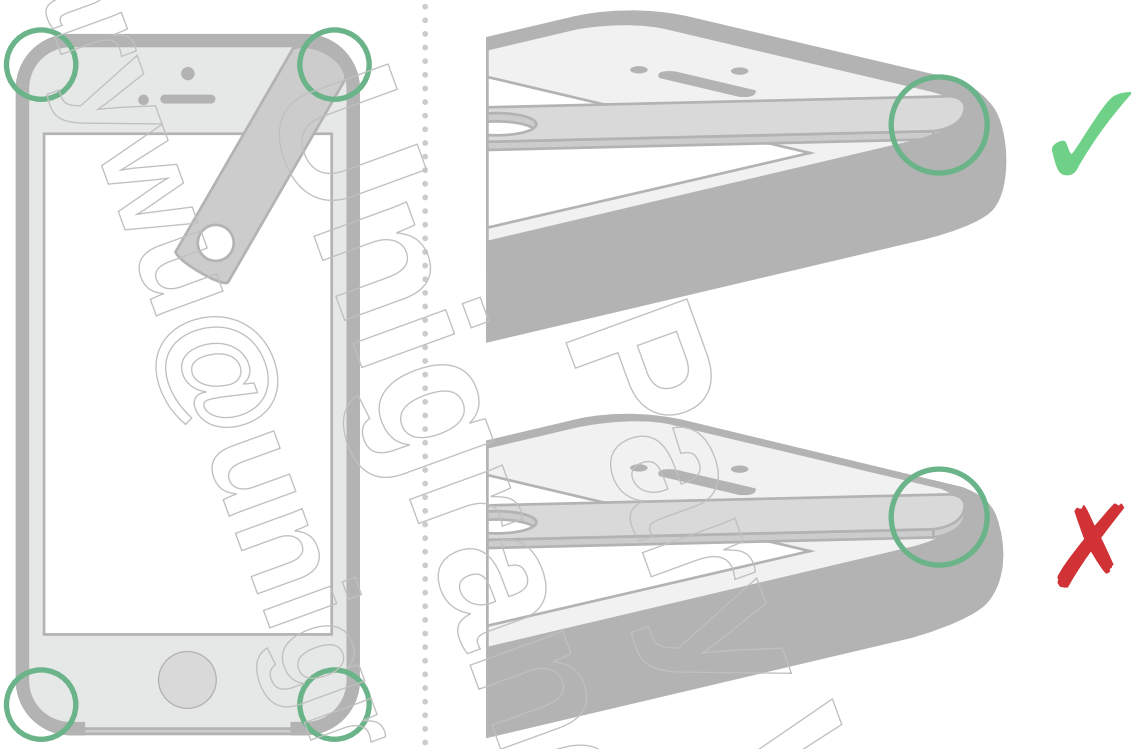
Authorized by Wu
Quinland LTD
quinland.com.tw

3. Ensure that all buttons are accessible.
4. Inspect for button feel. The buttons must not be too hard to press or take a lot of effort to press.
5. For Apple devices that have an Apple Lightning connector:
   a. Insert the Apple Lightning Digital AV Adapter into the Lightning receptacle and ensure that it fits.
   b. Using vernier calipers, measure the Lightning connector opening on the case. Ensure that the opening is measured to be at least 12.05 mm by 6.30 mm.
6. For Apple devices that have an Apple 30-pin connector:
   a. Insert the Apple 30-pin to USB Cable into the 30-pin receptacle and ensure that it fits.
   b. Using vernier calipers, measure the 30-pin connector opening on the case. Ensure that the opening is measured to be at least 22.1 mm by 2.8 mm.
7. Insert Apple EarPods with Remote and Mic into the headphone jack of the Apple device and ensure it fits.
8. Using vernier calipers, measure the headphone jack opening on the case. Ensure that the opening is measured to be at least 6 mm in diameter and no more than 14 mm deep.
9. For iPhone 5s, iPhone 6, iPhone 6 Plus, iPad mini 3, and iPad Air 2, use vernier calipers to ensure that the case is at least 2 mm away from the Touch ID sensor.

Testing Property of Wu Fu Chang and LTD
Testing@wufuchang.com.tw

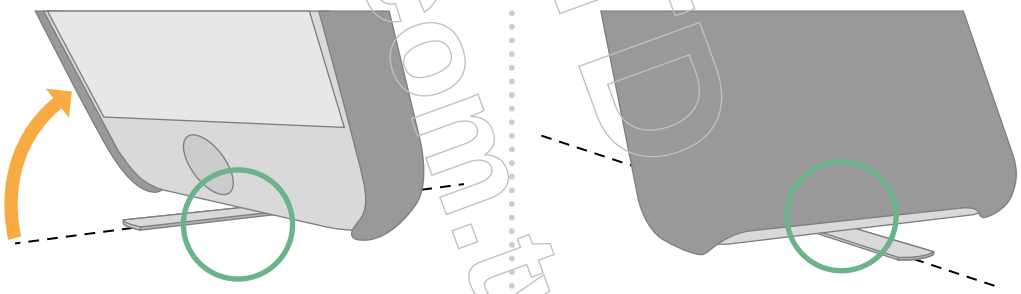
10. Ensure that the case is always proud of the feeler gauge when the gauge is placed at each corner of the Apple device. See Figure 20-5 (page 349).

Figure 20-5 Apple Device Proudness Test



11. Set the Apple device flat on its face (screen facing down).
12. Roll the device towards any side that is not enclosed by the case until the gap between the Apple device's exposed glass and flat surface is smallest.
13. Ensure that the feeler gauge fits into the gap between the Apple device's exposed glass and flat surface.

Figure 20-6 Apple Device Gap Test



20.9.4 RF (OTA)

This section contains test procedures for accessory cases that contain a Lightning connector.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

20.9.4.1 Definitions

- TRP: Total Radiated Power (dBm)
- TIS: Total Isotropic Sensitivity (dBm)
- EIS: Effective Isotropic Sensitivity (dBm)
- EIRP: Effective Isotropic Radiated Power (dBm)
- Electric Permittivity: $\varepsilon = \varepsilon_0 k_e$, where k_e is dielectric constant and ε_0 is permittivity of free space.

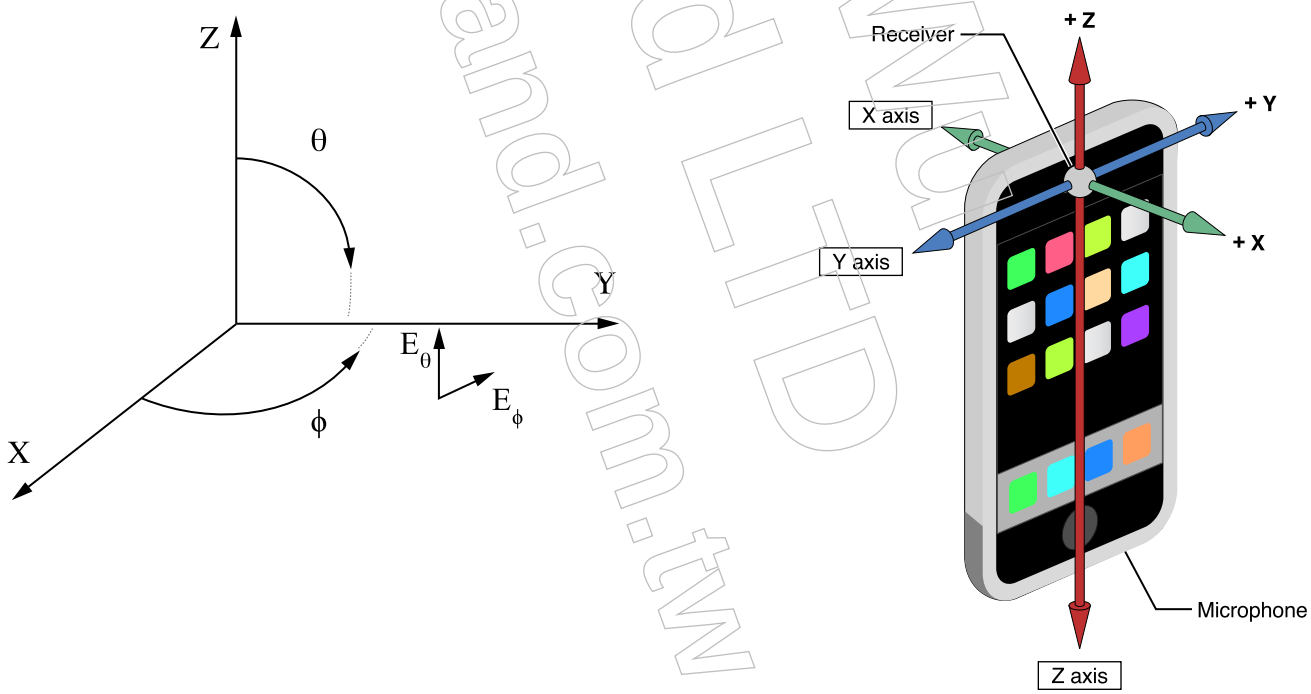
20.9.4.2 Labs

Testing must be performed by CTIA Test Labs (http://www.ctia.org/policy-initiatives/wireless-device-certification/ctia-authorized-test-labs) with the authorization and equipment needed for the following tests:

- EIRP Measurement for cellular (UMTS/LTE) and Wi-Fi (802.11a/b)
- EIS Measurement for cellular (UMTS/LTE)
- TRP for Wi-Fi (802.11a/b)

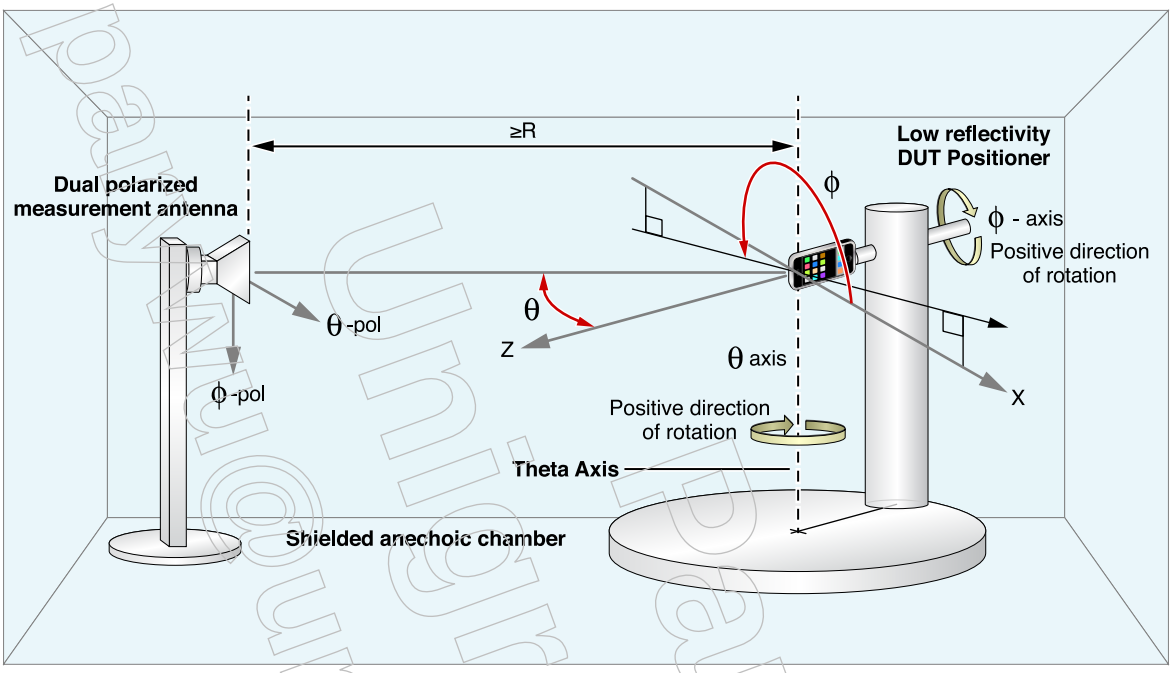
20.9.4.3 Reference Device Setup

Figure 20-7 Apple Device Setup



Preliminary Unofficial Version
Unofficial Version
Preliminary
Unofficial
Version
Preliminary
Unofficial
Version
Preliminary
Unofficial

Figure 20-8 Generic Setup for TRP/TIS(Free Space)

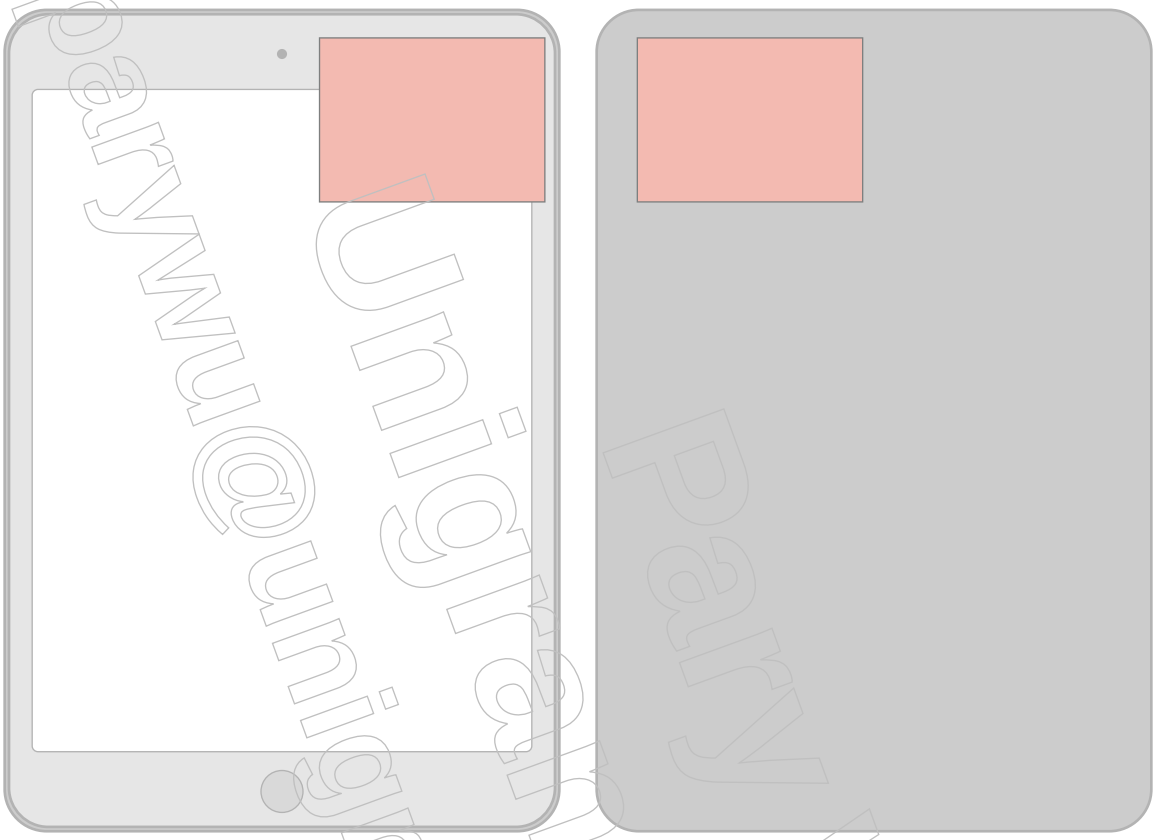


20.9.4.3.1 iPad

If the reference Apple device is an iPad, secure 3 layers of Post-It notes or equivalent on the top right front and top left back. The paper must remain in place for all testing, both for reference data collection and during testing of iPad cases. See Figure 20-9 (page 352).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 20-9 iPad with Post-it Notes



Parywu@unigrandtech.com Unigrand IED Parywu

20.9.4.3.2 TRP Uncertainty

TRP measurements must be performed under Free Space conditions. Multiple reference Apple devices may be used under the following conditions:

- TRP values of each Apple device must not vary more than 0.5 dB.
- Uncertainty of the same Apple device must be less than 0.3 dB from each measurement.
- Uncertainty of the Apple device and accessory must be less than 0.5 dB.
- If the Apple device is changed during the measurement, the uncertainty must be less than 0.5 dB.

20.9.4.3.3 TIS Uncertainty

TIS measurements must be performed under Free Space conditions. Multiple reference Apple devices may be used under the following conditions:

- TIS values of each Apple device must not vary more than 1.0 dB.
- Uncertainty of the same Apple device must be less than 1.0 dB from each measurement.
- Uncertainty of the Apple device and accessory must be less than 1.0dB.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- If the Apple device is changed during the measurement, the uncertainty must be less than 1.0 dB.

20.9.4.3.4 Peak Position

3D TRP/TIS measurements must be taken from each reference Apple device in each available chamber to determine peak position.

20.9.4.3.5 EIRP/EIS Measurement Matrix

Table 20-2 EIRP/EIS Measurement Matrix for case OTA testing procedure

Band	Channel	EIRP with Case	EIS with Case	Intermediate Channel
Band IV	L	Y	N	N
	M	Y	N	N
	H	Y	N	N
Band V	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
Band VIII	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
Band II	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
Band I	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
LTE 17	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N

Payment Digital LTD
Payment@unionpaydigital.com.tw

Band	Channel	EIRP with Case	EIS with Case	Intermediate Channel
LTE 7	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
TD-LTE 40	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N

20.9.4.3.6 TRP/TIS Measurement Matrix

Note: This measurement must be tested using the latest CTIA spec. No intermediate test is required.

- UMTS Band I/II/IV(TRP only)/V/VIII Mid channel TRP/TIS
- LTE 17/7 Mid channel TRP/TIS
- TD-LTE B40 Mid channel TRP/TIS

Table 20-3 TRP/TIS Measurement Matrix for case OTA testing procedure

Band	Channel	TRP	TIS	Intermediate Channel
Band IV	L	N	N	N
	M	Y	N	N
	H	N	N	N
Band V	L	N	N	N
	M	Y	Y	N
	H	N	N	N
Band VIII	L	N	N	N
	M	Y	Y	N
	H	N	N	N

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Band	Channel	TRP	TIS	Intermediate Channel
Band II	L	N	N	N
	M	Y	Y	N
	H	N	N	N
Band I	L	N	N	N
	M	Y	Y	N
	H	N	N	N
LTE 17	L	N	N	N
	M	Y	Y	N
	H	N	N	N
LTE 7	L	N	N	N
	M	Y	Y	N
	H	N	N	N
TD-LTE 40	L	N	N	N
	M	Y	Y	N
	H	N	N	N

20.9.4.3.7 iOS Configuration

- Cellular Testing:
  - Cellular must be turned on (Airplane Mode off)
  - Wi-Fi must be turned off
  - Bluetooth must be turned off
  - Screen must be turned off
- Wi-Fi Testing:
  - Cellular must be turned off (Airplane Mode on)
  - Bluetooth must be turned off
  - Wi-Fi must be turned on
  - Location Services must be turned off

Party Wu Unionland LTD partywu@unionland.com.tw

- Auto-Lock must be set to Never
- Screen must be turned on

- The Apple device must be fully charged when starting EIRP measurement.
- EIRP measurements must not be performed when the Apple device's battery has less than 25% remaining.
- EIS measurements are not affected by battery level, but Apple highly recommends that EIS measurements be performed on a fully charged battery.
- See Peak Position (page 353) for additional setup information.

20.9.4.3.8 Apple Device Models

Specific Apple device models must be used to test certain OTA bands.

Table 20-4 Required Apple device models for testing of specific OTA bands

Device	Type A	Type B
iPhone 6	Unlocked A1586	Unlocked A1586
iPhone 6 Plus	Unlocked A1524	Unlocked A1524
iPhone 5s	Unlocked A1453	Unlocked A1530
iPhone 5c	Unlocked A1456	Unlocked A1529
iPad Air 2	Unlocked A1567	Unlocked A1567
iPad mini 3	Unlocked A1600	Unlocked A1601
iPad Air	Unlocked A1475	Unlocked A1476

- Cellular for LTE B7, TD-LTE B40 must be tested using Type B
- Cellular for UMTS I/II/IV/V/VIII, LTE B17 must be tested using Type A
- Wi-Fi: Either Type A or Type B must be used

20.9.4.4 Setup

Note: If the case has multiple configurations (e.g. 'open' and 'closed', and/or 'off' and 'on' modes), all modes must be tested and passed.

Confidential Property of Wu Equipment Taiwan Co., LTD

Case:

- The case and Apple device must be mounted using a material with a very low dielectric constant. Refer to *CTIA 3.1 Appendix A.1* for more information on free space mounts that may be used.
- The case and Apple device must be mounted in the same position (with respect to the measuring antenna) as the corresponding reference Apple device free space setup.
- If the case has 'on' and 'off' operating modes, it must be 'off' or not operating during TRP testing.
- If the case has 'on' and 'off' operating modes, it must be 'on' or operating during TIS testing.
- If the case can provide power to the Apple device during a phone call, the Apple device must be charging during TIS testing

Note: The link/communication antenna must be at least 3-5 feet away from the case and Apple device.

20.9.4.5 Procedure

Apple device setup:

1. Turn on Apple device
2. Put Apple device into Airplane Mode
3. Turn off the Apple device
4. Attach case to Apple device
5. Take Apple device out of Airplane Mode
6. Wait for Apple device to search for and attach to network
7. Connect call and wait > 1 minute
8. Start testing

20.9.4.5.1 TRP/EIRP for All Bands

- Cellular EIRP measurements must be taken at the Apple device's peak position (see Peak Position (page 353)) in free space.
- Wi-Fi EIRP Short Sweep measurements must be taken within +/- 15deg the Apple device's peak position in free space.
- See TRP Uncertainty (page 352) for TRP uncertainty requirements
- Cellular EIRP measurements must be taken at Low/Mid/High channels for each available band.
- Wi-Fi peak EIRP measurements must be taken at channels 1,7,13, 36, 60, 100, 120, and 165.

Gary Wu
Unigrand LTD
garywu@unigrand.com.tw

- 3D TRP measurements must be taken only when EIRP does not meet specification

20.9.4.5.2 TIS/EIS for All Bands

- EIS measurements must be taken at the Apple device's peak position (see Peak Position (page 353)) in free space.
- See TIS Uncertainty (page 352) for TIS uncertainty requirements
- EIS measurements must be taken at Low/Mid/High channels for each available band.
- 3D TIS measurements must be taken only when EIS does not meet specification.
- When testing WCDMA, place the Apple device at peak EIS position and decrease the output power of the base station emulator until 1.2% BER is reached.
- When testing LTE, place the Apple device at peak EIS position and decrease the output power of the base station emulator until 5% BLER is reached.

20.9.4.5.3 TRP/EIRP for UMTS Band I/II/IV/V/VIII

EIRP measurements must be taken every 15° in the Theta and Phi axis to derive the TRP value. All equipment settings must be set according to CTIA spec except for the following:

- FSP (Spectrum Analyzer)
  - Resolution BW = 30KHz
  - Video BW = 10MHz
  - Sweep Time = 100ms
  - Points per trace = 501
  - Averaging Factor = 1 point.
  - Filter Trace Average Count = 2
  - Filter Selection = Integrated Channel Power
  - Detector Selection = RMS Detector
- Power meter can be used per CTIA spec
- TS-RSP (RF Switch Setting)
  - Pause after switching = 0ms.

20.9.4.5.4 TRP/EIRP for FD-LTE Band 17/7 (10MHz/QPSK modulation only)

EIRP measurements must be taken every 15° in the Theta and Phi axis to derive the TRP value. All equipment settings must be set according to CTIA spec except for the following:

For Private Party Wu
Issued to Compliance and LTD
Integrand.com.tw

- Power meter/Spectrum analyzer can be used per CTIA spec
- UL RB allocation of 12
- UL RB start 0 for low channel, 20 for mid channel, 38 for high channel
- TS-RSP (RF Switch Setting)
  - Pause after switching = 0ms.

20.9.4.5.5 TRP/EIRP for TD-LTE Band B40 (20MHz BW,QPSK)

EIRP measurements must be taken every 15° in the Theta and Phi axis to derive the TRP value. All equipment settings must be set according to CTIA spec except for the following:

- Power meter/Spectrum analyzer (FSQ/FSU)
  - Resolution BW = 50Mhz Video BW = 30 MHz
  - Span = 0Hz
  - Sweep Time = 1.7ms
  - Points per trace = 501
  - Trace = Average
  - Detector Selection = RMS Detector
- UL RB allocation of 18
- UL RB start 0 for low channel, 41 for mid channel, 82 for high channel

20.9.4.5.6 TIS/EIS for UMTS Band I/II/IV/V/VIII (all bands that are supported by DUT)

Note: Expected time for the TIS/EIS test is 3-5 hours/band

All equipment settings must be set according to CTIA spec except for the following:

- Sensitivity Setting
  - Sensitivity Search Algorithm = "Fast Probing" (RSCP Replacement is not allowed)
  - Max/Min Frames = 82
  - Power Settling Time = 0ms
  - Output Channel Power Settling time = 0ms
  - Initial Power Setting = -55 ~ -60dBm
  - TPC Setting = "all 1"

Pary Wu
Usignal Int'l LTD
paywu@usignal.com.tw

- Probing Search Frames = 5
- Initial Probe retries = 3
- Probe Bias = 1
- Mobile PCL Settle Time = 0ms
- Base Settle Time = 0ms
- Max Step = 2dB
- Fine Step = 1dB
- Minimum Step = 0.5dB

- TS-RSP (RF Switch Setting)
  - Pause after switching = 0ms

20.9.4.5.7 TIS/EIS for FD-LTE Band 17/7 (10MHz BW/QPSK modulation only)

Note: Expected time for the TIS/EIS test is 3-5 hours/band

All equipment settings must be set according to CTIA spec except for the following:

- Sensitivity Setting
  - Max/Min Frames = 2000
  - Initial Power Setting = -55 ~ -60dBm
  - Power Setting = "all 1"
  - Max Step = 2dB
  - Fine Step = 1dB
  - Minimum Step = 0.5dB
  - DL RB allocation of 50
- TS-RSP (RF Switch Setting)
  - Pause after switching = 0ms

20.9.4.5.8 TIS for TD-LTE Band 40 (20MHz BW/QPSK modulation only)

Note: Expected time for this test is 5 hours/band

Easy Wu @ unigrand.com.tw
Unigrand LTD Party Wu

All equipment settings must be set according to CTIA spec except for the following:

- Sensitivity Setting
  - Max/Min Frames = 2000
  - Initial Power Setting = -55 ~ -60dBm
  - Power Setting = "all 1"
  - Max Step = 2dB
  - Fine Step = 1dB
  - Minimum Step = 0.5dB
  - DL RB allocation of 50
- TS-RSP (RF Switch Setting)
  - Pause after switching = 0ms

20.9.4.5.9 TRP for Wi-Fi

EIRP measurements must be taken every 15° in the Theta and Phi axis to derive the TRP value. Use the following test equipment settings for 802.11b and 802.11a:

- Data rate:
  - 802.11b: 11Mbps
  - 802.11a: 6Mbps
- Packet size: 500 bytes
- Interval between packets: 19ms
- Preamble: Long (if applicable)

Connect the Apple device to the test network SSID:

- Under Settings > Wi-Fi, search for test network SSID and manually connect
- Once associated return Apple device to home screen

Payu Wu Unigrand LTD payu@unigrand.com.tw

21. Device Authentication

The Device Authentication feature is used to verify the authenticity and/or identity of an Apple device.

21.1 Device Authentication Requirements

All accessories that support the Device Authentication feature via iAP2 must send or receive the following iAP2 control session message(s):

- RequestDeviceAuthenticationCertificate (page 594)
- DeviceAuthenticationCertificate (page 595)
- RequestDeviceAuthenticationChallengeResponse (page 595)
- DeviceAuthenticationResponse (page 595)
- DeviceAuthenticationSucceeded (page 596)
- DeviceAuthenticationFailed (page 596)

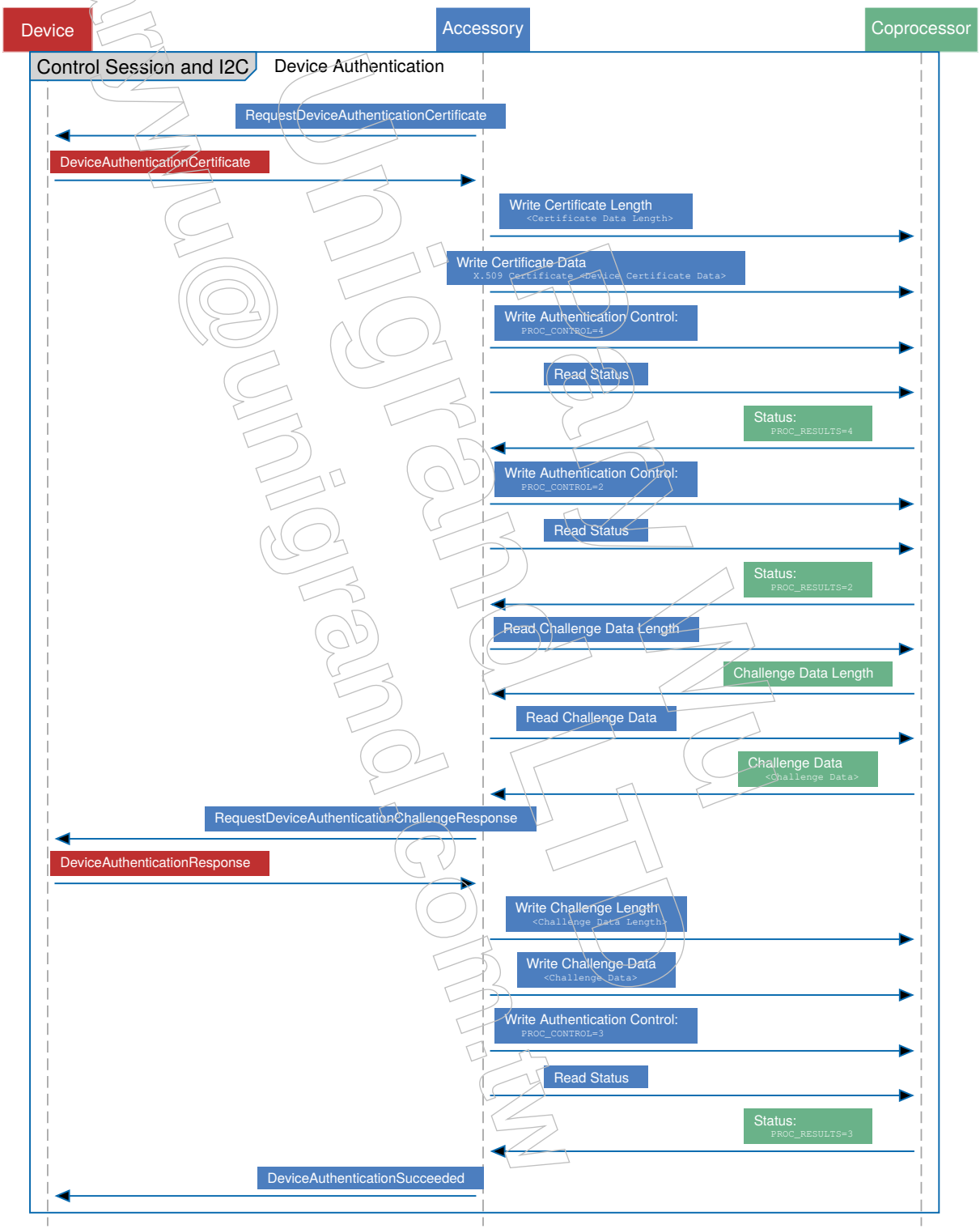
21.2 Device Authentication Usage

The Device Authentication process is the same as that specified in Accessory Authentication (page 201) except that the roles are reversed, and the accessory must terminate the iAP2 connection if the device fails to authenticate. For example, the accessory may continue to communicate with the device but with restricted (non-iAP2) functionality. Accessory Authentication (page 201) and Accessory Identification (page 205) must take place before Device Authentication may occur.

Third Party Wireless Standard Inc. Unauthorized Copy

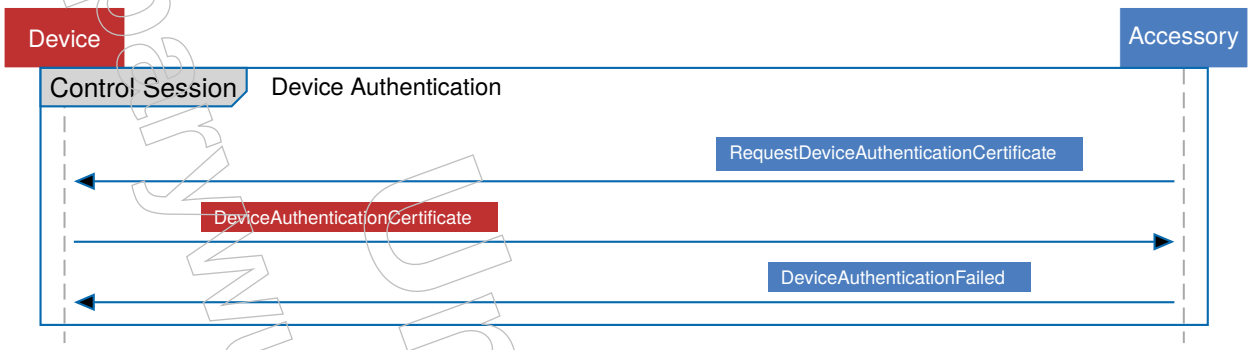
21.3 Device Authentication Examples

21.3.1 Typical Device Authentication

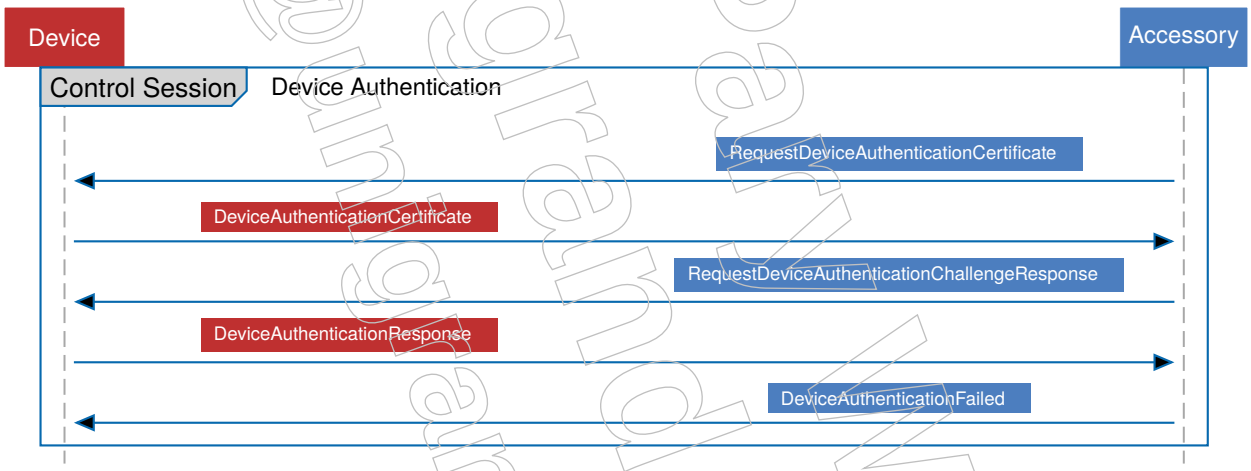


Preliminary Draft Unigrand Confidential Unigrand

21.3.2 Device Authentication Failure Due To Invalid Certificate



21.3.3 Device Authentication Failure Due To Invalid Response



Paul Wu LTD
paulwu@inbrand.com.tw

22. Device Notifications

An accessory that supports the Device Notifications feature can register for messages from the Apple device that convey various parts of device state. Device information that can be provided includes:

- Human-readable name as set by the user (i.e. "John Doe's iPhone")
- Current UI language
- Current date/time information

22.1 Device Notifications Requirements

All accessories that support the Device Notifications feature via iAP2 may also send or receive the following iAP2 control session message(s):

DeviceInformationUpdate (page 596)

DeviceLanguageUpdate (page 597)

DeviceTimeUpdate (page 597)

All messages related to this feature are optional.

22.2 Device Notifications Usage

The Apple device will automatically send update messages if the accessory claims to receive them during Accessory Identification. After successful identification, the Apple device will send one update message to inform the accessory of initial state, and will send additional messages if that state changes. Accessories must be prepared to receive multiple update messages while connected to the device.

After receiving DeviceTimeUpdate (page 597), the accessory must maintain its own clock. Subsequent notifications will be sent when there is a significant change, e.g., a device time zone or daylight savings time change.

Party Wu
Digital Island LTD
parwu@digitalisland.com

23. Digital Audio

Apple strongly recommends the use of digital audio paths to and from accessories. USB Host Mode audio is the recommended approach; the USB Device Mode audio feature is being placed into a maintenance mode.

When an Apple device is in USB Host Mode, it can interact with accessories that are compliant with either the USB Audio 1.0 or 2.0 specification. Version 2.0 is recommended, even for USB Full Speed accessories. USB High Speed accessories must not implement the USB Audio 1.0 specification.

When an Apple device is in USB Device Mode, it can appear as a USB Audio 1.0 device for the purpose of streaming audio to the accessory. The audio stream contains stereo PCM data and is synchronized using USB 1 ms Start-Of-Frame (SOF) packets. USB Device Mode audio does not support streaming audio from an accessory to the device.

23.1 Digital Audio Requirements

Note: All accessories must comply with TDMA noise requirements. See TDMA Noise (page 53).

23.1.1 Readiness for Audio Streaming

Failure to render all audio output from an Apple device once it has routed outgoing audio to an accessory is grounds for failure to pass self certification. Accessories must not establish an audio transport connection (in either direction) to an Apple device until they are completely ready to provide or receive audio streaming data.

23.1.2 Multiple Audio Connections

All accessories must maintain a maximum of one audio transport connection with a device at any given time. This requirement covers the following audio transport connections:

- USB Device Mode audio
- USB Host Mode audio
- Bluetooth A2DP audio
- Headphone Jack audio

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- CarPlay audio

If an accessory is capable of streaming audio to or from an Apple device via more than one audio transport connection, it must shut down the active audio transport connection before starting the next one. The presence or absence of audio on an audio transport connection has no bearing on the connection state. Shutdown and startup actions for each audio transport connection are specified in the following table:

Table 23-1 Audio Transport Connection Actions

Audio Transport Connection	Startup Action	Shutdown Action
USB Device Mode	StartUSBDeviceModeAudio (page 620)	StopUSBDeviceModeAudio (page 621)
USB Host Mode	USB attach	USB detach
Bluetooth A2DP	Bluetooth connect	Bluetooth disconnect
Headphone Jack	Jack insertion	Jack removal

If an accessory is capable of establishing a Bluetooth A2DP audio transport connection along with any other audio transport connection to an Apple device, the accessory must satisfy all requirements stated in Bluetooth (page 216).

23.1.3 Audio Input Source Switching

Accessories that can accept audio input from sources other than Apple devices must select the Apple device input source as soon as the accessory establishes an audio transport connection to the device (see Table 23-1 (page 367)), whether as a result of direct user action or otherwise.

Conversely, if the user takes direct action to switch the accessory's audio input source away from the Apple device, the accessory must shut down the active audio transport connection (see Table 23-1 (page 367)) before switching to the alternate source.

23.1.4 Copy Protection of Digital Audio Output

Any accessory that outputs digital audio obtained from an Apple device must implement copy protection in its output stream; for example, by setting the output's Serial Copy Management System (SCMS) bits to 10.

23.1.5 USB Host Mode Audio

An accessory implementing USB Host Mode Audio must support the following USB features:

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- 16-bit linear PCM.
- Support for both 44100 and 48000 Hz sampling rates.
- If the input and output endpoints are both asynchronous, both endpoints must use the same clock source (i.e. be sample locked) and use implicit feedback.
- The maximum packet size descriptor must comply with requirements for USB Audio packet sizes per Section 2.3.1.1 of the USB Audio Data Formats 2.0 specification. Packets must never vary by more than ±1 sample frame per packet.
- The maximum packet size for each endpoint must comply with the calculation method specified by Technical Note TN2274 in the Mac Developer Library online documentation.
- When streaming, accessories must always observe the bandwidth restrictions for the current sample rate and format settings.
- Volume Control Feature Units must use a Status Interrupt Pipe, per Section 3.7.1.2 of the USB Audio 1.0 specification or Section 6 of the USB Audio 2.0 specification.
- A USB Host Mode accessory that wants to give iOS access to its volume control capabilities must implement either a USB Master Volume Control Feature Unit and/or USB Individual Volume Control Feature Units on all the relevant streaming audio endpoints. If iOS wants to set the accessory's overall input/output volume and no master control exists, it will aggregate the individual controls and set them in tandem.
- All USB audio interfaces must include zero-bandwidth alternate settings.

The following USB features are recommended for accessories implementing USB Host Mode Audio, but are not required:

- Synchronous audio endpoints, particularly for input-only or output-only accessories.
- 24-bit linear PCM.
- Apple devices running iOS 5.0 or later support more than two channels of audio input in USB Host Mode. Accessories that use this feature must bundle all audio input channels into a single audio streaming interface.
- Apple devices running iOS 6.0 or later support more than two channels of audio output in USB Host Mode. Accessories that use this feature must bundle all audio output channels into a single audio streaming interface.

Accessories should be tested against the latest OS X Audio driver; the application Audio MIDI Setup, in the OS X Applications/Utilities folder, can be used to verify compatibility. Additionally, the USB Prober application, available with Xcode, should be used to inspect and verify the contents of USB descriptors.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

23.1.6 USB Device Mode Audio

The accessory must register a USB Device Mode component during Identification to make use of this feature. Additionally, the USB Device Mode component must report the audio sampling rates that the accessory supports. All accessories must at least support the 32000 Hz, 44100 Hz, and 48000 Hz sample rates.

Note: The Apple device USB isochronous audio data endpoint descriptor bmAttributes field erroneously returns the Synchronization Type field (D3:2) as b00 (no synchronization) instead of the correct value, b11 (synchronous). The Apple device supports synchronous data transfers, so the accessory must ignore those erroneous attribute bits. The erroneous b00 value is retained for backwards compatibility with older accessories.

When receiving USB Device Mode audio over a USB streaming interface, the accessory may buffer data to achieve consistent audio playback. Since buffering introduces latency, it is recommended that this latency be kept below 200 ms to ensure timely response to user input, such as a play/pause button.

All accessories that support the USB Device Mode Audio feature via iAP2 must send or receive the following iAP2 control session message(s):

StartUSBDeviceModeAudio (page 620)

USBDeviceModeAudioInformation (page 620)

StopUSBDeviceModeAudio (page 621)

23.2 Digital Audio Usage

23.2.1 USB Host Mode Audio

The accessory must authenticate and identify itself to the Apple device using iAP2 before the Apple device will enumerate and start using the USB Audio interfaces. When audio streaming is active, the streaming interface will be selected, and when audio streaming is inactive, the zero-bandwidth interface will be selected. Device-powered accessories must use this event to trigger compliance with requirements stated in Power (page 480).

Service@iPadandLED.com.tw Bay Wu

23.2.2 USB Device Mode Audio Usage

The accessory's USB host controller must enumerate the device and select the second configuration, which corresponds to a composite USB Device Mode Audio and iAP2 HID device. Once an iAP2 connection is established and the accessory has successfully identified and authenticated, it must select a nonzero-bandwidth alternate USB audio streaming interface on the Apple device. The accessory must not send USB IN tokens to the audio streaming interface until a USBDeviceModeAudioInformation (page 620) message has been received.

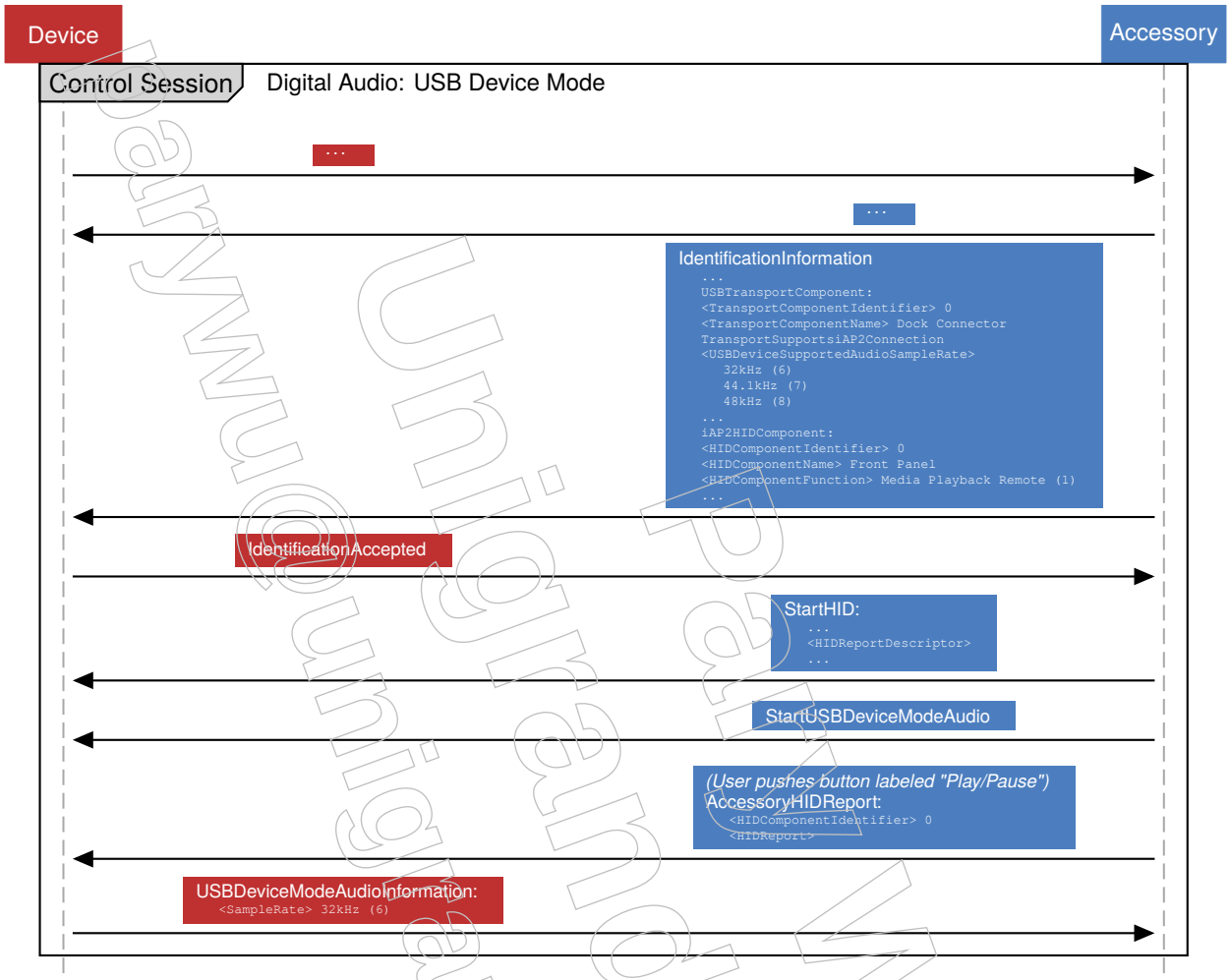
When the accessory is ready for audio streaming output, it must send StartUSBDeviceModeAudio (page 620) to the device. The device will inform the accessory of the stream properties via USBDeviceModeAudioInformation (page 620) messages. Multiple update messages will be sent while streaming is active to inform the accessory of changes in audio stream properties. When the accessory no longer wants to consume audio streaming output, it must send StopUSBDeviceModeAudio (page 621) to the device.

23.3 Digital Audio Examples

23.3.1 Typical USB Device Mode Digital Audio

The following sequence diagram illustrates how to correctly identify for and start USB Device Mode Digital Audio using iAP2.

Jay Wu Parkland LTD
jay@parkland.com.tw



Darwin Wu
Unigrand LTD
Darwin@unigrand.com.tw

24. External Accessory Protocol

The iOS External Accessory framework provides accessories with the means to communicate with one or more iOS apps via one or more External Accessory (EA) sessions. These sessions present a simple read/write bytestream interface; it is up to the accessory developer to specify a custom protocol between the app and the accessory. The design and maintenance of communication protocols between accessories and applications are entirely the responsibility of the developers of these products. Apple makes no attempt to secure protocol name space or provide for communication security between accessories and applications.

All accessories may implement this feature by defining and implementing an iAP2 EA session. Additionally, accessories with a USB Host Mode transport component may choose to implement an EA Native Transport (USB Host Mode). This will generally provide higher performance than the iAP2 EA session. Apple does not guarantee reliable delivery and/or a minimum level of transfer speed.

24.1 External Accessory Protocol Requirements

The following requirements apply to accessories that support this feature:

- Accessories must declare at least one `ExternalAccessoryProtocol` parameter in their IdentificationInformation (page 579) message.
- All declared `ExternalAccessoryProtocolIdentifier` and `ExternalAccessoryProtocolName` parameters must be unique in an IdentificationInformation (page 579) message.
- Protocol names must be in reverse-DNS format and must be associated with domains that are registered with the Internet Corporation for Assigned Names and Numbers (ICANN), so that Apple can rely on each protocol having a unique owner. Apple does not require a developer to be the owner of the domain name used, but does require that the developer have permission to use the domain name for the protocol. Protocol names used for demonstration or example purposes in accessory reference designs must not be used in shipping accessories.
- Accessories must choose a match action (Table 52-11 (page 582)) for each supported protocol. Apple recommends that accessories support the App Match feature (see App Match (page 212)) and use 1 unless the protocol in question is meant for infrequently accessed accessory features, such as firmware updates or diagnostics.
- Accessories must demonstrate use of all supported protocols during self certification.

Apple Confidential
Do Not Distribute

Accessories must also either implement one iAP2 EA session or one EA Native Transport over a USB Host Mode transport component. Accessories only need to implement one iAP2 EA session regardless of the number of EA protocols supported using that session. Conversely, an EA Native Transport (USB Host Mode) can only support a single EA protocol.

Accessories may implement both an iAP2 EA session for some protocols and use EA Native Transports for others, but an EA protocol must not use both an iAP2 EA session and an EA Native Transport.

For more information on developing iOS apps that communicate with accessories, see *External Accessory Programming Topics* in the iOS SDK documentation at http://developer.apple.com/library/ios/featuredarticles/ExternalAccessoryPT/Introduction/Introduction.html.

24.1.1 iAP2 EA Session Requirements

All accessories that support the External Accessory Protocol feature via iAP2 must send or receive the following iAP2 control session message(s):

StartExternalAccessoryProtocolSession (page 598)

StopExternalAccessoryProtocolSession (page 598)

To implement the iAP2 EA session, accessories must also declare the presence of that session during iAP2 link synchronization. Details on the session implementation can be found in External Accessory Session (page 575).

All accessories that support the External Accessory Protocol feature via iAP2 may also send or receive the following iAP2 control session message(s):

StatusExternalAccessoryProtocolSession (page 598)

If the accessory declares support for StatusExternalAccessoryProtocolSession (page 598), the Apple device will allow multiple iAP2 EA sessions per protocol (excluding EA Native Transport).

24.1.2 EA Native Transport (USB Host Mode) Requirements

To implement an EA Native Transport over USB Host Mode, an accessory must declare one additional USB interface in its device descriptor. The interface must have two alternate settings; alternate setting 1 must have one bulk IN and one bulk OUT endpoint, and alternate setting 0 must be a zero-bandwidth setting with no endpoints. Also, the accessory must declare the presence of an EA Native Transport component during identification.

Confidential Part Wu
Paul@unionchip.com.tw

Note: Accessories must not reuse the same USB Bulk In/Out endpoints for multiple purposes, such as iAP2 and EA native transport. They also must not shut down the iAP2 connection once a EA Native Transport session has been started.

Table 24-1 USB Interface Interface Descriptor (Alternate Setting 1 - Active Session) for External Accessory Native Transport (USB Host Mode)

USB Descriptor	Value	Comments
Interface Number	`0xNN`	Must be different from the iAP2 interface number if also present
Interface Class	`0xFF`	Vendor-specific interface
Interface SubClass	`0xF0`	MFi Accessory
Interface Protocol	`0x01`	External Accessory Native Transport
Interface String	`com.company.protocol`	Must match an identified `ExternalAccessoryProtocolName`
Number of Endpoints	2	1 Bulk IN and 1 Bulk OUT endpoint descriptor must be specified
Alternate Setting	1	

Table 24-2 USB Interface Descriptor (Alternate Setting 0 - Zero Bandwidth) for External Accessory Native Transport (USB Host Mode)

USB Descriptor	Value	Comments
Interface Number	`0xNN`	Must be the same as the default interface
Interface Class	`0xFF`	Vendor-specific interface
Interface SubClass	`0xF0`	MFi Accessory
Interface Protocol	`0x01`	External Accessory Native Transport
Interface String	`com.company.protocol`	Must match an identified `ExternalAccessoryProtocolName`
Number of Endpoints	0	
Alternate Setting	0	

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

24.2 External Accessory Protocol Usage

24.2.1 iAP2 EA Session Usage

iAP2 EA Session protocol initiation is always driven by an app running on the Apple device. When the accessory receives a StartExternalAccessoryProtocolSession (page 598) message, it may start to send and receive traffic on the identified iAP2 EA Session for the specified protocol and session identifier.

Conversely, the accessory must cease sending and receiving traffic for the specified session identifier when a StopExternalAccessoryProtocolSession (page 598) message is received. Device-powered accessories may also be required to transition to a low power state as specified in the Accessory Power feature.

24.2.2 EA Native Transport (USB Host Mode) Usage

If it is using EA Native Transport over USB Host Mode, the accessory's default interface will be selected by the Apple device when an app opens an EA session with the accessory. Once selected, the accessory must continually service both the bulk IN and OUT endpoints to maintain communication with the app. When the app has closed the EA session, the alternate zero-bandwidth interface will be selected to signal the accessory that EA traffic for that protocol has ceased. Device-powered accessories may also be required to transition to a low power state as specified in the Accessory Power feature.

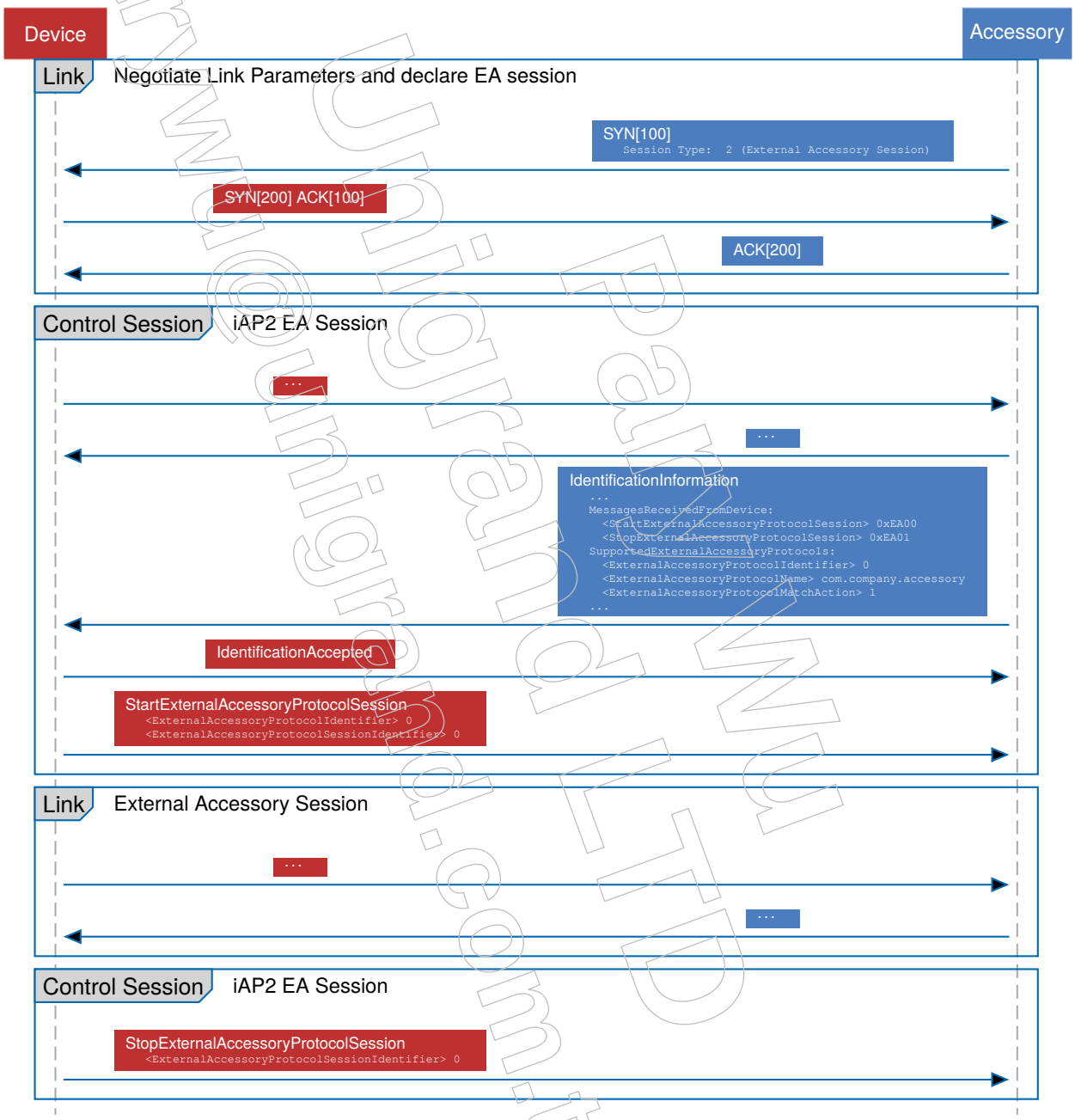
EA Native Transport (USB Host Mode) performance will vary based on many factors including, but not limited to:

- USB Full Speed versus USB High Speed.
- Bulk endpoint packet size.
- Depth of USB packet queues.
- Other concurrent USB traffic on the same bus/controller.

Licensed By Wu Bai Chi Hang LTD
bai@baichi-hang.com.tw

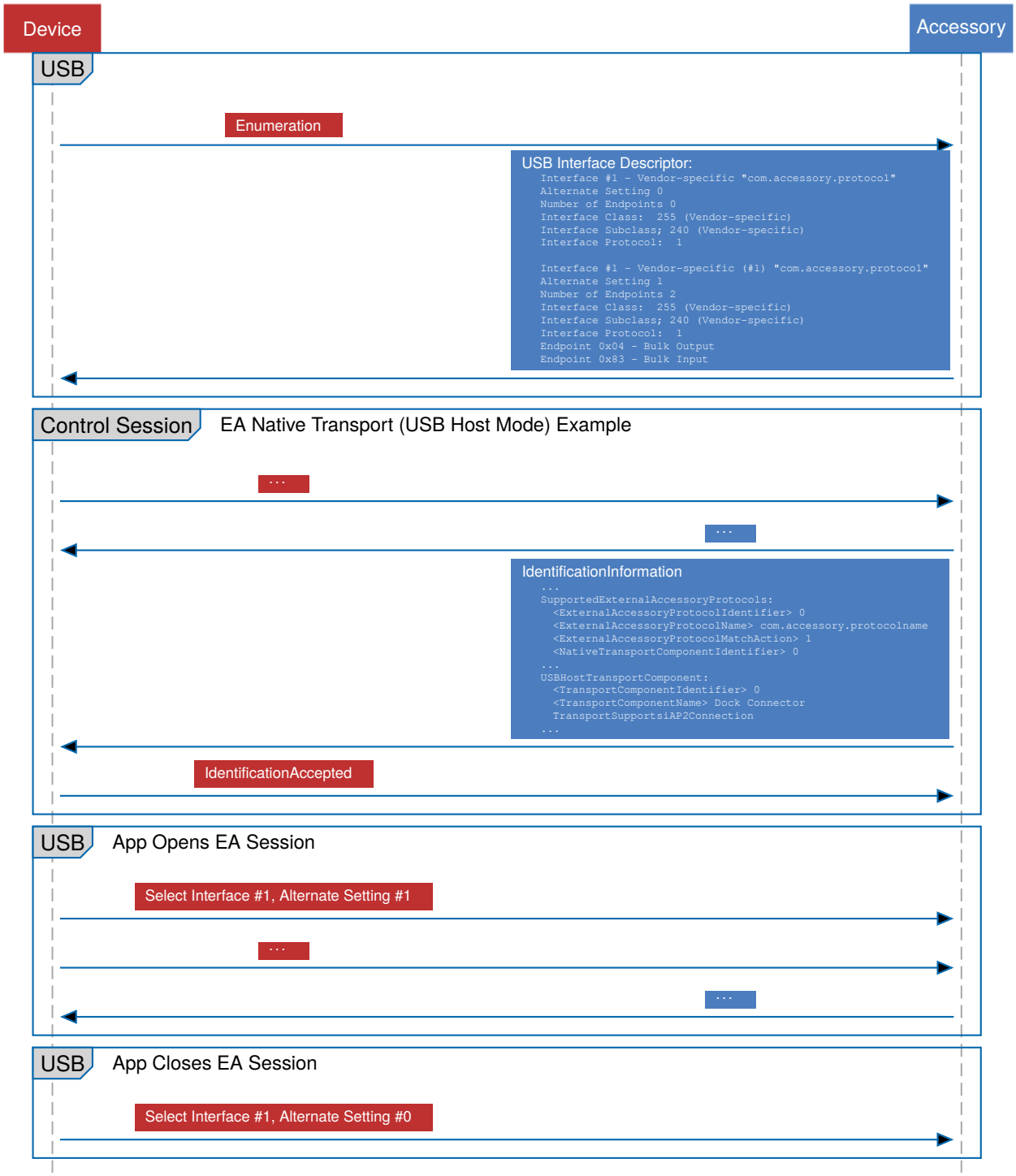
24.3 External Accessory Protocol Examples

24.3.1 iAP2 EA Session Example



Apple Inc. Confidential
Authorized Personal Use Only
Unauthorized Distribution Prohibited
Copyright © 2014 Apple Inc. All Rights Reserved.

24.3.2 EA Native Transport (USB Host Mode) Example



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

25. Headphones

Accessories may implement headphone components. Apple devices treat accessory headphones differently from accessories with integrated speakers. Headphones are expected to help provide the user with a personal audio experience.

> **Note:** All accessories must comply with TDMA noise requirements. See TDMA Noise (page 53).

25.1 Product Design

The following product design requirements apply to accessory headphone components:

- They must have drivers that are positioned on, over, or in a user's ears.
- If the headphones also have an integrated microphone, it must be positioned to primarily record audio from the user's mouth.

25.2 Audio and Data Interfaces

Headphones must establish audio connections to Apple devices via one of the following interfaces:

- Headphone jack (see Headphone Jack (page 382))
- Lightning connector (see Apple Lightning Connector (page 114))
- Bluetooth HFP and A2DP (see Bluetooth (page 216))

If connected via the headphone jack, headphones must support the Headphone Remote and Mic feature (see Headphone Remote and Mic (page 387).

If connected via the Lightning connector, headphones must integrate a Lightning Audio Module (see Apple Lightning Audio Module (page 80)).

If connected via Bluetooth, the headphones must support iAP2 and identify themselves as headphones to the Apple device by declaring the presence of a HID Headphone Remote component (see Accessory Identification (page 205). Integration of a microphone for hands-free telephony is also recommended.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

25.3 Remote Controls

Accessory headphones that connect to an Apple device via the Lightning connector or Bluetooth must implement a 3-button array that mimics the behavior of the Apple Headphone Remote. The buttons must be implemented as one of the following:

- direct electrical connections to the Lightning Audio Module S0-S2 inputs as specified in S0-S2 Headphone Remote Inputs (page 95)
- part of a HID Headphone Remote component via the Lightning Audio Module
- part of a HID Headphone Remote component via Bluetooth (HID over iAP2)

Additional remote control buttons/control surfaces may be implemented to control media playback and interact with Apple features like iTunes Radio.

25.4 Native Voice Recognition

If the accessory has native voice recognition capability of its own separate from the Siri feature, it must not be triggered via means like 'always listening' or buttons/switches that toggle between native voice recognition and Siri. The native voice recognition capability must be triggered via one of the following two approaches:

- The accessory may implement a physical button or control that is different/separate from the physical button or control that triggers Siri.
- The accessory may use the same physical button or control, but the user must trigger the native capability via a 'press-and-release' action, as opposed to the 'press-and-hold' action associated with Siri.

25.5 Test Procedures

25.5.1 Product Design

1. Verify that the headphone drivers are positioned on, over, or in the user's ears.
2. If the headphone has an integrated microphone, verify that it is positioned to primarily record audio from the user's mouth.

25.5.2 Headphone Jack

If the headphone connects to the Apple device via the headphone jack:

1. Execute Headphone Jack Test Procedures (page 386).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2. Execute Headphone Remote and Mic Test Procedures (page 404).

25.5.3 Lightning

If the headphone connects to the Apple device via the Lightning connector:

1. Verify that it integrates a Lightning Audio Module.
2. Verify that it integrates a Lightning (C10E) connector.
3. Verify that it implements 3 buttons that mimic the behavior of the Apple Headphone Remote.
4. Execute Lightning Audio Module Test Procedures (page 112).

25.5.4 Bluetooth

If the headphone connects to the Apple device via Bluetooth:

1. Verify that it supports iAP2.
2. Verify that it implements 3 buttons that mimic the behavior of the Apple Headphone Remote.
3. Verify that it identifies a HID Headphone Remote component to the Apple device over iAP2.
4. Execute Bluetooth Test Procedures (page 230).

25.5.5 Controls

For all headphones:

1. Verify that there is a button for Volume Up and that it increases the volume on the Apple device.
2. Verify that there is a button for Volume Down and that it decreases the volume on the Apple device.
3. Verify that there is a button that has the same center button behavior as on the Apple Headphone Remote.
    a. Verify Play and Pause works for songs and video tracks when pressing center button.
    b. Verify Next Track works when pressing center button twice quickly.
    c. Verify Fast Forward works when pressing then press-and-holding the center button.
    d. Verify Previous Track works when pressing center button 3 times quickly.
    e. Verify that pressing center button once will answer incoming phone call.
    f. Verify that pressing center button once will end phone call.
    g. Verify that holding down center button will decline an incoming phone call. Verify you hear 2 low beeps to confirm that you declined the call.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

If the Volume Up and Volume Down buttons are implemented as part of a HID Headphone Remote component:

1. Verify that pressing the Volume Up button results in a button press usage immediately followed by a button released usage even if the button is pressed and held. It must not be possible to ramp the Apple device device to full volume by pressing and holding the button.
2. Verify that pressing the Volume Down button results in a button press usage immediately followed by a button released usage even if the button is pressed and held. It must not be possible to ramp the Apple device device to zero volume (mute) by pressing and holding the button.

If the headphones have additional controls/buttons:

1. Verify that each additional control has only one function associated with it.
2. Verify that each additional control has appropriate iconography/labeling for the associated function.
3. Verify that each associated function only triggers due to direct user action.

25.5.6 Native Voice Recognition

If the headphone has its own native voice recognition capability separate from Siri:

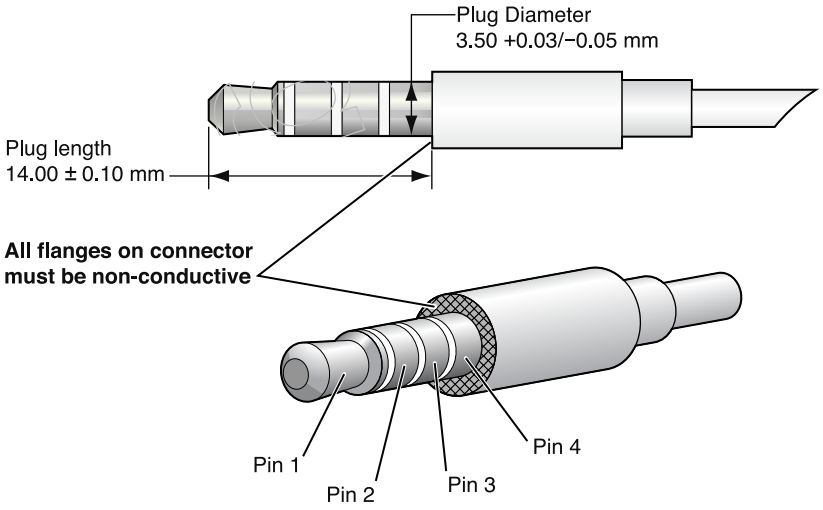
1. Verify that it does not 'always listen' and trigger either the native voice recognition or Siri
2. Verify that it does not toggle between native voice recognition and Siri via buttons or switches
3. Verify that it has its own separate physical button or control that activates native voice recognition or triggers the native voice recognition via a 'press-and-release' action, while 'press-and-hold' triggers Siri.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

26. Headphone Jack

Each Apple device contains a standard stereo headphone jack supporting microphone input and control buttons. Most dimensions of the headphone jack conform to the JEITA standard RC 5325A, "4-Pole Miniature Concentric Plugs and Jacks."

Figure 26-1 Headphone/Microphone Plug Dimensional and Flange Details



The following requirements apply to all accessory connectors intended for insertion into an Apple device's headphone jack:

- The connector plug length and diameter dimensions must match those specified in Figure 26-1 (page 382).
- The connector's cable shell diameter must not exceed 5.9 mm for a minimum length of 11 mm from where the flange starts.
- The connector's cable shell must not have any bends, curves, or protrusions for a minimum length of 11 mm from where the flange starts.

Additionally, every accessory connector with 4 conductors intended for insertion into an Apple device's headphone jack must not have a conductive flange as specified in Figure 26-1 (page 382).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Any accessory headphone jack that can accept the Apple Earpods with Remote and Mic and is electrically connected to the headphone jack on an Apple device must support full functionality of the Apple Earphones with Remote and Mic.

26.1 Pin Assignments

The following table lists pin assignments for the headphone jack:

Table 26-1 Headphone Plug Pin Assignments

Pin	Description
1	Audio Output (Left)
2	Audio Output (Right)
3	Signal Return
4	Microphone Input

Note: Some Apple devices will apply and then measure a voltage across Pin 3 and Pin 4 in order to determine whether the headset is wired as specified in Table 26-1 (page 383) or not. The measured voltage must be greater than 1.5 V.

Some Apple devices will also apply a reverse bias voltage across Pin 3 and Pin 4 and then measure the response for the same purpose. The measured voltage must not be less than -0.75 V.

26.2 Extension Cables

Accessories must not bundle headphone jack extension cables or adapters unless:

- The accessory is form-fitting as defined in Form-Fitting Accessories (page 153).
- The accessory design prevents the Apple Earpods with Remote and Mic from being connected.

All bundled headphone jack extension cables or adapters must:

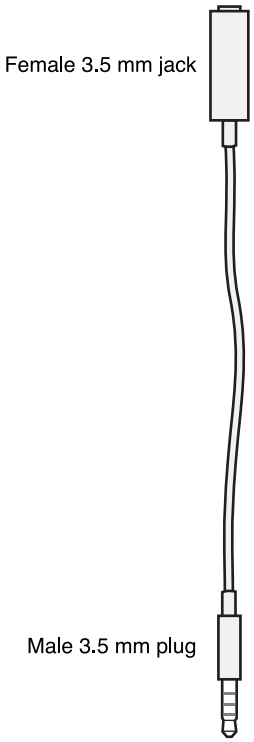
- Incorporate 3.5 mm plugs and jacks with 4 conductors.
- Carry the 4 signals from the plug to the jack using separate conductors.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- Not support the Headphone Remote and Mic feature as specified in Headphone Remote and Mic (page 387).

Additionally, the headphone jack extension cable or adapter should be as short as possible.

Figure 26-2 Example Headphone Extension Cable



26.3 Example Circuitry

The following figure is an example of circuitry in a headphone/microphone accessory with an action button.

Figure 26-3 Typical circuitry in a headphone/microphone accessory

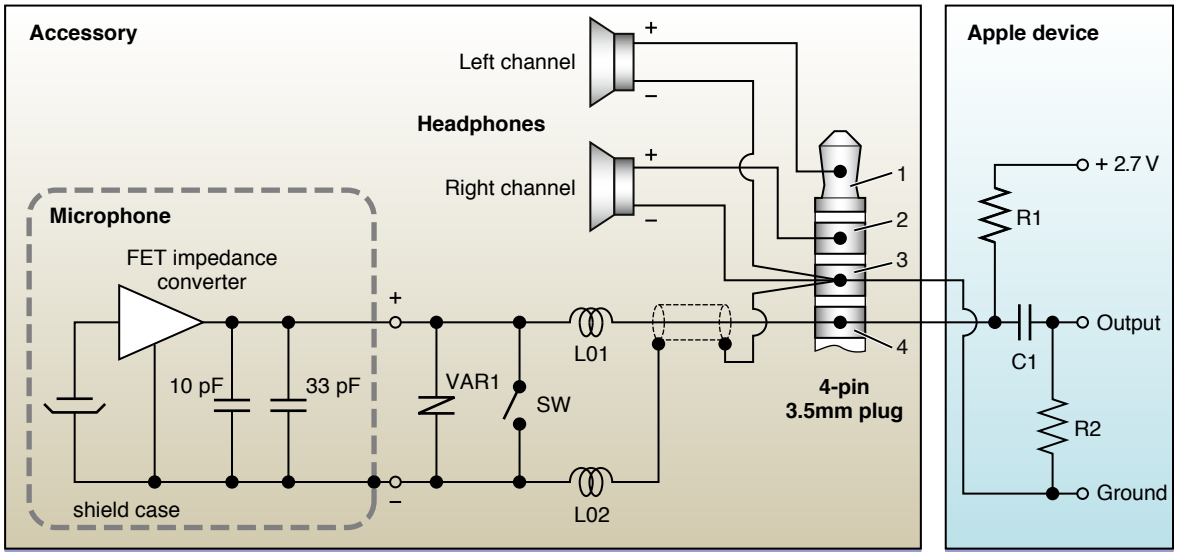


Table 26-2 Recommended component values for typical circuitry in a headphone/microphone accessory

Component	Value
C1	1 µF
R1	2.21 kΩ ±1%
R2	Codec input impedance ≥ 2 kΩ
L01/L02	600 Ω at 100 MHz
SW	Action Button
VAR1	12 V varistor

Note: The value given for L01 and L02 is typical. These ferrite chokes reduce time division multiple access (TDMA) noise; their exact values depend on the specific accessory design.

Switch SW shorts the microphone signal to ground. The Apple device treats its closure as a headset button press and initiates a context-specific action (for example, answering a phone call). The microphone bias current must be between 210 and 500 µA, measured into a circuit pulled up to 2.7 V through 2.21 kΩ, to ensure button

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

press detection. The recommended microphone sensitivity is -44 dBV with a maximum impedance of 2.2 kΩ at the output of the equivalent circuit shown above, measured under test conditions of Vs = 2.0 V, RL = 2.2 kΩ, Ta = 20° C, and relative humidity = 65%.

26.4 Test Procedures

1. Verify that every accessory cable and every connector that can physically be inserted into an Apple device's headphone/mic jack has non-conductive flanges.
2. For headphones with removable cables, ensure that end of the cable closest to the microphone is inserted into the headphone and not the Apple device. Perform subsequent tests on the Apple device end of the cable.
3. Verify that the 3.5 mm shell diameter does not exceed 5.9 mm for a minimum length of 11 mm from where the flange starts.
4. Verify that the 3.5 mm plug length is 14.00 mm (+,- 0.10 mm).
5. Verify that the 3.5 mm plug diameter is 3.50 mm (+ 0.03 / - 0.05 mm).
6. Verify 3.5 mm jack can be fully inserted into the Apple device's headphone port. There must not be physical defects in the 3.5 mm jack preventing insertion.
7. Verify that there are no grounding issues, heavy static, audio interference, or audio interference when headset is attached to Apple device.
    a. Repeatedly plug the headset in and out of the headphone jack.
    b. Wiggle the connector.
    c. Touch the input connector to the rim of the headset.

Unauthorized Party Wu
Unauthorized LTD
tanwu@authorized.com.tw

27. Headphone Remote and Mic

Apple devices can receive button press information from accessory headsets that incorporate the headphone remote and mic system feature via the headphone jack (see Headphone Jack (page 382).

27.1 Headphone Remote and Mic Requirements

All accessories implementing the Headphone Remote and Mic feature must comply with the following requirements:

- The remote microphone must be located 120-160 mm from the center of a headphone driver.
- There must be 6 wires originating from the 3.5 mm audio plug that attaches the Apple device to the headphone, corresponding to the following signals:
  - Right Driver
  - Right Return
  - Left Driver
  - Left Return
  - Microphone Bias
  - Microphone Return
- All signals must be run separately to their respective components.
- There must be 3 physical remote buttons that perform the volume up, volume down, and push-to-talk functions.

Accessory headphones that support the Headphone Remote and Mic feature must integrate the following components:

- The Apple-provided transmitter chip from Table 27-1 (page 391).
- A MEMS digital microphone from Table 27-10 (page 404).

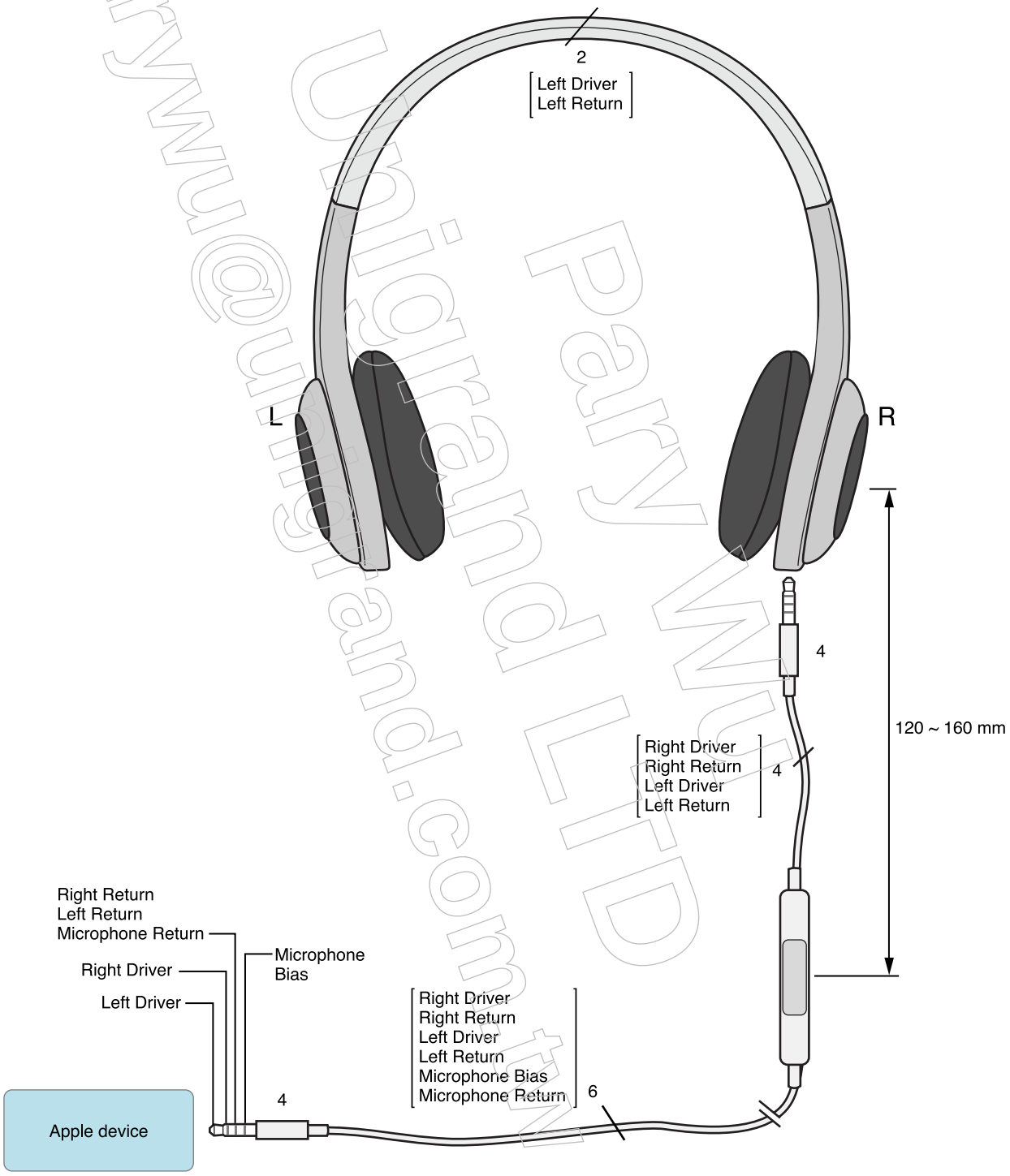
Accessory headphones must comply with the following requirements:

- The drivers must have a minimum load impedance of 16 ohms.
- The drivers must have a maximum load capacitance of 150 pF.

Party Wu (partywu@uniguand.com) Unigrand Ltd

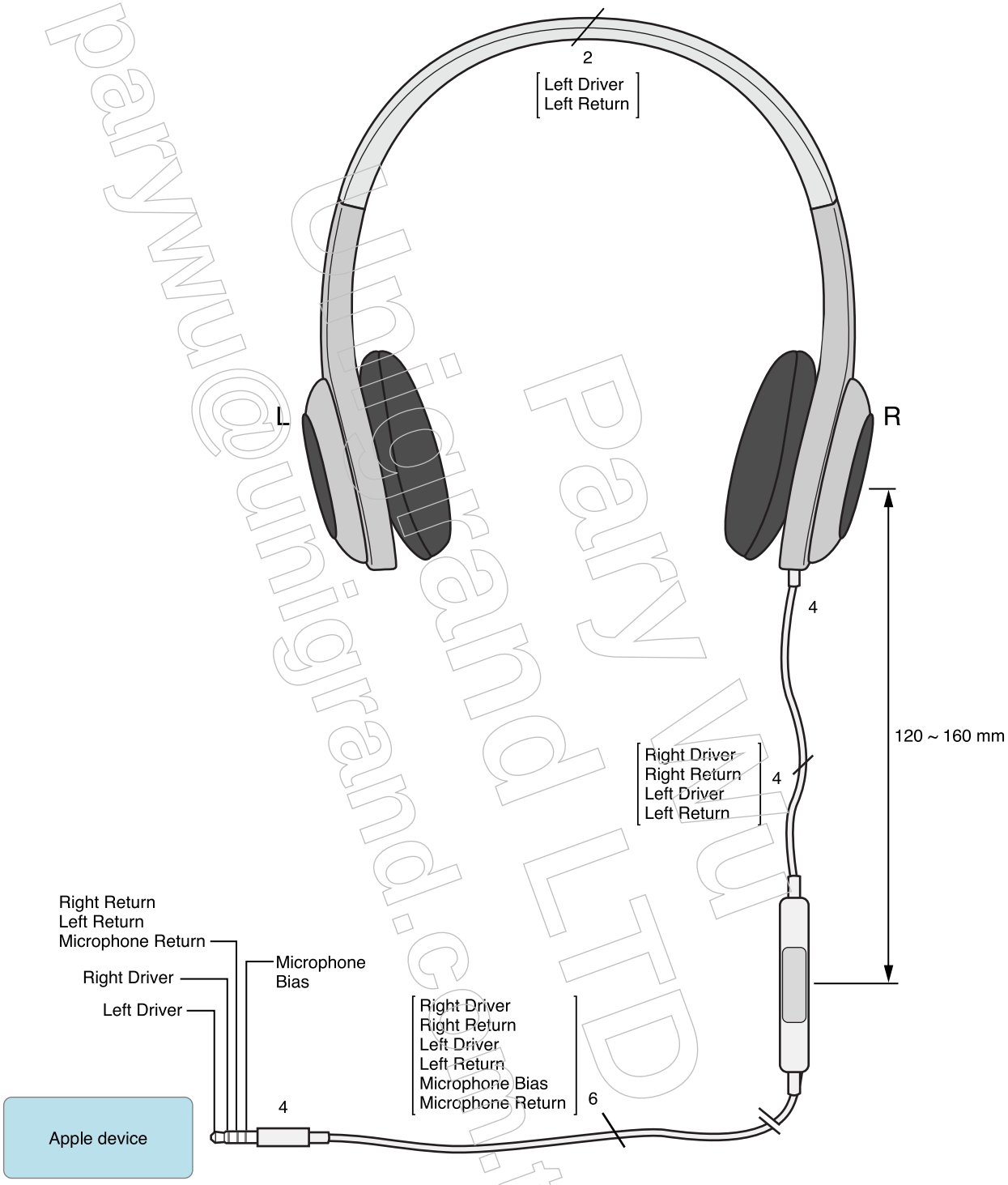
Accessory headphones must also implement one of the following configurations with one exception. The microphone may be located on either the left or right side of the headphone.

Figure 27-1 Example Headphone Remote and Mic System A



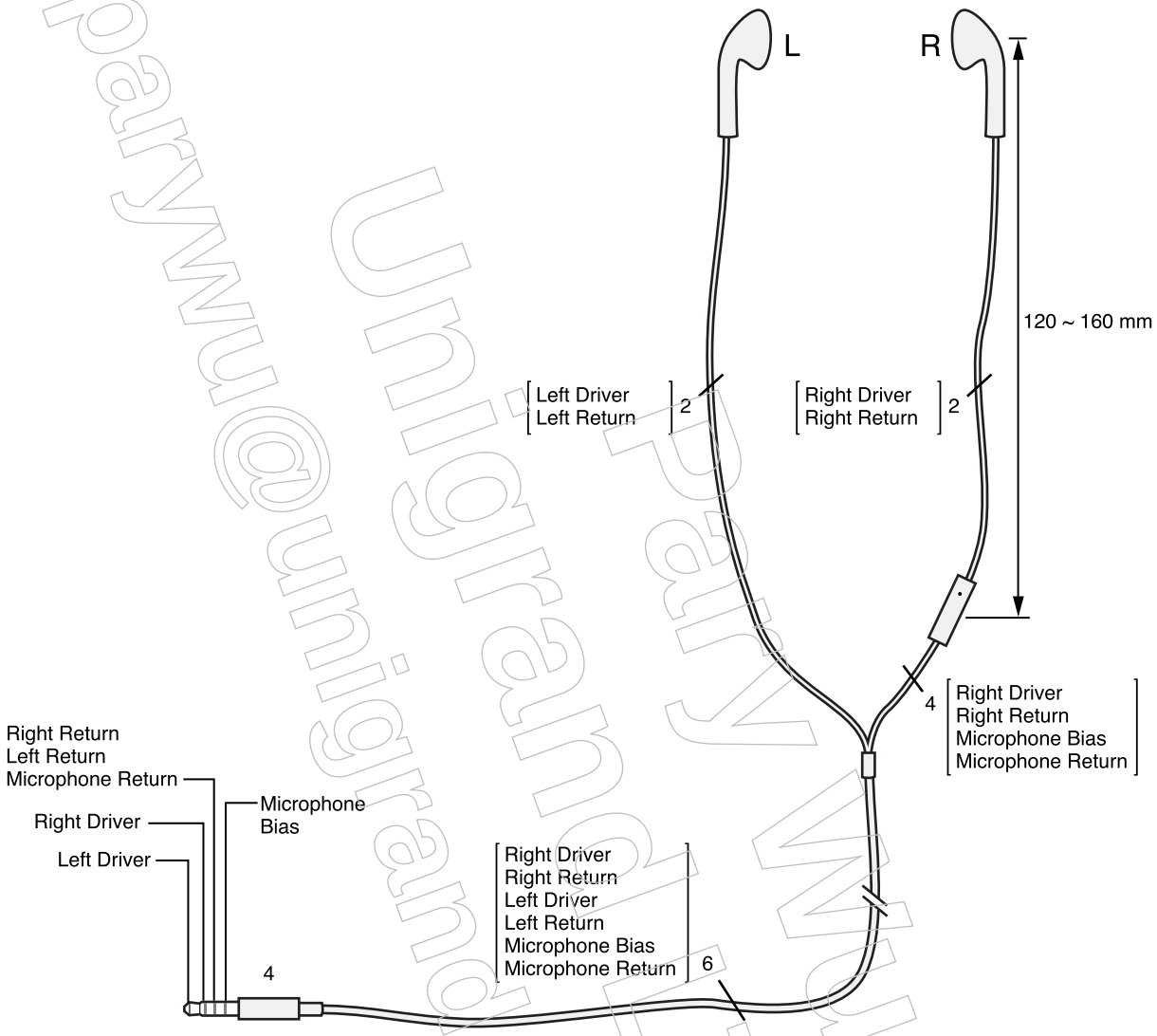
Unigrand Ltd Paryvu@unigrand.com

Figure 27-2 Example Headphone Remote and Mic System B



parywu@unigrand.com.tw Pary Wu Unigrand LTD

Figure 27-3 Example Headphone Remote and Mic System C



27.2 Headphone Remote Transmitter Chip Usage

The transmitter chip operates together with a controller in the Apple device to enable remote button press detection via their common microphone bias line. The transmitter chip is a MEMS microphone interface and button decoder device located at the microphone and button end of the line, in the headset accessory. The controller in the Apple device provides regulated downstream power (nominally 2.7 V or 2.0 V) to the transmitter chip and MEMS microphone through the microphone bias line, and it decodes the button information from the transmitter chip.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

The transmitter sends button-press information over the microphone bias line in either of two modes: button mode or tone mode. If the voltage on the microphone bias line is less than 2.35 V, indicating that the microphone is not in use, the transmitter enters button mode and sends button-press information as discrete voltage levels. If the microphone bias voltage is greater than 2.35 V, indicating that the microphone is in use, the transmitter enters tone mode and sends the same button-press information as ultrasonic tone sequences in the range of 99 to 300 kHZ.

Subjective listening tests with the newest Apple device are recommended to determine which chip produces the best user experience.

Table 27-1 Transmitter Chip Part Numbers

Part Number	Usage
MFI353S2429	Noise-occluding headphones. These are headphones that block or cancel outside sound.
MFI353S2430	Standard non-occluding headphones.

27.2.1 Transmitter Chip Pin Assignments and Physical Packaging

Table 27-2 Transmitter Chip Pin Assignments

Number	Name	I/O	Description
A1	TONE	Output	Tone generator output
A2	GND	Power	Audio return
B1	MIC	Input	Microphone bias
B2	REM	Input/Output	Remote switch network
C1	VSHUNT	Input	Shunt regulator supply
C2	MICPWR	Output	Microphone power

Taiwan Wu
Unigrand Ltd.
tw.linkedin.com/in/taiwanwu

Figure 27-4 Transmitter Chip Package

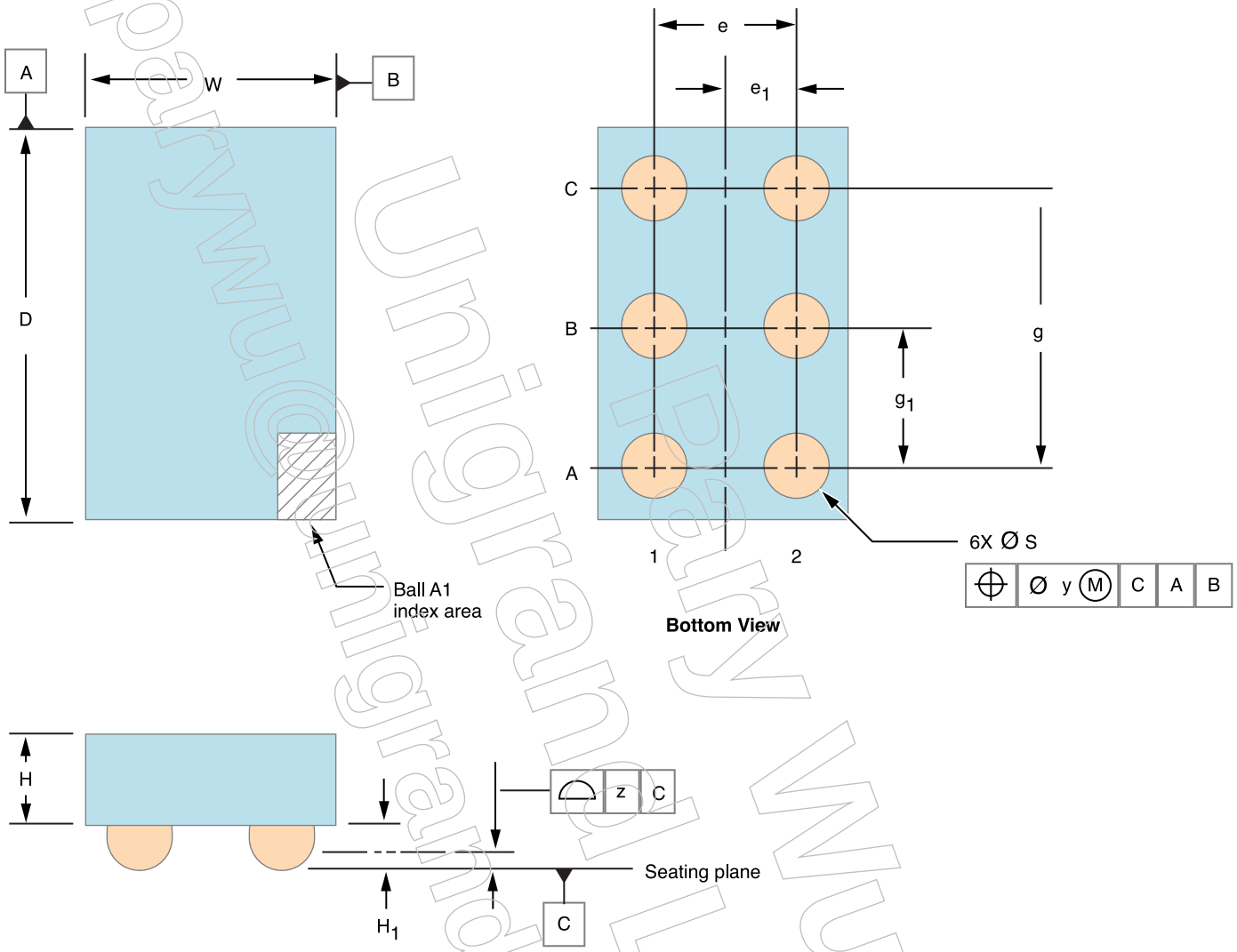


Table 27-3 Transmitter Chip Package Dimensions

Dimension	Value in mm
W	0.95/0.85
D	1.45/1.35
H	0.50 max
H1	0.19/0.15
e	0.50
e1	0.25

Paywu@unigrand.com.tw Unigrand LTD Payway Wu

Dimension	Value in mm
g	1.00
g1	0.50
s	0.25/0.21
y	0.015
z	0.05

27.2.2 Transmitter Chip Maximum Voltage and Current Ratings

Table 27-4 (page 393) lists the transmitter chip's maximum voltage and current ratings while operating over a free-air temperature range (TA) of -40 to +85° C. The chip's maximum storage temperature range is -65 to +150° C. Stresses beyond these ratings may cause permanent damage, and exposure to maximum conditions for extended periods may degrade its reliability. These are stress ratings only; functional operation of the chip at these or any other conditions beyond those specified is not implied.

All voltages are measured with respect to ground. All input and output clamp-current ratings must be observed.

Table 27-4 Transmitter Chip Maximum Voltage and Current Ratings

Symbol	Description	Minimum	Maximum
VSUPPLY	Supply voltage, VSHUNT, MIC	-0.5 V	4.6 V
VI	Input voltage, REM	-0.5 V	4.6 V
V0	Output voltage, MICPWR, TONE	-0.5 V	4.6 V
IIK	Input clamp current, REM (VI < 0)	-20 mA	
I0K	Output clamp current, MICPWR, TONE (VO < 0)	-20 mA	
ISUPPLY, IGND	Continuous current through VSHUNT, MIC, or GND	-50 mA	50 mA

27.2.3 Transmitter Chip Thermal Impedance

The transmitter chip's package thermal impedance, calculated in accordance with Specification JESD51-7, is 123° C/W.

Licensed to Wu Tai Wu (taiwu@etiquettelab.com)

27.2.4 Transmitter Chip Moisture Sensitivity

The transmitter chip has a Moisture Sensitivity Level (MSL) of 1, as defined by industry standard JEDEC specifications.

27.2.5 Transmitter Chip Electrical Characteristics

Table 27-5 (page 394), Table 27-6 (page 395), and Table 27-7 (page 396) list the transmitter chip's electrical and timing characteristics under the following conditions:

- Operating temperature = -40 to +85° C.
- Button mode, VMICBIAS = 1.8 to 2.1 V; MIC is connected to VMICBIAS through a 2.21 kΩ ±1% resistor.
- Tone mode, VMICBIAS = 2.56 to 2.84 V; MIC is connected to VMICBIAS through a 2.21 kΩ ±1% resistor.

The values in the "Typical" column of the tables are measured at 25° C.

Table 27-5 Transmitter Chip Electrical Characteristics (General)

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
$I_{MICBIAS-B}$	Quiescent Current into MIC+VSHUNT	Button Mode, $V_{MICBIAS}$ = 2.1 V		3 µA	6 µA
$I_{MICBIAS-B}$	Quiescent Current into MIC+VSHUNT	Button Mode, $V_{MICBIAS}$ = 1.5 V		3 µA	6 µA
I_{MIC-T}	Quiescent Current into MIC	Tone Mode		34 µA	46 µA
$I_{VSHUNT-T}$	Quiescent Current into VSHUNT	Tone Mode (see note below)		60 µA	70 µA
I_{MIC-TA}	Active Current into MIC	Tone Mode		35 µA	45 µA
$I_{VSHUNT-TA}$	Active Current into VSHUNT	(see note below)		104 µA	118 µA
V_{TR}	Tone Mode Threshold Voltage	MIC Rising (Microphone enable), V_{MICPWR} = 1.0 V	2.20 V	2.35 V	2.50 V

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
V_{TF}	Tone Mode Threshold Voltage	MIC Falling (Microphone disable), V_{MICPWR} = 400 mV	0.55 V	0.8 V	1 V
V_{MICPWR}	MICPWR Output Voltage	I_{MICPWR} = 120--150 µA	1.51 V	1.56 V	1.61 V
R_{SO}	Shunt Regulator Output Impedance	Freq = 100 Hz	5 Ω	18 Ω	25 Ω
R_{SO}	Shunt Regulator Output Impedance	Freq = 20 Hz	12 Ω	21 Ω	35 Ω
R_{ONA}	Switch A, R_{DSON}	Tone Mode, I_{MICPWR} = 1 mA, $V_{MICBIAS}$ = 2.56V		40 Ω	55 Ω
R_{ONB}	Switch B, R_{DSON}	V_{MIC} = 1.2V, I_{REM} = 1 mA		22 Ω	30.5 Ω

Note: This current is pulled through R_{VSHUNT} between MIC and VSHUNT and is the minimum current to keep VSHUNT regulated at 1.56 V. Excess current through R_{VSHUNT} is available to the load at MICPWR. Excess current not used by the load at MICPWR is internally shunted to GND.

Table 27-6 Transmitter Chip Electrical Characteristics (Tone Mode)

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
$e_{n\text{-}mic100}$	MIC Integrated Noise	100 Hz to 20 kHz		1.5 µVrms	2 µVrms
f_{TONE1}	Button 1 Frequency	R_{REM} = 6.81 kΩ	109 kHz	130 kHz	159 kHz
f_{TONE2}	Button 2 Frequency	R_{REM} = 9.42 kΩ	138 kHz	165 kHz	200 kHz
f_{REL}	Button Release Frequency		81 kHz	97 kHz	117 kHz

Part Number Unknown Pty LTD (unknown@unknown.com.tw)

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
R_{BT1}	Button 1 Boundary		6.61 kΩ	6.81 kΩ	7.01 kΩ
R_{BT2}	Button 2 Boundary		9.33 kΩ	9.42 kΩ	9.51 kΩ
V_{TA}	Tone Amplitude	R_{TONE} = 1 MΩ	350 mV	550 mV	720 mV
V_{TA}	Tone Amplitude	R_{TONE} = 100 kΩ	300 mV	515 mV	710 mV

Table 27-7 Transmitter Chip Electrical Characteristics (Button Mode)

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
t_{ONA}	Switch A Enable Time	Time to turn on Switch A	0.8 ms	1.2 ms	2 ms
t_{OFFB}	Switch B Disable Time	Time to turn off Switch B	0.7 ms	1 ms	2 ms
t_{REG}	Shunt regulator enable time	Time From MIC = 2.3 V to MICPWR = 1.56 V	1 ms	2.5 ms	3.5 ms

27.2.6 Transmitter Chip Theory of Operation

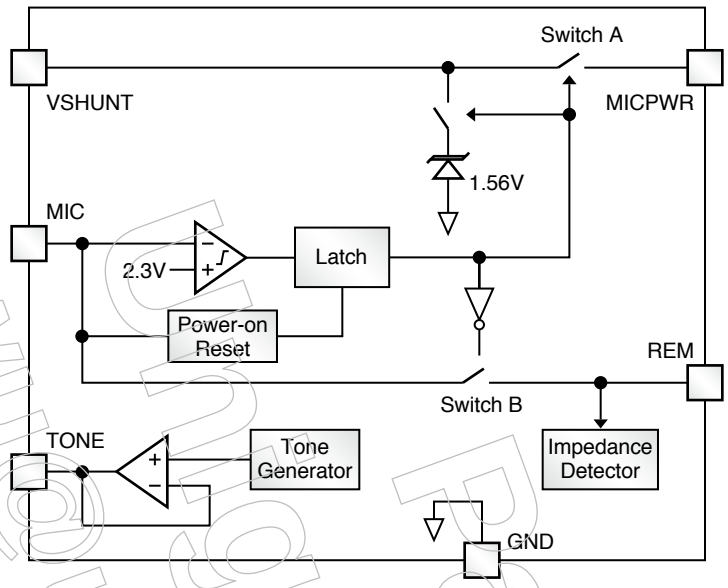
The transmitter chip has three primary functions:

- Provide an interface to a button switch-resistor network.
- Provide power for a local microphone.
- Provide a tone generator for sending discrete frequency tones on the bias line corresponding to button events.

The controller in the Apple device provides regulated downstream power (nominally 2.7 or 2.0 V) to the transmitter chip and microphone through the microphone bias line. Figure 27-5 (page 397) illustrates the functional components of the transmitter chip. In this diagram, a latch drives the configuration of switches A and B. The power-on reset monitors voltage on the MIC pin to ensure that there is a enough power before initiating the turn-on sequence; it shuts the chip down if there is insufficient voltage.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 27-5 Transmitter Chip Block Diagram



Button events are sent from the transmitter to the controller in one of two modes, button mode or tone mode. When a microphone is not present or is not in use, the transmitter is put in button mode by the controller in the Apple device, and button events are detected using discrete voltage levels. These discrete voltage levels are a percentage of a regulated output voltage on the microphone bias line. When a microphone is in use, the controller puts the transmitter into tone mode by placing more than 2.35 V on the microphone bias line, and the transmitter then sends button events using tone sequences of discrete frequencies in the range 99 kHz to 300 kHz.

27.2.7 Transmitter Chip Button Mode

In button mode, the transmitter chip operates as a pass-through element switching a button switch-resistor network onto the bias line. Each switch represents a unique button. When a button is pressed, the DC level on the bias line is changed and detected by the controller. Table 27-8 (page 397) shows the DETECT pin voltages with $V_{MICBIAS}$ = 2.0 V.

Table 27-8 Transmitter Chip DETECT Pin Voltages

Switch Closed	Voltage
S0	0.000 V ±1%
S1	1.510 V ±1%
S2	1.603 V ±1%

paywu@pubpad.com.tw Pay Wu Pubpad Inc. LTD

When the transmitter chip is in button mode (VMIC has never reached 2.35 V), it shorts the MIC and REM pins together and disables all other inputs and outputs. When a button event occurs, the DC voltage on the microphone bias line changes. Table 27-8 (page 397) shows the DC voltage corresponding to a given button press when using the R1 and R4 resistor values listed in Table 27-9 (page 403). This DC level is then detected by the controller in the Apple device. Switch S0 is a unique switch that shorts the V_{MIC} line to ground.

When the V_{MIC} line is shorted to ground, power is removed from the transmitter chip. When power recovers, the transmitter chip enters button mode or tone mode, depending on the voltage detected at the MIC pin.

27.2.8 Transmitter Chip Tone Mode

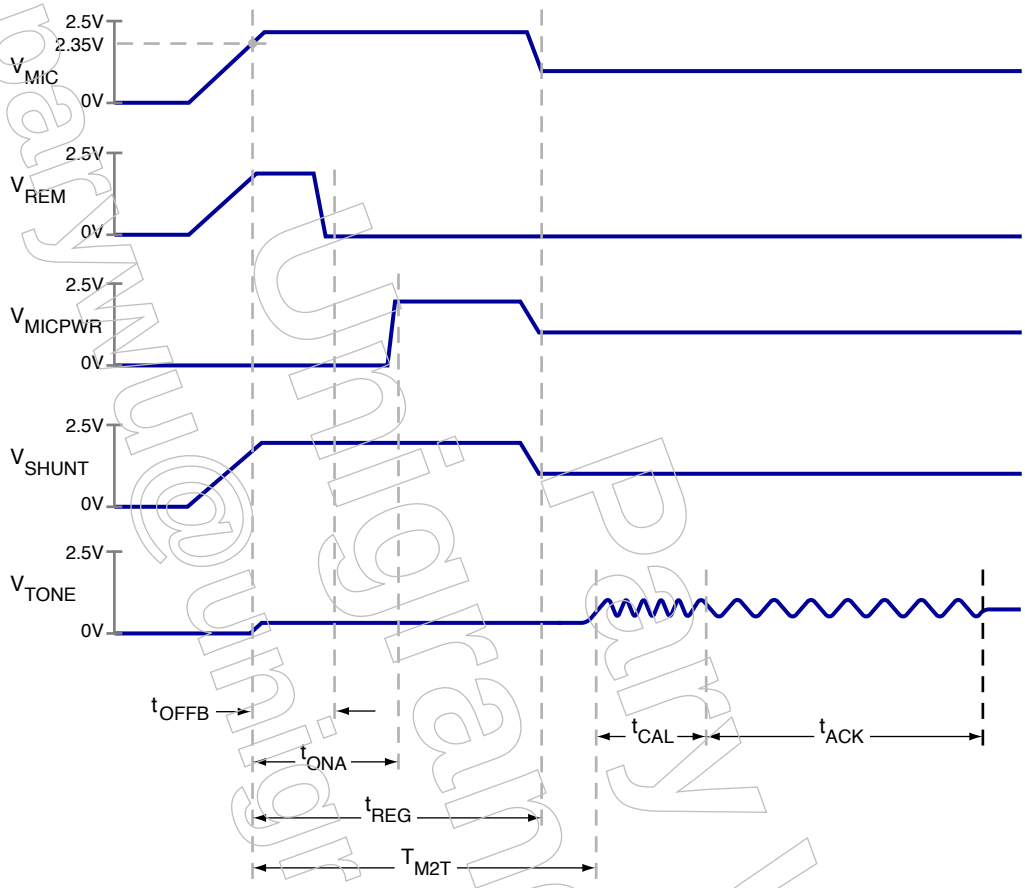
When the transmitter chip detects a voltage greater than 2.35 V at the MIC pin, it enters tone mode. With a microphone biased and in use, the switch-resistor network used for button mode would cause large DC level shifts in the bias voltage. Such shifts would result in unwanted audible clicks or pops or would cause de-biasing of the microphone. To prevent this problem, when the transmitter chip enters tone mode it disconnects the switch-resistor network from the microphone bias line, enables the microphone via the FET switch, and engages the tone generation circuit shown in Figure 27-5 (page 397).

In tone mode the transmitter chip has two functions. First, it turns on the MEMS microphone by forcing a FET switch to ground. Second, it detects button events and places a discrete tone sequence onto the microphone bias line. The tone frequencies in each sequence are unique to each button press. The controller detects the tones on the bias line and determines the corresponding button event.

The transmitter chip's startup timing when it enters tone mode is shown in Figure 27-6 (page 399). Values for the timing parameters are given in Table 27-6 (page 395).

Jay Wu (jay@chipfind.com.tw) ChipFind LTD

Figure 27-6 Transmitter Chip Startup Timing

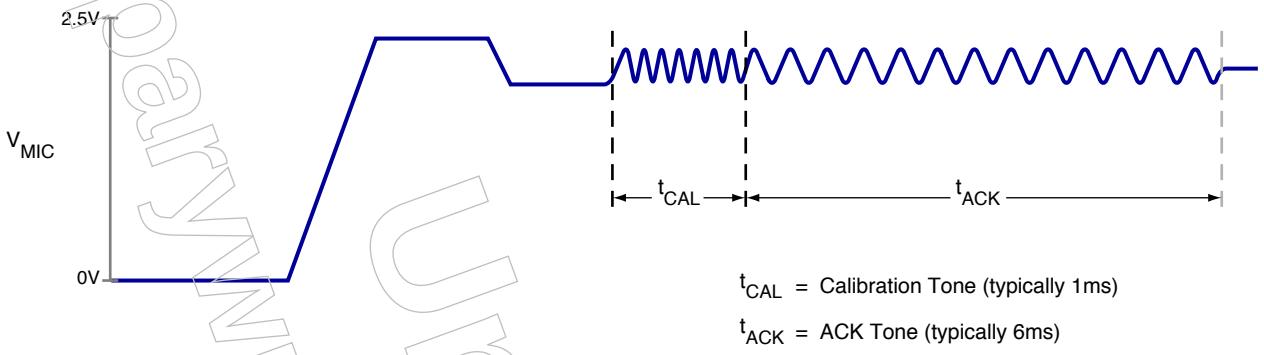


The tone mode startup sequence is as follows:

1. Upon detecting V_{MIC} > 2.35 V, the switch connecting the MIC and REM pins together is opened after time t_{OFFB} (see Figure 27-6 (page 399) and Table 27-6 (page 395)).
2. After a delay of t_{REG} after V_{MIC} > 2.35 V, the SHUNT pin and the MICPWR pins are shorted. The microphone is enabled by turning on the FET switch through the MICPWR pin.
3. Once the noise prevention process has settled, the transmitter chip sends a preset acknowledge (ACK) tone sequence.
4. The controller detects the ACK sequence (see Figure 27-7 (page 400)) and authenticates the presence of the transmitter chip.

Patent Pending Unregulated and Uncertified

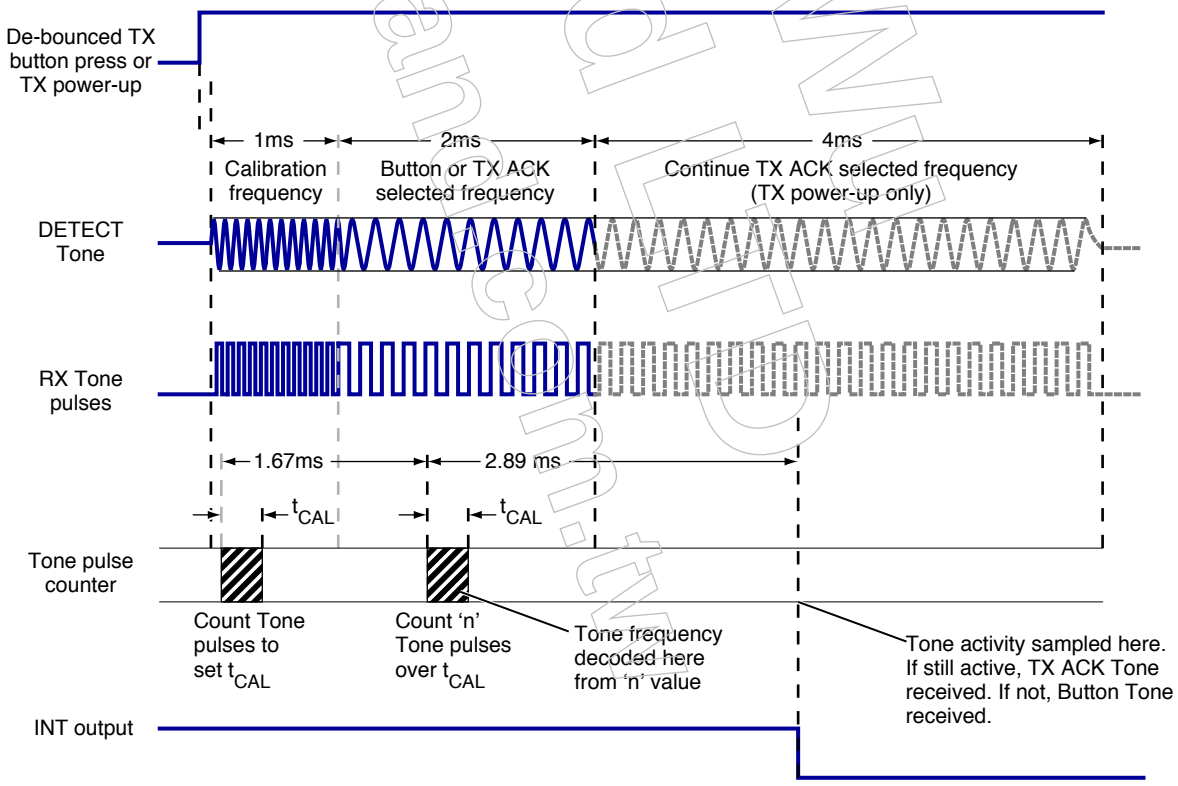
Figure 27-7 Tone Mode ACK Sequence



The tone generation circuit of the transmitter chip internally detects each button press and sends a high frequency tone sequence between 99 kHz and 300 kHz. The high frequency tone sequence is unique to each button. The controller detects the frequency of each tone and translates it into a predetermined button event. A button release has a different frequency than a button press.

For accuracy, the transmitter chip sends two tones for each button press as shown in Figure 27-8 (page 400). The first tone, lasting 1 ms, is a calibration frequency and the second, lasting 2 ms, is the unique frequency for the selected button. The ratio of these two frequencies is calculated and translated into button press information. This provides a very accurate result that is independent of clock frequency variation.

Figure 27-8 Tone Transmit/Decode Method



Unreleased Digital Content

The transmitter chip remains in tone mode until the MIC pin is pulled below 0.8 V. When power recovers, the transmitter chip enters button mode or tone mode depending on the voltage detected at the MIC pin.

27.3 Button Detection Circuitry Usage

The circuits in the accessory that support these components must be those shown in Figure 27-9 (page 402) and Figure 27-10 (page 402). The nominal values of the components shown in these schematics are given in Table 27-9 (page 403).

These circuits are designed to produce a tone amplitude between the microphone bias line and the microphone return, at the end of a cable 1 meter long, of at least 30 mV peak-to-peak into a 2000-ohm load. If necessary, the value of R3 must be adjusted to achieve this result. Figure 27-10 (page 402) shows how a voltage on the Microphone Power line from the transmitter chip enables the MEMS microphone chip through Q1. It also shows components R7, C4, and R8, which control the microphone's frequency response. The equation that determines the values of these components is given in Button Detection Circuitry Adjustments (page 403).

Figure 27-9 (page 402) and Figure 27-10 (page 402) are two parts of one circuit. The two Microphone Return lines shown in these sub-circuits must be connected at the component locations. Their common return line and the return lines for each of the two drivers must then be routed separately through the cable that goes to the Apple device, being tied together only at the headphone connector. This configuration is required to minimize crosstalk between the separate driver channels and the microphone.

Note: With the exception of the MEMS digital microphone listed in Table 27-9 (page 403), symbol U2, components of equal or better specifications may be substituted for the components called out below.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 27-9 Transmitter Circuit

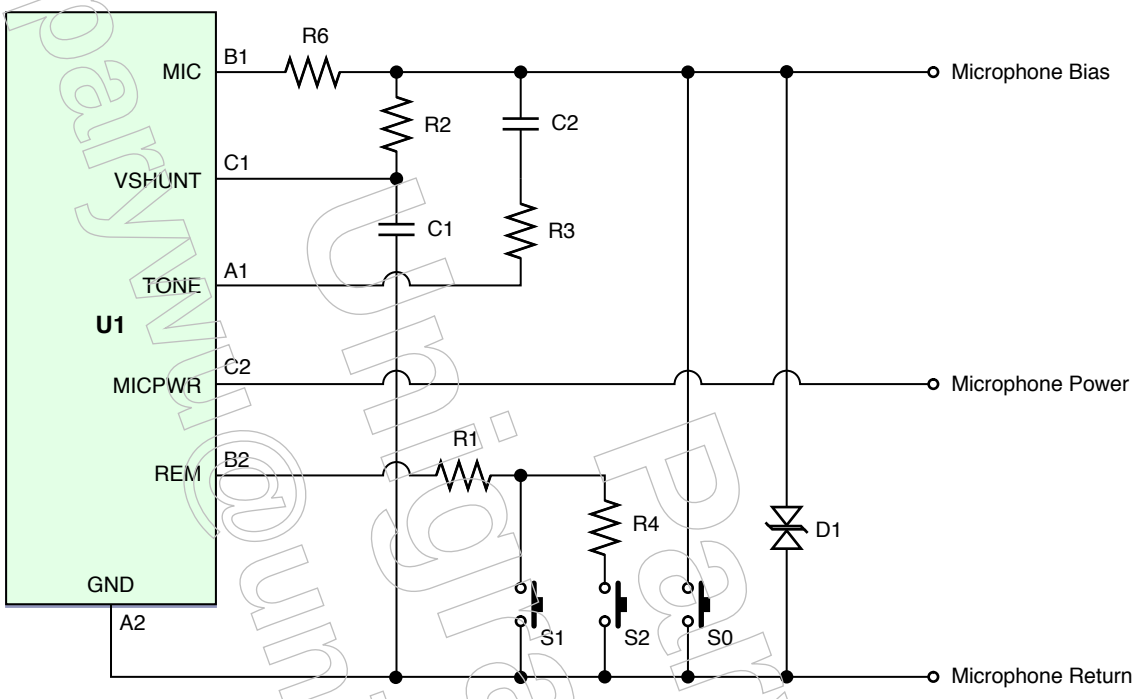
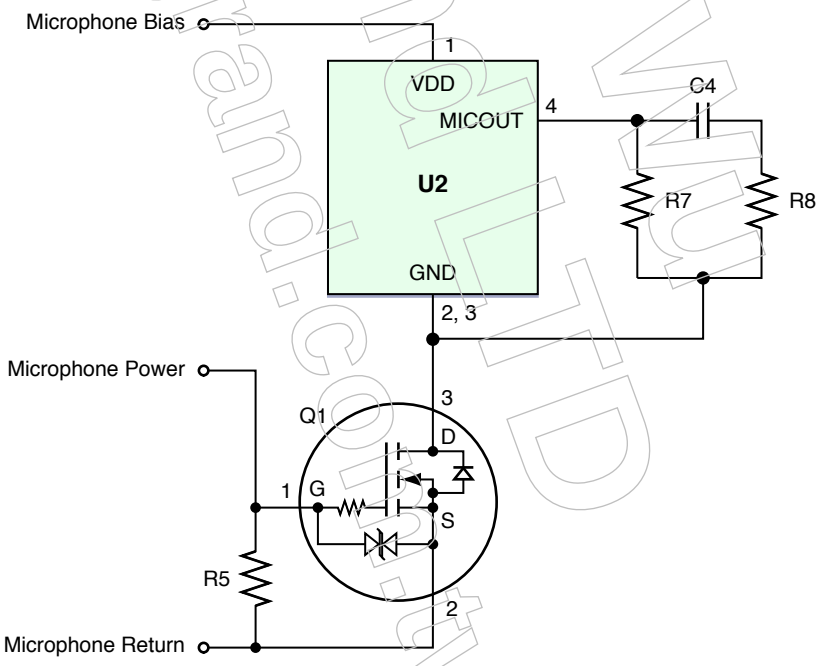


Figure 27-10 Microphone Circuit



Party Unplanned Ltd
Party@unplanned.co.nz

Table 27-9 Transmitter Circuit Components

Symbol	Description	Notes
C1	Capacitor, 0.1 µF ±10%, 6.3 V	
C2	Capacitor, 220 pF ±5%, 25 V	Ceramic
C4	Capacitor, 2.2 µF ±10%, 6.3 V	
D1	ESD protection diode, 5 pF, 6.1 V	ST Micro ESDALC6V1-1BU2; install as close to chip pin B1 as possible
Q1	MOS field-effect transistor	CEDM 7001
R1	Resistor, 6.81 kΩ ±0.5%, 1/20 W	
R2	Resistor, 2 kΩ ±1%, 1/20 W	
R3	Resistor, 1.2 kΩ ±0.5%, 1/20 W	
R4	Resistor, 2.61 kΩ ±0.5%, 1/20 W	
R5	Resistor, 887 kΩ ±1%, 1/20 W	
R6	Resistor, 49.9 Ω +0.2%/-1%, 1/20 W	Must not exceed 50 Ω.
R7	Resistor, 17.4 kΩ ±1%, 1/20 W	
R8	Resistor	see Table 27-10 (page 404)
S0	Dome switch	Push-to-talk; must not exceed 20Ω when closed.
S1	Dome switch	Volume down; must not exceed 20Ω when closed.
S2	Dome switch	Volume up; must not exceed 20Ω when closed.
U1	Headset interface transmitter chip	Provided by Apple
U2	MEMS digital microphone	see Table 27-10 (page 404)

27.3.1 Button Detection Circuitry Adjustments

The accessory must use one of the MEMS digital microphone components and its associated R8 resistor value specified in Table 27-10 (page 404).

Party Wu
Digi-Qualified.com.tw

Table 27-10 Approved transmitter circuit MEMS digital microphone components

MEM Digital Microphone (U2)	Resistor (R8)
Knowles SPQ2409HE5H-PB	1.2 kΩ ±1%, 1/20 W
Knowles SPY1824LR5H-B	1.6kΩ ±1%, 1/20 W
Goertek S15OB383-002	1.91kΩ ±1%, 1/20 W

The values of some of the components listed in Table 27-9 (page 403) may be adjusted to optimize the performance of the headphone accessory, using these formulas:

- **High-pass filter corner frequency in Hertz** ≈ $1/(2\pi \cdot R_8 \cdot C_4)$, where R_8 is the value of resistor R8 in ohms and C_4 is the value of capacitor C4 in Farads. This formula assumes that the value of R7 is greater than the value of R8.
- **System sensitivity at 1 Pascal in Volts** = $(M_0/R_8) \cdot R_2$, where M_0 is the microphone sensitivity in Volts per Pascal, R_8 is the value of resistor R8 in ohms, and R_2 is the value of resistor R2 in ohms in parallel with 1.05 kΩ.
- **Maximum excursion of the microphone in Volts** = $(1/R_7) \cdot R_2$, where R_7 is the value of resistor R7 in ohms, and R_2 is the value of resistor R2 in ohms in parallel with 1.05 kΩ.

Note: If the microphone bias voltage drops below 1.6 V, the transmitter chip will begin to fail and the microphone chip may produce indeterminate outputs.

27.4 Test Procedures

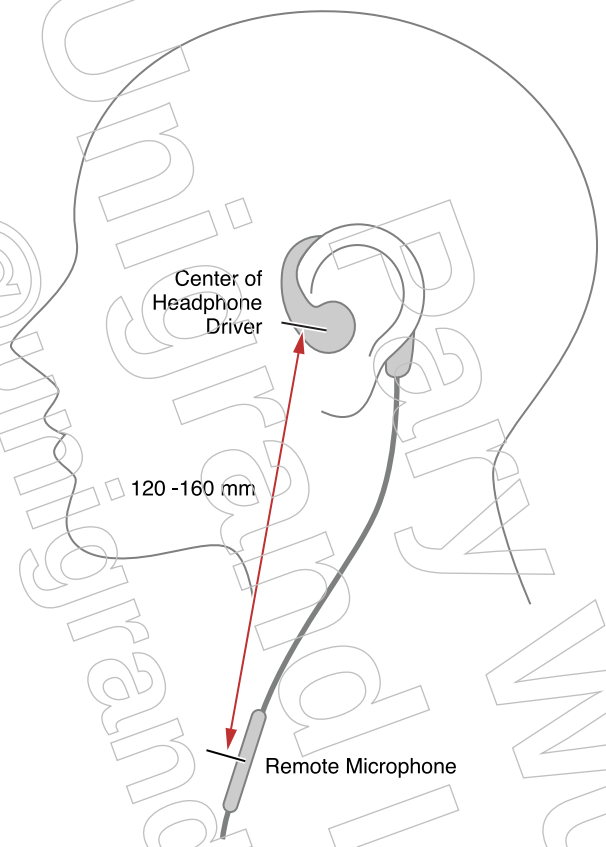
27.4.1 Product Design

1. Verify that the remote microphone is located 120-160 mm from the center of a headphone driver.
2. See Test Procedures (page 379).

Apple Confidential
Authorized Use Only
Quanta Computer Inc. ITD

Note: If the headphone cabe is intended to loop around the ear (see Figure 27-11 (page 405)), measure the distance as if the accessory is being worn.

Figure 27-11 Headphones Looped Around Ear Test Setup



27.4.2 Electrical

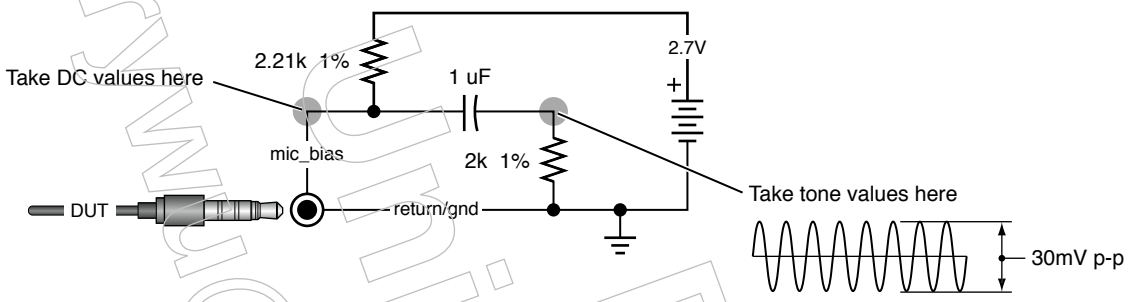
The following items are required:

- A precision power supply
- 4-pin headphone jack, 3.5 mm diameter, such as Digikey SJ-43515RS-SMT
- An oscilloscope with measuring capabilities
- A 2.21 kΩ 1% resistor
- A 2 kΩ 1% resistor
- A 1 µF capacitor

Ray Wu
Unigrand LTD
rayWu@unigrand.com.tw

These items must be connected as shown in Figure 27-12 (page 406)

Figure 27-12 Headphone Remote and Mic Test Setup



Testing must demonstrate the following:

- All tones must have a peak-to-peak amplitude of at least 30 mV as measured across the 2 kΩ load.
- Labeled oscilloscope pictures must be generated for all button press tones, button release tones, and acknowledgment tones, showing their amplitudes. The button release frequency of 99 kHz must have an amplitude of at least 30 mV peak-to-peak.
- The mic_bias point static DC voltage must be between 1.85 and 2.0 V. A labeled oscilloscope picture must be generated showing that the voltage at the mic_bias point is stable and within specification within 50 ms after power-up.
- Set the power supply in the test circuit to 2.0 V to place the transmitter in button mode. An oscilloscope picture must be generated to demonstrate DC shifts occurring during button presses with static DC mic_bias voltages as shown in Table 27-11 (page 406).

Table 27-11 Headphone Remote and Mic Expected mic_bias Voltages in Button Mode

Switch Closure	Voltage
S0	0.000 V ±1%
S1	1.510 V ±1%
S2	1.603 V ±1%

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

28. HID (Human Interface Device)

Some Apple devices can accept input from Human Interface Device (HID) accessories, such as external keyboards, and also send HID reports to those accessories. This capability is then made available system-wide for all apps on the device or certain features built into the device system software. If an accessory is designed to provide human input events to a specific third-party app, it should consider implementing the External Accessory Protocol feature instead.

The HID feature can be implemented both over an iAP2 control session and over native transports that support HID classes or profiles, such as USB Host. Implementing HID over native transports may make sense when using off the shelf chipsets with built-in support for HID, but HID over the iAP2 control session is recommended when transport independence and/or Apple device-powered operation is desired (because HID over iAP2 can use the Serial transport). Also, multiple HID report descriptors can be registered from one accessory over one iAP2 control session and the Apple device will route HID reports from each HID device appropriately.

The HID feature can also be implemented over the CarPlay communication protocol.

28.1 HID Requirements

Accessories supporting the HID feature over iAP2 or over native transports must comply with the following requirements:

- The accessory must identify at least one Table 52-17 (page 584) or Table 52-22 (page 586).
- Each identified HID component must have a unique `HIDComponentIdentifier`.
- Each identified HID component must declare one intended function.

Accessories supporting the HID feature over the CarPlay communication protocol must comply with the following requirements:

- User events must be represented as Human Interface Device (HID) usage reports and must be transported using the CarPlay communication protocol.
- Accessories must not use iAP2 to transport HID events while using CarPlay, with one exception: accessories supporting HID Media Playback Remote may send those commands (except for Voice Control / Siri) over iAP2 (see HID Media Playback Remote (page 456)).

Apple Confidential Information

- When supporting CarPlay, the Voice Command / Siri usage must always be transported using the CarPlay communication protocol. To send HID events over CarPlay use the command hidSendReport as described in hidSendReport (page 321).

All accessories supporting the HID feature must comply with the following requirements:

- The accessory must be capable of generating and receiving all HID usages declared in the component's HID descriptor.
- Unless otherwise specified, the accessory's declared HID usages must map directly to physical or virtual control surfaces on a 1:1 basis. For example, a button labeled "Play/Pause" must send a Play/Pause HID usage and not "Play" or "Pause" usages.
- Unless otherwise specified, physical or virtual control surfaces that generate accessory HID usage reports must be labeled with appropriate iconography or text that corresponds to the resulting Apple device behavior. For example, a Play/Pause button must be labeled with the text 'Play/Pause' or a Play/Pause icon.
- Unless otherwise specified, one accessory HID usage report must be sent in response to each direct user action on the corresponding physical or virtual control surface. For example, when the user presses a button, one 'button pressed' usage report must be sent, and a separate 'button released' usage report must be sent when the user releases the button.
- Accessories must only send HID reports for changes in physical or virtual control surfaces registered in the HID descriptor.
- Each HID report must contain the correct number of bytes as described in its corresponding HID report descriptor.
- The accessory must not anticipate or assume corresponding state changes in the Apple device after sending HID usages. For further explanation, see Presentation of Apple Device Updates (page 49).
- Accessories must not send a HID report if there has not been any change in the state of the control surface(s). For example, the accessory must never generate a "Play/Pause" event without the user pressing a dedicated "Play/Pause" button.

28.1.1 HID Report Descriptor Requirements

- When padding packets to align within a byte boundary, each Main item tag (Input, Output, or Feature) must be marked constant. Padding bits should be set to 0.
- When defining Variable type Input/Output fields, the Report Count number must match the number of Usages specified.

Confidential
Apple Inc.
Do not distribute

28.1.2 HID over iAP2 Requirements

If implementing HID over iAP2, an accessory must send or receive the following iAP2 control session messages:

- StartHID (page 599)
- AccessoryHIDReport (page 600)

An accessory may also send or receive the following iAP2 control session messages:

- DeviceHIDReport (page 600)
- StopHID (page 600)

28.1.3 HID Native Transport (USB Host Mode) Requirements

If implementing HID over USB Host Mode transport, the accessory must comply with the *Device Class Definition for Human Interface Devices* (available from the USB-IF). Additionally, the accessory must declare a USB Host Mode transport component during identification.

28.2 HID Usage

28.2.1 HID over iAP2 Usage

To use HID over an iAP2 control session, the accessory must send a StartHID (page 599) message to the device. The `HIDComponentIdentifier` parameter must correspond to a previously identified Table 52-17 (page 584). Each Table 52-17 (page 584) must be registered with a separate StartHID (page 599) message. Similarly, to stop usage of a particular HID report descriptor, a separate StopHID (page 600) message must be sent for that component.

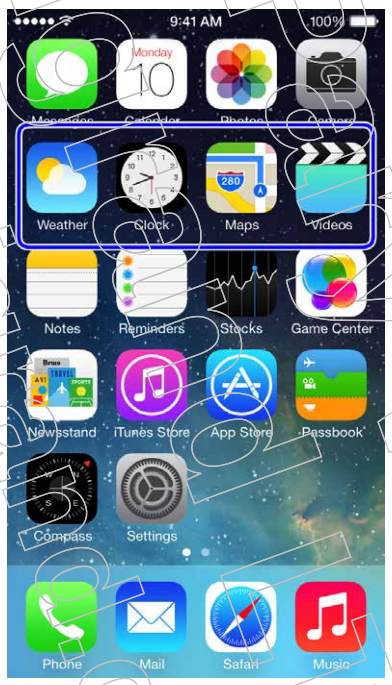
In between the StartHID (page 599) and StopHID (page 600) pairs for a particular component, the accessory may send one or more AccessoryHIDReport (page 600) messages to the device. Each AccessoryHIDReport (page 600) message must reference a specific HID component in addition to the HID report itself. Similarly, the Apple device may send one or more DeviceHIDReport (page 600) messages to the accessory with the same parameters.

Confidential Property of Unigrand Ltd.
Reviewing@unigrand.com.tw

29. HID Assistive Switch Control

The Assistive Switch Control feature permits users to operate Apple devices using one or more switches. Accessory developers should consider supporting Assistive Switch Control if the accessory is intended for use by users with special needs who can see the display and manipulate simple switch inputs. If the user cannot see the display, consider supporting the VoiceOver feature instead.

Figure 29-1 HID Assistive Switch Control



29.1 HID Assistive Switch Control Requirements

Accessories that implement HID Assistive Switch Control components must support the HID (Human Interface Device) (page 407) feature and comply with all the requirements listed in HID Requirements (page 407).

A maximum of one HID component in an accessory may declare itself to be an Assistive Switch Control, and the accessory must be intended for use by a user with special needs.

Assistive Switch Controls must have at least one button. Two are recommended. Up to 16 are supported.

- Button 1 must be labeled "Select" or its localized equivalent.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- If present, button 2 must be labeled "Next" or its localized equivalent.
- If present, buttons 3-16 must not be labeled.

29.2 HID Assistive Switch Control Usage

If only Button 1 is present, the Apple device will automatically configure the Assistive Switch Control feature to use that button to select an onscreen element and scan through onscreen elements automatically.

If both Button 1 and Button 2 are present, the Apple device will automatically configure the Assistive Switch Control feature to use the first button to select an onscreen element and only scan through onscreen elements when the second button is pressed.

If more than two buttons are present, Button 1 and Button 2 are automatically configured and additional buttons may be mapped to other actions by the user.

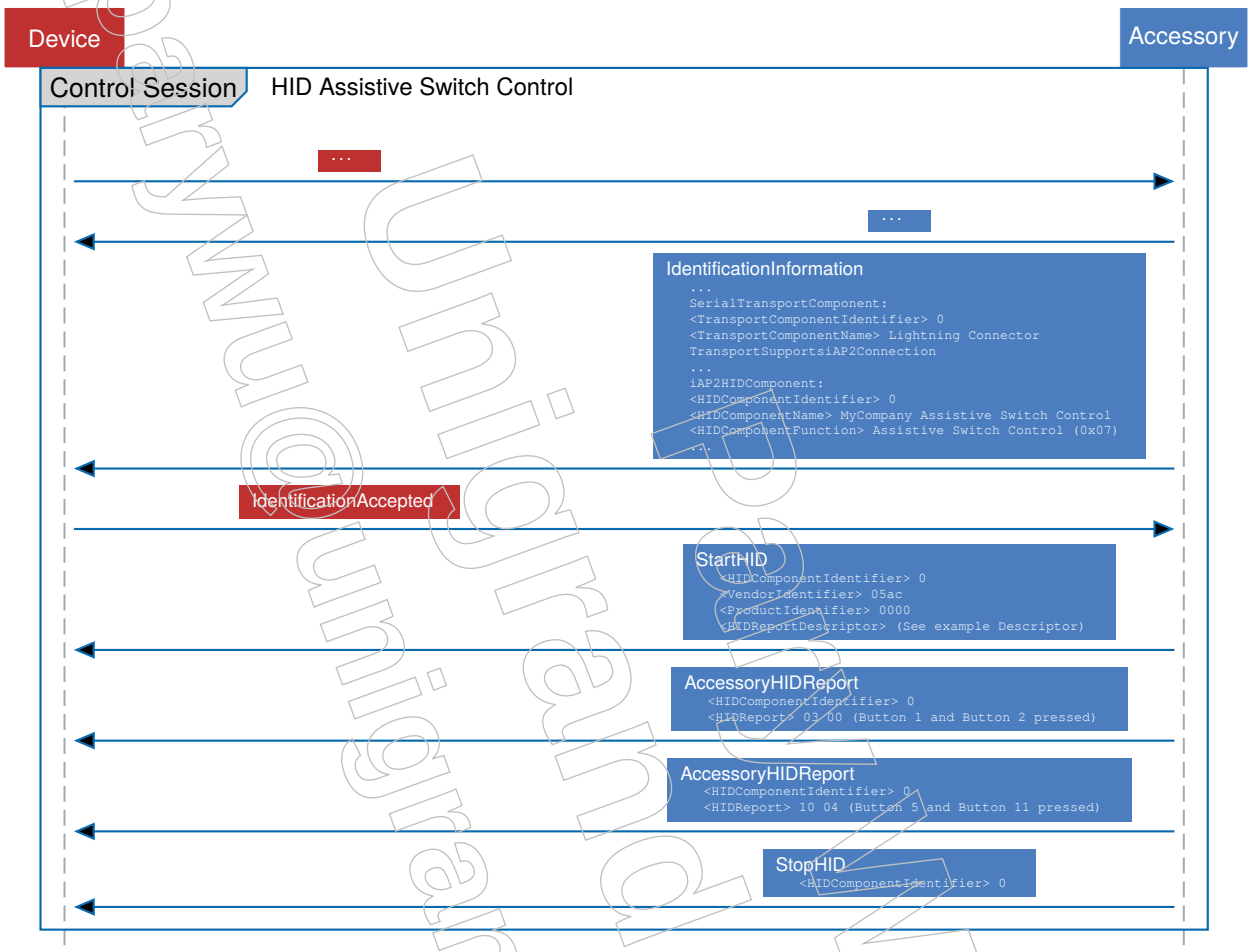
29.3 HID Assistive Switch Control Examples

29.3.1 Assistive Switch Control HID Report Descriptor

```
USAGE PAGE (Button)                 05 09
USAGE (Button 1)                    09 01
COLLECTION (Application)            A1 01
  USAGE (Button 1)                  09 01
  COLLECTION (Physical)             A1 00
    USAGE MINIMUM (Button 1)        19 01
    USAGE MAXIMUM (Button 16)       29 10
    LOGICAL MINIMUM (0)             15 00
    LOGICAL MAXIMUM (1)             25 01
    REPORT COUNT (16)               95 10
    REPORT SIZE (1)                 75 01
    INPUT (Data,Var,Abs)            81 02
  END COLLECTION (Physical)         C0
END COLLECTION (Application)        C0
```

Payway Global LTD
payway@payway-global.com.tw

29.3.2 Assistive Switch Control Usage Example



developer@uniquand.com.tw Unique LTD

30. HID AssistiveTouch Pointer

30.1 HID AssistiveTouch Pointer Requirements

Accessories that implement HID AssistiveTouch Pointer components must support the HID (Human Interface Device) (page 407) feature and comply with all the requirements listed in HID Requirements (page 407). Additionally, they must support the AssistiveTouch (page 214) feature.

A maximum of one HID component in an accessory may declare itself to be an AssistiveTouch pointer. That component must be a physical pointing device intended for use by a user with special needs.

The HID report descriptor for an AssistiveTouch pointer must declare support for the HID Generic Desktop Page and the Mouse usage.

The following requirements apply to the generation of HID mouse reports from the accessory:

- All x and y movements must be reported in increments of 1, proportionally scaled to the physical movement of the user. If the accessory is a joystick, for example, then a small movement of the joystick must report a movement delta of 1, but a large movement of the joystick must report a larger movement delta.
- The accessory must send repeated HID pointer movement reports at a constant rate appropriate for the accessory. The accessory must not perform its own scaling of the report rate; the AssistiveTouch feature uses its own speed scaler setting for this purpose. If no movement has taken place, the accessory must send a movement report of 0 in both x and y directions.
- The accessory must generate HID reports for two buttons, one for a touch event and the other for a contextual menu trigger. Both button down and button up reports must be sent individually and must match actual user actions on the accessory. When the user presses on the first button, a button1 'down' report must be sent, and button1 events must not be sent until the user releases the button, after which a button1 'up' report must be sent.
- The accessory must start sending HID reports to the Apple device as soon as the Apple device sends a notification to the accessory indicating that the AssistiveTouch cursor has been enabled.
- The accessory must cease sending HID reports to the Apple device as soon as the Apple device sends a notification to the accessory indicating that the AssistiveTouch cursor has been disabled.
- The accessory must be capable of interleaving pointer movement reports with button up and down reports. The accessory must let the user hold a button down and move the pointer at the same time.

farwell@unitedtechpatents.com Confidential Unauthorized Distribution Prohibited

See Appendix E.10: Report Descriptor(Mouse) in *Device Class Definition for Human Interface Devices 1.11* (available from the USB-IF) for a sample HID descriptor.

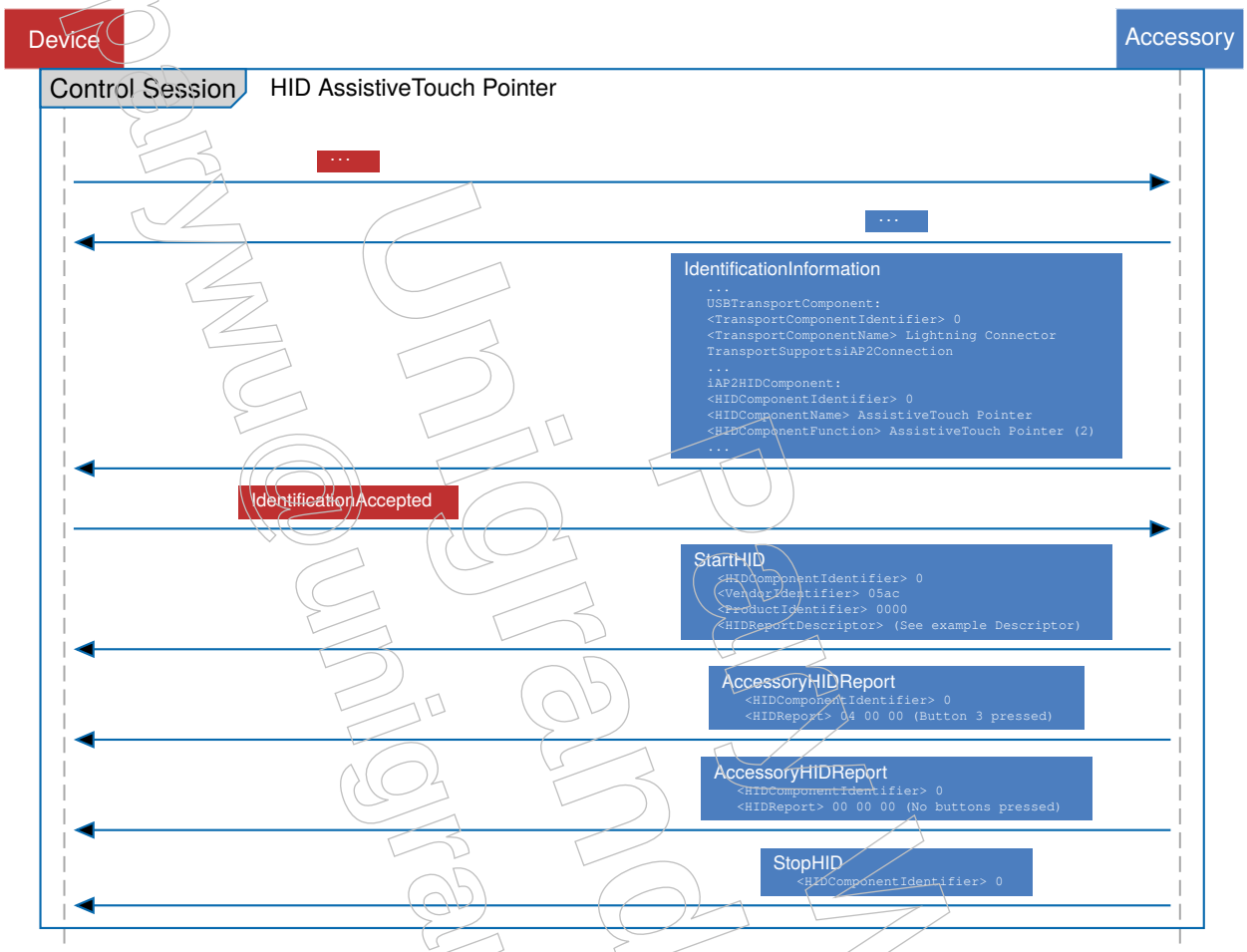
30.2 HID AssistiveTouch Pointer Examples

30.2.1 AssistiveTouch HID Report Descriptor

```
USAGE_PAGE (Generic Desktop)        05 01
USAGE (Mouse)                       09 02
COLLECTION (Application)            A1 01
  USAGE (Pointer)                   09 01
  COLLECTION (Physical)             A1 00
    USAGE_PAGE (Button)             05 09
    USAGE_MINIMUM (Button 1)        19 01
    USAGE_MAXIMUM (Button 3)        29 03
    LOGICAL_MINIMUM (0)             15 00
    LOGICAL_MAXIMUM (1)             25 01
    REPORT_COUNT (3)                95 03
    REPORT_SIZE (1)                 75 01
    INPUT (Data,Var,Abs)            81 02
    REPORT_COUNT (1)                95 01
    REPORT_SIZE (5)                 75 05
    INPUT (Cnst,Ary,Abs)            81 01
    USAGE_PAGE (Generic Desktop)    05 01
    USAGE (X)                       09 30
    USAGE (Y)                       09 31
    LOGICAL_MINIMUM (-127)          15 81
    LOGICAL_MAXIMUM (127)           25 7F
    REPORT_SIZE (8)                 75 08
    REPORT_COUNT (2)                95 02
    INPUT (Data,Var,Rel)            81 06
  END_COLLECTION                    C0
END_COLLECTION                      C0
```

Party Wu Unleashed LTD
partywu@unleashed.com.tw

30.2.2 AssistiveTouch Usage Example



Copyright © Unigrand LTD Vu
vu@unigrand.com.tw

31. HID Game Controller

iOS apps may take advantage of accessories that implement one or more HID Game Controller components via the Game Controller framework. Three controller component types are defined in this specification:

- Form-Fitting Standard Gamepad
- Form-Fitting Extended Gamepad
- Non Form-Fitting Extended Gamepad

Information on how to to create iOS apps that make use of HID Game Controller accessories can be found at https://developer.apple.com/library/ios/documentation/ServicesDiscovery/Conceptual/GameControllerPG/Introduction/Introduction.html.

31.1 HID Game Controller Requirements

Accessories that implement HID Game Controller components must support the HID (Human Interface Device) (page 407) feature and comply with all the requirements listed in HID Requirements (page 407).

Form-fitting controllers must physically enclose the Apple device (in landscape orientation) and allow the user's thumbs to reach the Apple device's touchscreen. They must also comply with the requirements set forth in Form-Fitting Accessories (page 153) and Cases (page 336). Specifically, the device's touchscreen must not be occluded in any way.

An accessory must implement only one form-fitting controller component.

If an accessory has multiple physical configurations that would allow it to be both a form-fitting gamepad and a non form-fitting gamepad, it must terminate the iAP2 connection, re-connect, and appropriately identify itself to the Apple device whenever it enters a new physical configuration. Such accessories must not always claim to be a non form-fitting gamepad.

Partner Confidential - Apple Inc.

Figure 31-1 Form-Fitting Standard Gamepad Sample

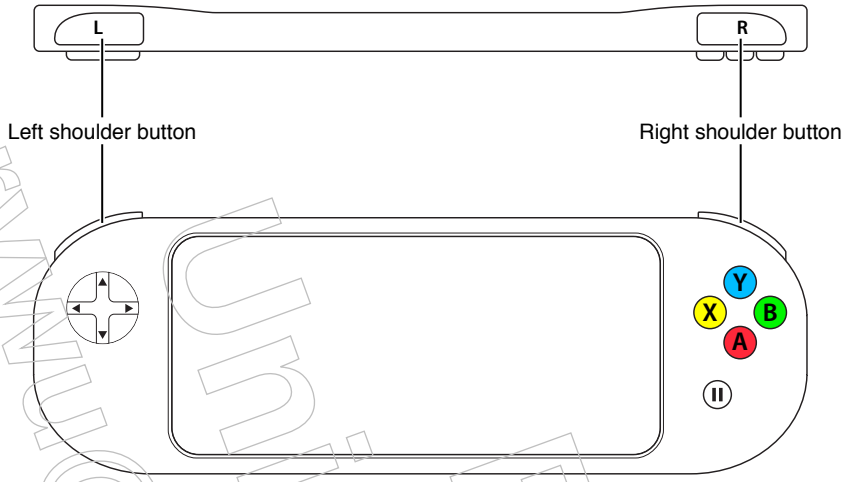
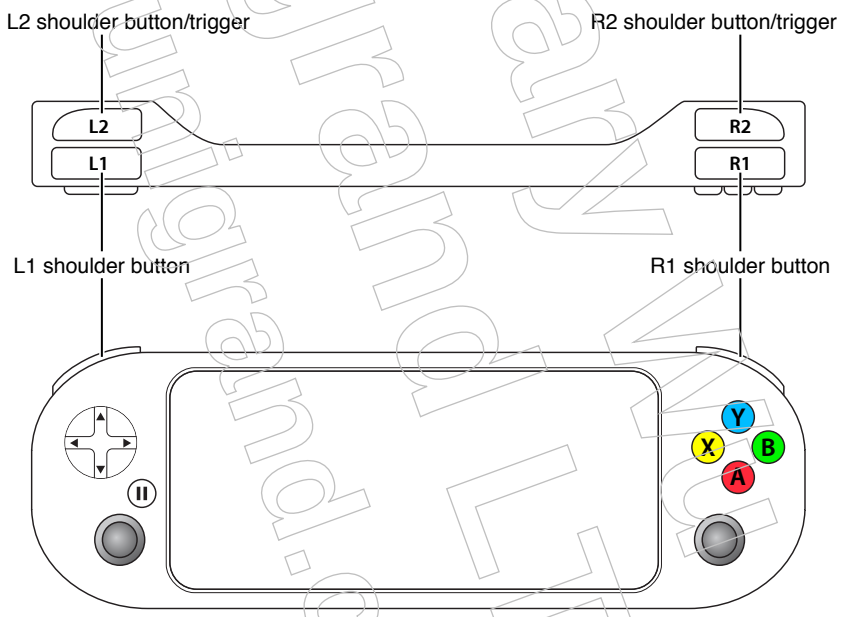


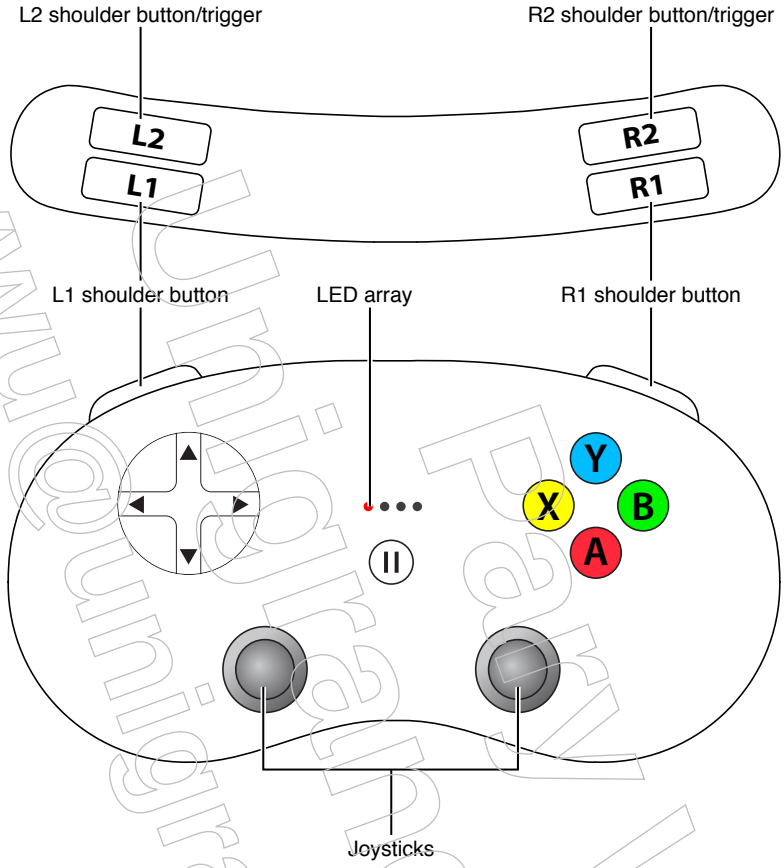
Figure 31-2 Form-Fitting Extended Gamepad Sample



Non form-fitting controllers must not physically enclose the Apple device and must comply with Extended Gamepad requirements. One accessory may implement up to 4 physically separate non form-fitting controller components.

paywin@uniland.com.tw Uniland Paywin Ltd.

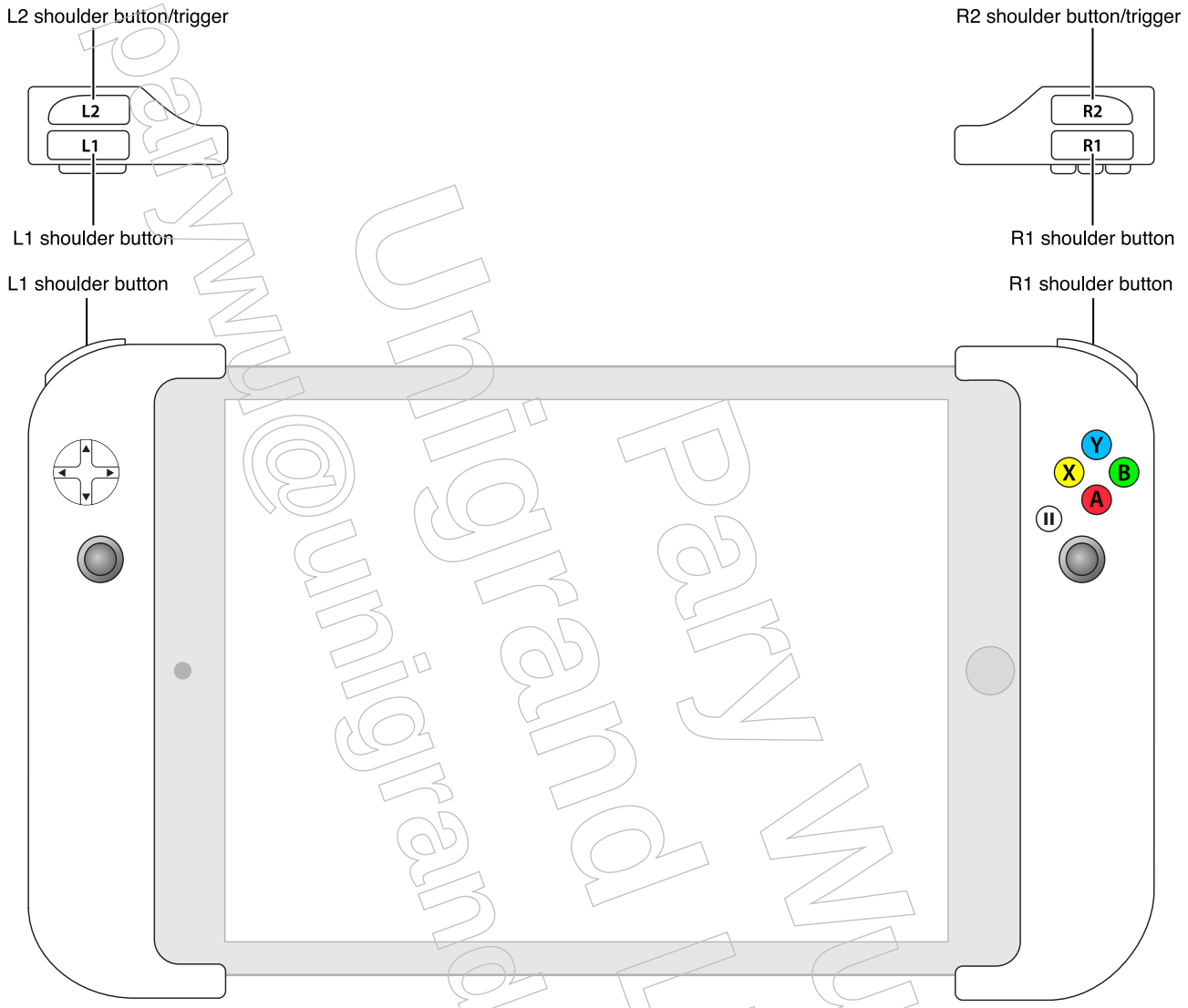
Figure 31-3 Non Form-Fitting Extended Gamepad Sample



Form-fitting controllers can be made for any Apple device supporting this feature such as the iPad mini (see Figure 31-4 (page 419)).

Party Wu Digital Pro LTD
partywu@unigrand.com.tw

Figure 31-4 iPad mini Form-Fitting Extended Gamepad Sample



Each HID component that declares itself as a game controller must implement a prescribed configuration of physical control surfaces. All HID reports sent from the controller must occur in response to direct user action, i.e. depressing one of the control surfaces.

Controllers must not implement additional physical control surfaces other than those required by the Standard or Extended Gamepad definitions except for the following:

- Charging on/off switch for a controller with integrated battery that provides power to the Apple device.
- On/off button for a self-powered controller.
- Button to temporarily display charge level for a controller with integrated battery using the player indicator LEDs. See LED Array (page 429).
- Button to initiate Bluetooth pairing. See Bluetooth Button (page 429).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Controller accessories must be serialized, i.e. the *SerialNumber* parameter in their IdentificationInformation (page 579) message must be unique for a given *Manufacturer* and *ModelIdentifier* pairing.

Controllers must sample the state of all control surfaces at a rate of 120 Hz.

When the state of any control changes compared to the previously taken sample, the controller must send a HID report to the Apple device. Controller HID reports:

- Must be complete and contain the state of every control surface, changed or not.
- Must not be sent if there is no change in control state compared to the previous sample.
- Must not be sent to the Apple device at a rate exceeding 120 Hz.

Controllers should allow users to adopt comfortable postures during normal use, namely grip, reach, and finger placement on input buttons. Children and females with small hands (5th percentile), as well as large males (95th percentile) should be considered during the design process. Apple recommends taking advantage of resources such as AnthroKids (http://ovrt.nist.gov/projects/anthrokids/), the U.S. Army Anthropometry Survey (http://mreed.umtri.umich.edu/mreed/downloads.html#ansur), and the Civilian American and European Surface Anthropometry Resource (http://store.sae.org/caesar/index.htm).

A user must be able to accurately hold a control at any position.

All controllers must be capable of field-deployable firmware updates from Apple devices running iOS and/or Macintosh computers running OS X.

Controllers must not implement alternate operating modes that emulate keyboards or any other type of Human Interface Device besides a gamepad.

31.1.1 Standard Gamepad

To support the Standard Gamepad definition, a controller must be form-fitting and implement the following control surfaces:

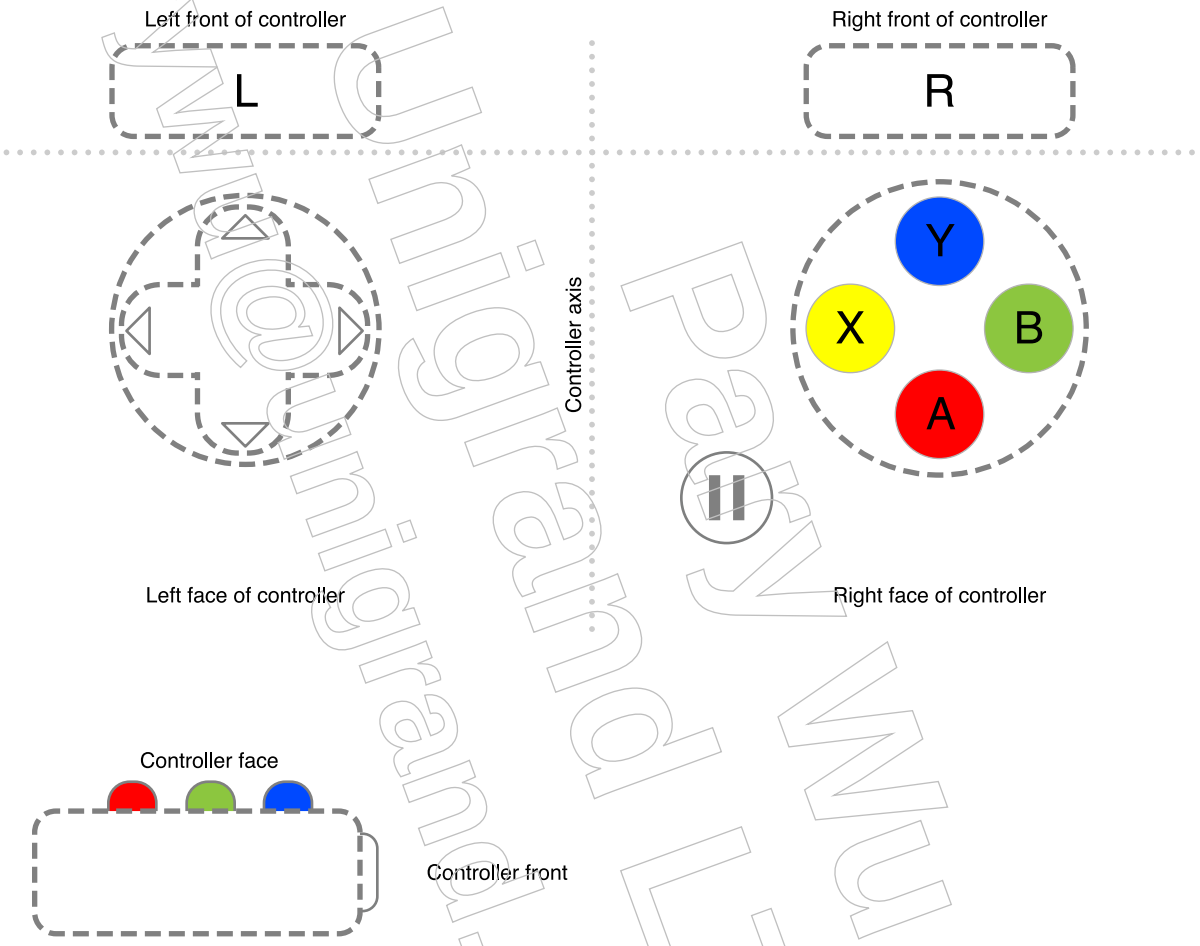
- Directional pad
- Face button group
- Pause button
- Left shoulder button (L)
- Right shoulder button (R)

The layout and labeling of the control surfaces must match that of Figure 31-5 (page 421) except in two cases:

Licensed to Bluetooth SIG, Inc.

- Form-fitting standard gamepads may position the pause button on either the left or right side of the controller axis. The right side is recommended.
- Form-fitting standard gamepads may implement a LED array as defined in LED Array (page 429).

Figure 31-5 HID standard gamepad control layout



31.1.2 Extended Gamepad

Some apps may take advantage of the additional controls available on accessories that support the Extended Gamepad definition. These games will also all work with the Standard Gamepad, but some in-game features may not be accessible from the gamepad.

To support the Extended Gamepad definition, an accessory must implement the following control surfaces:

- Directional pad
- Face button group
- Pause button
- Top left shoulder button (L1)

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

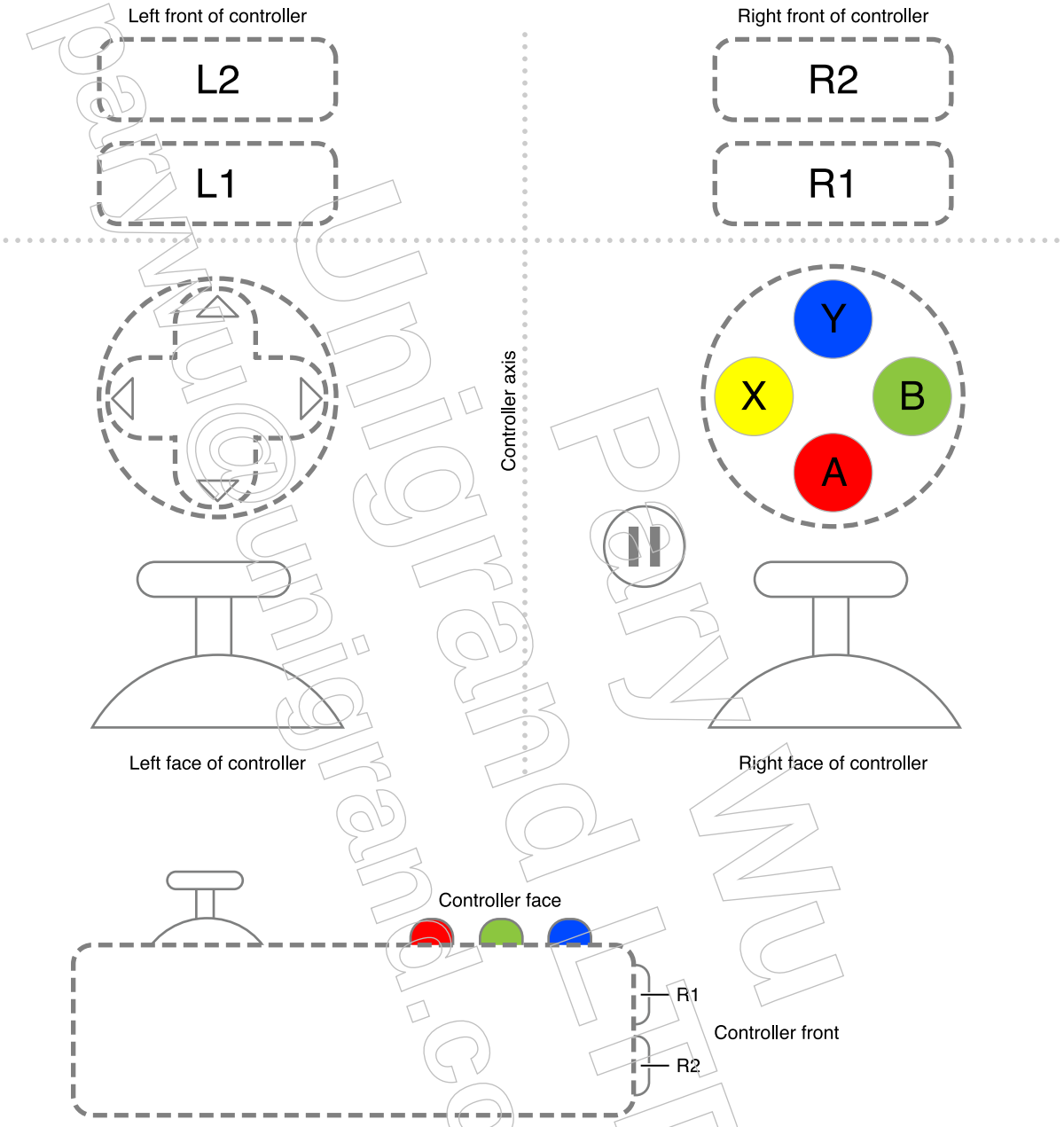
- Top right shoulder button (R1)
- Bottom left shoulder button/trigger (L2)
- Bottom right shoulder button/trigger (R2)
- Left joystick
- Right joystick

The layout and labeling of the control surfaces must match that of Figure 31-6 (page 423), except in four cases:

- Form-fitting extended gamepads may position the pause button on either the left or right side of the controller axis. The right side is recommended.
- Non form-fitting extended gamepads should position the pause button on the controller axis.
- Extended gamepad joysticks may be swapped vertically with their corresponding face button group or directional pad.
- Non form-fitting extended gamepads must also implement a LED array as defined in LED Array (page 429). Form-fitting extended gamepads may implement the LED array.

Property of Unigrand LTD
Terry Wu
terrywu@unigrand.com.tw

Figure 31-6 HID extended gamepad control layout



31.1.3 Switches and Displacement Triggers

Game controller switches are either analog (pressure sensitive) or digital. They may be combined with buttons or triggers to create controller control surfaces.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Pressure sensitive switches and displacement triggers must report status using an unsigned *n*-bit integer, where *n* is 8 or more. They must report a value of 0 when not depressed or 2^n-1 when fully depressed to their mechanical limit. They must report values between 0 and 2^n-1 inclusive on a linear scale throughout the control surface's range of motion.

Digital switches must report status using a 1-bit value. They must report 1 when fully depressed to their mechanical limit and 0 otherwise.

Pressure sensitive switches must be sourced from specific contacts at Apple approved suppliers. See the MFi Portal for more details.

Pressure sensitive switches from the above suppliers must meet the following mechanical requirements:

Table 31-1 HID Game Controller pressure sensitive switch mechanical requirements

Parameter	Value at 0 Cycles	Value at 1M Cycles
Force to achieve mechanical `click'	70 - 130 gf	50 - 150 gf
Force to reach `on' state	< 130 gf	< 155 gf
Force to reach `fully on' state	< 400 gf	< 480 gf
Switch travel at `on' state	0.8 - 1.4 mm	0.7 - 1.5 mm

31.1.4 Labels

All text labels should be rendered in at least 10 point font. Helvetica is recommended. Unless otherwise specified, the color of the labels should be black or white, whichever provides most contrast against the controller enclosure or button.

All markings must be legible after 1 million cycles of use.

31.1.5 Face Button Group

A face button group must contain four circular buttons oriented 90° in a diamond layout. The buttons must be at least 6 mm in diameter, and the outer circle of the layout must be at least 20 mm in diameter.

The button group should be oriented within 5° of the controller axis. All of the button switches must be pressure sensitive.

Buttons must comply with the layout, color, and labeling as illustrated in Figure 31-7 (page 425) or in Figure 31-8 (page 426).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

The following color definitions are provided as reference for the face button group:

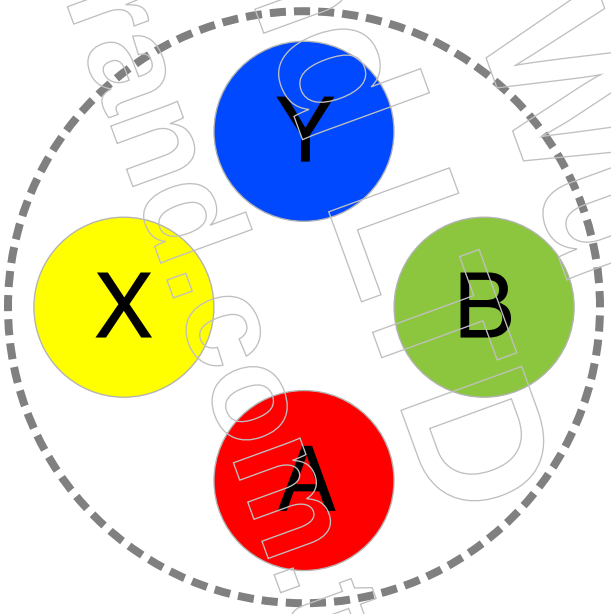
Table 31-2 HID Game Controller face button group color definitions

Button	Description	PTLI Color Code
A	Passion Red	912-0371-B
B	California Kiwi 120%	912-0754-A
X	Pollen Yellow Dark	912-0759-A
Y	Blue Fin Tuna 3	912-0753-A

To obtain color standards for matching or quality control purposes:

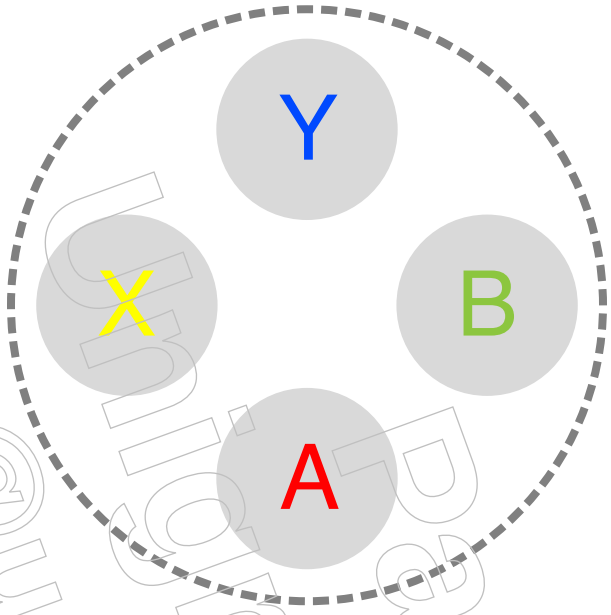
- Go to www.ptli.com/applecolor
- Choose "Order Color Standards"
- Enter user name and password to access and order standards. First time users must choose "Create New Account" and follow the instructions.

Figure 31-7 HID game controller face button group layout



Apple Confidential Party With HID Interest Only

Figure 31-8 HID game controller face button group layout (alternate)



31.1.6 Directional Pad

A directional pad must be implemented as one circular button that pivots in all directions. The button must not translate in any direction. The button must be at least 17 mm in diameter.

The directional pad must report its status using 4 pressure-sensitive switches, one for each cardinal direction (up, down, left, and right).

Users must be able to smoothly, quickly, and accurately generate all positions within the minimum unit circle using a directional pad.

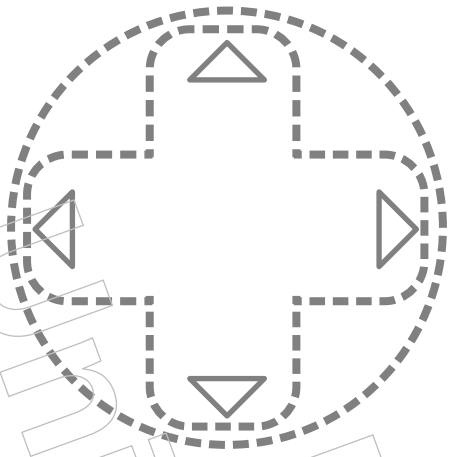
The button must implement a raised cross-shaped region that covers each cardinal direction. Each endpoint of the cross-shaped region must be labeled with a triangle oriented in the corresponding cardinal direction. The cross-shaped region must be proud of the controller's surface, but the remainder of the button may be flush or hidden from direct user contact.

The directional pad mechanical design must guarantee that opposing cardinal directions (i.e. up/down or left/right) cannot be depressed simultaneously.

The directional pad must be oriented within 5° of the controller axis. Directional pads must comply with the layout, color, and labeling as illustrated in Figure 31-9 (page 427).

paryvu@uniforandroid.com Paryvu Uniforandroid Ltd

Figure 31-9 HID gamepad directional pad layout



31.1.7 Shoulder Button

Shoulder buttons must be labeled according to the implemented controller definition.

Additionally,

- L and R on a standard gamepad must be pressure sensitive switches.
- L1 and R1 on an extended gamepad must be pressure sensitive switches.
- L2 and R2 on an extended gamepad must be either pressure sensitive switches or displacement triggers. The displacement trigger must have a minimum travel of 5 mm.

Figure 31-10 HID game controller shoulder button

31.1.8 Pause Button

Pause buttons must be circular with a diameter of at least 5 mm and labeled with the standard pause icon. They must use a digital switch.

The actuation force of the Pause button should be 1.2-3.5 N.

Note: The pause button must not be used as a power on/off button.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 31-11 HID game controller pause button

31.1.9 Joystick

Joysticks must be implemented using one of the following two-axis analog controls:

- Alps RKJXV1220001
- Alps RKJXY

The joystick must report a full range of movement between the unit circle and unit square using two signed *n*-bit integers where *n* is 8 or more, one for the X axis and one for the Y axis. The joystick should exhibit a positive centering force about both axes.

The joystick must rotate to at least 15° and up to 40° or translate at a minimum of 3 mm in all directions. The force to reach maximum travel in any direction must be between 0.7N and 1.3N. The breakout force to initiate values greater than (0,0) from rest state must be no greater than 30% of the force at maximum output. The output values of the joystick must be linearly reported with respect to physical displacement or angular displacement of the joystick. Extreme sensitivity or sluggishness in response to small control deflections or force must be avoided.

Joysticks must have a circular outline with a diameter of at least 10 mm.

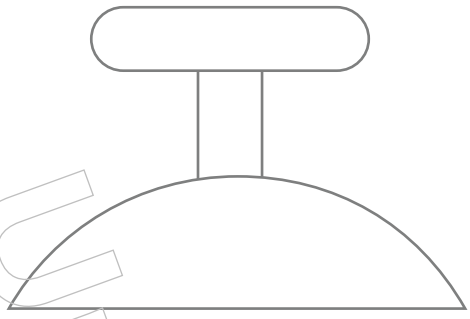
At rest, the joystick must report (0,0) for the X and Y axes respectively. Additionally, each axis must report a full range of values between $-2^{n-1}+1$ and $+2^{n-1}-1$ inclusive on a linear scale, where *n* is 8 or more. The maximum displacement on any given cardinal direction must be symmetrical to its opposite cardinal direction. The joystick dead zone must be a maximum of 5% of total joystick travel.

Users must be able to smoothly, quickly, and accurately generate all positions within the minimum unit circle using a joystick.

Note: Joysticks must not 'click' or permit the user to depress the stick to generate an input event.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 31-12 HID game controller joystick



31.1.10 LED Array

LED arrays must be physically implemented as a linear array of four red LEDs whose state maps to 4 unsigned 1-bit values (usage IDs 0xFF00-0xFF03) in the accessory's HID report descriptor. The LED state is controlled by the Apple device.

The LED array should be nominally visible while the user is holding the controller in a home position.

LED array state does not need to be reported to the Apple device.

The LED array may temporarily be used to display charge state for a gamepad's internal battery, if applicable. The LED behavior should be as follows:

- LEDs representing the current charge turn on, then fade, indicating the current battery level.
- The first LED flashes for 5 seconds, indicating that the battery has not yet been charged to what is required for a single indicator light.
- LEDs representing the current charge flash 5 times, then fade, indicating that the battery is charging, but does not have a full charge.
- All LEDs flash 5 times, indicating that the battery needs to be replaced.
- LEDs flash from left to right and back again 5 times, indicating that the (removable) battery is not installed or is not recognized.

31.1.11 Bluetooth Button

Controllers that utilize the Bluetooth transport must implement a button on the controller specifically for initiation of Bluetooth pairing. The button must not be in a location that can be accidentally pressed by the user during normal operation.

The button must be labeled with the Bluetooth icon or otherwise visually treated in such a way that makes its purpose apparent to the end user.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Pressing and holding the button for 2 seconds must place the controller into a discoverable and pairable mode for a minimum of 30 seconds. While the controller is in this mode, the controller must flash its LED array if it has one or otherwise visually indicate that it is in a pairable mode. The controller must automatically exit the discoverable and pairable mode if it does not receive any pairing requests. The visual indication must cease as soon as the controller exits a pairable mode.

The accessory must be able to remember at least 5 paired Apple devices. When the accessory receives a pairing request, if it has already remembered its maximum number of paired Apple devices, it must replace the pairing entry for the least recently connected Apple device.

31.1.12 Bluetooth Major/Minor Device Class

All controllers that utilize the Bluetooth transport must set their Major Device Class to 'Peripheral' and their Minor Device Class to 'Gamepad'.

31.1.13 App Match for Controller-Enabled Games

Game controllers must support App Match for an Apple-maintained list of controller-enabled games.

To do so, the controller must identify support for an External Accessory Protocol (see Table 52-10 (page 581)) with the following parameters:

Table 31-3 HID Game Controller App Match Parameters

Parameter	Value
ExternalAccessoryProtocolName	'com.apple.mfi.gamecontroller'
ExternalAccessoryProtocolMatchAction	1

This External Accessory Protocol must not be used for application data, firmware updates, or any other purpose. A separate External Accessory Protocol may be declared for these uses. See External Accessory Protocol (page 372).

31.1.14 Testing

All controllers must yield passing results for the tests defined in Test Procedures (page 435) to qualify for self certification.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

31.2 HID Game Controller Examples

31.2.1 Standard Gamepad Example HID Report Descriptor

This HID report descriptor includes the LED array that is optional for form-fitting controllers.

```
USAGE PAGE (Generic Desktop)        05 01
USAGE (GamePad)                     09 05
COLLECTION (Application)            A1 01
  USAGE (GamePad)                   09 05
  COLLECTION (Logical)              A1 02
    REPORT SIZE (8)                 75 08

    // Directional pad (up, right, down, left)
    REPORT COUNT (4)                95 04
    LOGICAL MINIMUM (0)             15 00
    LOGICAL MAXIMUM (255)           26 FF 00
    PHYSICAL MINIMUM (0)            35 00
    PHYSICAL MAXIMUM (255)          46 FF 00
    USAGE PAGE (Generic Desktop)    05 01
    USAGE (DPadUp)                  09 90
    USAGE (DPadRight)               09 92
    USAGE (DPadDown)                09 91
    USAGE (DPadLeft)                09 93
    INPUT (Data,Var,Abs)            81 02

    // Face button group and left/right shoulder buttons
    // (A, B, X, Y, L1, R1)
    REPORT COUNT (6)                95 06
    USAGE PAGE (Button)             05 09
    USAGE MINIMUM (Button 1)        19 01
    USAGE MAXIMUM (Button 6)        29 06
    INPUT (Data,Var,Abs)            81 02

    // Pause button
    LOGICAL MINIMUM (0)             15 00
```

Pary Wu Unicorn Land LTD
pary@unicornland.com.tw

```
    LOGICAL MAXIMUM (1)             25 01
    PHYSICAL MINIMUM (0)            35 00
    PHYSICAL MAXIMUM (1)            45 01
    REPORT SIZE (1)                 75 01
    REPORT COUNT (1)                95 01
    USAGE PAGE (Consumer)           05 0C
    USAGE (AC Home)                 0A 23 02
    INPUT (Data,Var,Abs)            81 02
    REPORT COUNT (7)                95 07
    INPUT (Cnst,Var,Abs)            81 03

    // LED array
    LOGICAL MINIMUM (0)             15 00
    LOGICAL MAXIMUM (1)             25 01
    PHYSICAL MINIMUM (0)            35 00
    PHYSICAL MAXIMUM (1)            45 01
    REPORT SIZE (1)                 75 01
    REPORT COUNT (4)                95 04
    USAGE PAGE (LED)                05 08
    USAGE MINIMUM (Reserved 0xFF00) 1A 00 FF
    USAGE MAXIMUM (Reserved 0xFF03) 2A 03 FF
    OUTPUT (Data,Var,Abs)           91 02
    REPORT SIZE (4)                 75 04
    REPORT COUNT (1)                95 01
    OUTPUT (Cnst,Ary,Abs)           91 01

  END COLLECTION (Logical)          C0
END COLLECTION (Application)        C0
```

31.2.2 Extended Gamepad Example HID Report Descriptor

This HID report descriptor includes the LED array that is required for non form-fitting controllers.

```
USAGE PAGE (Generic Desktop)        05 01
USAGE (GamePad)                     09 05
```

Party Wu LTD
party@unigrand.com

```
COLLECTION (Application)            A1 01
  USAGE (GamePad)                   09 05
  COLLECTION (Logical)              A1 02
    REPORT SIZE (8)                 75 08

    // Directional pad (up, right, down, left)
    REPORT COUNT (4)                95 04
    LOGICAL MINIMUM (0)             15 00
    LOGICAL MAXIMUM (255)           26 FF 00
    PHYSICAL MINIMUM (0)            35 00
    PHYSICAL MAXIMUM (255)          46 FF 00
    USAGE PAGE (Generic Desktop)    05 01
    USAGE (DPadUp)                  09 90
    USAGE (DPadRight)               09 92
    USAGE (DPadDown)                09 91
    USAGE (DPadLeft)                09 93
    INPUT (Data,Var,Abs)            81 02

    // Face button group and left/right shoulder/trigger buttons
    // (A, B, X, Y, L1, R1, L2, R2)
    REPORT COUNT (8)                95 08
    USAGE PAGE (Button)             05 09
    USAGE MINIMUM (Button 1)        19 01
    USAGE MAXIMUM (Button 8)        29 08
    INPUT (Data,Var,Abs)            81 02

    // Pause button
    LOGICAL MINIMUM (0)             15 00
    LOGICAL MAXIMUM (1)             25 01
    PHYSICAL MINIMUM (0)            35 00
    PHYSICAL MAXIMUM (1)            45 01
    REPORT SIZE (1)                 75 01
    REPORT COUNT (1)                95 01
    USAGE PAGE (Consumer)           05 0C
    USAGE (AC Home)                 0A 23 02
```

Party Wu Unigiand LTD
party@unigiand.com.tw

```
INPUT (Data,Var,Abs)            81 02
REPORT COUNT (7)                95 07
INPUT (Cnst,Var,Abs)            81 03

// LED array
LOGICAL MINIMUM (0)             15 00
LOGICAL MAXIMUM (1)             25 01
PHYSICAL MINIMUM (0)            35 00
PHYSICAL MAXIMUM (1)            45 01
REPORT SIZE (1)                 75 01
REPORT COUNT (4)                95 04
USAGE PAGE (LED)                05 08
USAGE MINIMUM (Reserved 0xFF00) 1A 00 FF
USAGE MAXIMUM (Reserved 0xFF03) 2A 03 FF
OUTPUT (Data,Var,Abs)           91 02
REPORT SIZE (4)                 75 04
REPORT COUNT (1)                95 01
OUTPUT (Cnst,Ary,Abs)           91 01

// Left/right joysticks
LOGICAL MINIMUM (-127)          15 81
LOGICAL MAXIMUM (127)           25 7F
PHYSICAL MINIMUM (-127)         35 81
PHYSICAL MAXIMUM (127)          45 7F
USAGE PAGE (Generic Desktop)    05 01
USAGE (Pointer)                 09 01
COLLECTION (Physical)           A1 00
  REPORT SIZE (8)               75 08
  REPORT COUNT (4)              95 04
  USAGE (X)                     09 30 // left axis horizontal
  USAGE (Y)                     09 31 // left axis vertical
  USAGE (Z)                     09 32 // right axis horizontal
  USAGE (Rz)                    09 35 // right axis vertical
  INPUT (Data,Var,Abs)          81 02
```

Party Wu
Unlimited LTD
party@unlimitedltd.com

```
    END COLLECTION (Physical)       C0
  END COLLECTION (Logical)          C0
END COLLECTION (Application)        C0
```

31.3 Test Procedures

The test procedures make use of the following tools and resources:

- ATS 3.4 or greater
- Caliper
- Force Gauge
- PTLI Color standards - see HID Game Controller face button group color definitions (page ?)

31.3.1 General Requirements

1. Verify that the controller does not implement alternate operating modes that emulate keyboards or any other type of Human Interface Device besides HID Game Controller.
2. Using ATS 3.4 or greater, verify that multiple controller samples have unique serial numbers via inspection of the SerialNumber parameter in the IdentificationInformation (page 579) message.
3. Verify that the controller is capable of having its firmware updated from Apple devices running iOS and/or Macintosh computers running OS X.

> **Note:** When submitting samples to Apple for self-certification, include the necessary files and instructions to verify the firmware update mechanism.

31.3.2 Bluetooth Controllers

1. Verify that the Bluetooth Major and Minor Device Class fields are set as specified in Bluetooth Major/Minor Device Class (page 430).
2. See Bluetooth Button (page 438).

31.3.3 Form-Fitting Controllers

1. Verify that the form-fitting controllers must meet the same requirements as Cases (page 336).

Partwuau@unigrand.com.tw Unigrand LTD

2. Verify that there is no relative movement between the accessory's Lightning connector and the Apple device when connected.
    a. Attach/connect the Apple device to the accessory.
    b. While holding the accessory, attempt to move the Apple device in various directions.
    c. Verify that the accessory's Lightning connector and the Apple device do not move independent of each other.
3. Verify that controller physically encloses the Apple device in landscape orientation and allows the user's thumbs to reach the Apple device's touchscreen.
4. Verify that the Apple device's touchscreen is not occluded by the accessory in any way.
5. Verify that the controller consists of only one physical form-fitting component.
6. Verify that the controller implements a 4 element LED array or no LED array.

31.3.4 Non Form-Fitting Controllers

1. Verify that controller does not physically enclose the Apple device.
2. Verify that the controller is an extended controller.
3. Verify that the controller consists of 1 to 4 physically separate components.
4. Verify that the controller implements a 4 element LED array.

31.3.5 Control Surface Layout and Labeling

1. Verify that the controller does not implement physical control surfaces other than those required by the standard or extended gamepad definitions (see Table 31-4 (page 436)), with the following exceptions:
    a. Charging on/off switch for a controller with integrated battery that provides power to the Apple device.
    b. Button to temporarily display charge level for a controller with integrated battery using the player indicator LEDs.
    c. Button to initiate Bluetooth pairing for a controller with Bluetooth support.

Table 31-4 Required Game Controller Control Surfaces

Control Surface	Standard Controller	Extended Controller
Pause button	√	√
Directional pad	√	√

Apple Confidential (apple@pubtalk.com)

Control Surface	Standard Controller	Extended Controller
Face button group (A, B, X, Y)	√	√
Left shoulder button (L)	√	
Right shoulder button (R)	√	
Top left shoulder button (L1)		√
Top right shoulder button (R1)		√
Bottom left shoulder button/trigger (L2)		√
Bottom right shoulder button/trigger (R2)		√
Left joystick		√
Right joystick		√

2. Verify that the layout and labeling of the control surfaces match as shown in Figure 31-5 (page 421) or Figure 31-6 (page 423) along with the following exceptions:
   a. Form-fitting controllers may position the pause button on either the left or the right side of the controller axis.
   b. Extended gamepad joysticks may be swapped vertically with their corresponding face button group or directional pad.
3. Verify that all text labels are rendered in at least 10 point font.
4. Verify that label text is black or white for all buttons except the face button group (see Face Button Group (page 437)).
5. Verify that the color of labels provides contrast against the surface (black text on light surface or white text on dark surface).

31.3.5.1 Face Button Group

1. Verify that the face button group contains four circular buttons oriented 90° in a diamond layout.
2. Using a caliper, verify that the face button group buttons are at least 6 mm in diameter.
3. Using a caliper, verify that the outer circle of the face button group is at least 20 mm in diameter.
4. Verify that the face button group is oriented within 5° of the controller axis.

Paul Wu
Unigrand Ltd
paul.wu@unigrand.com

5. Verify that the layout and labeling of the face button group matches either Figure 31-7 (page 425) or Figure 31-8 (page 426).
6. Verify that the face buttons match the color requirements in Table 31-2 (page 425).

31.3.5.2 Directional Pad

1. Verify that the directional pad is one circular button that pivots in all directions and does not translate in any direction.
2. Using a caliper, verify that the directional pad is at least 17 mm in diameter.
3. Verify that the directional pad implements a raised cross-shaped region covering each cardinal direction, and that the cross-shaped region is proud of the controller's surface. Verify that each endpoint of the raised region is labeled with a triangle oriented in the corresponding direction.
4. Verify that the directional pad is oriented within 5° of the controller axis.
5. Verify that the layout and labeling of the directional pad matches Figure 31-9 (page 427).

31.3.5.3 Joystick

1. Using a caliper, verify that the joystick has a circular outline with a diameter of at least 10 mm.
2. Using a caliper, verify that the maximum displacement on any given cardinal direction is symmetrical to its opposite cardinal direction.
3. Verify that the joystick does not "click" or translate in any axes other than up/down and left/right.

31.3.5.4 Pause Button

1. Using a caliper, verify that the pause button is circular with a diameter of at least 5 mm.
2. Verify that the pause button is labeled with the standard pause icon.

31.3.5.5 LED Array

1. Verify that the LED array is implemented as a linear array of four red LEDs. (Use whatever means is available to activate the LEDs. For example, use the battery status button if one is present.)
2. Verify that the LED array is nominally visible while the user is holding the controller in a home position.
3. Using ATS 3.4 or greater, verify that the array of four red LEDs state maps to 4 unsigned 1-bit values in the accessory's HID report descriptor.

31.3.5.6 Bluetooth Button

1. Verify that there is a dedicated button on the controller to initiate Bluetooth pairing.

Barwin@uniranet.com Universal Serial Bus HID

2. Verify that the button can't be accidentally pressed by the user during normal operation.
3. Verity that the button is labeled with the Bluetooth icon or otherwise visually treated to makes its purpose apparent.
4. Press and hold the button for 2 seconds. Verify that the accessory is discoverable and pairable for at least 30 seconds. Verify that it indicates this visually.
5. Verify that the accessory exits discoverable and pairable mode automatically.
6. Pair the accessory with an Apple device (device "A"). Verify that the accessory is paired.
7. Pair the accessory with 1 more Apple device (device "B").
8. Pair the accessory with as many more Apple devices as it can remember (at least 3 more).
9. Use first Apple device (device "A") to connect to the accessory.
10. Pair the accessory with 1 more Apple device.
11. Verify that the accessory does not connect to device "B" and does connect with device "A" and the other paired devices.

31.3.6 Control Surface Switches

31.3.6.1 Pressure Sensitive Buttons

For every pressure sensitive button in the face button group, shoulder buttons, & directional pad:

1. Using ATS 3.4 or greater, verify that pressure sensitive switches report a value of 0 when not depressed or 2^n-1 when fully depressed to their mechanical limit. (n = the Report Size value for the HID report descriptor's definition of the switch.)
2. Using ATS 3.4 or greater, verify that button down vs. button up reports are received separately.
3. Using ATS 3.4 or greater, verify that HID reports are only sent when button state / control state changes (not sent repeatedly if value is not changing).
4. Using a digital force gauge, verify that the button achieves a mechanical `click' when force between 70 to 130 gf is applied.
5. Using a digital force gauge, verify that the button reaches its `on' state (a value that is not 0) when < 130 gf is applied.
6. Using a digital force gauge, verify that the button reaches its `fully on' state (2^n-1) when < 400 gf is applied.
7. For all pressure sensitive buttons: Using calipers, measure the travel of the button to reach `on' state and verify that it is between 0.8 - 1.4 mm.
8. For all displacement triggers: Using calipers, measure the travel of the trigger from `on' to `fully on' and verify that it is at least 5 mm.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Additionally, for the directional pad:

1. Using ATS 3.4 or greater, verify that the directional pad reports its status using 4 pressure-sensitive switches, one for each cardinal direction (up, down, left, and right).
2. Using ATS 3.4 or greater, verify that opposing cardinal directions commands (i.e. up/down or left/right) are not sent while the buttons are depressed simultaneously.

31.3.6.2 Pause Button

1. Using a digital force gauge, verify that the pause button actuates when force between 1.2 to 3.5 N is applied.
2. Using ATS 3.4 or greater, verify that the pause button is digital by verifying that the HID usage report size is equal to 1.
3. Using ATS 3.4 or greater, verify that the pause button reports 1 when fully depressed and 0 otherwise.

31.3.6.3 Joystick

For each joystick:

1. Using a digital force gauge, verify that the joystick rotates to 30°±10° or translates at a minimum of 1.5 mm in all directions when a linear, horizontal force between 0.7 N and 1.3 N is applied.
2. Using ATS 3.4 or greater, verify that the joystick is implemented as a two-axis analog control using two signed *n*-bit integers where *n* is 8 or more, one for the X axis and one for the Y axis.
3. Using ATS 3.4 or greater, verify that the joystick reports values of *0* for each axis when at rest.
4. Verify that the joystick reports a full range of movement between the unit circle and unit square.
    a. Extend the joystick as far as possible in both directions of each cardinal direction (up, down, left, and right) and verify that the values $-2^{n-1}+1$ and $+2^{n-1}-1$ are appropriately sent.
    b. Extend the joystick as far as possible in both directions of each intracardinal direction (up-left, up-right, down-right, down-left) and verify that the values $\leq \left(\frac{\sqrt{2}}{2}\right)\left(-2^{n-1}+1\right)$ and $\geq \left(\frac{\sqrt{2}}{2}\right)\left(+2^{n-1}-1\right)$ are appropriately sent.
5. Verify that the joystick dead zone is a maximum of 5% of total joystick travel.
    a. Using a caliper, measure the total joystick distance of travel in one direction from center.
    b. Calculate 5% of the measured distance from above and verify that non-zero values are sent when the joystick is moved this distance from center in the 4 cardinal and 4 intercardinal directions.

Unauthorized Part Number Reproduction Prohibited

32. HID Headphone Remote

Accessories that support the HID Headphone Remote feature may interact with Apple devices in a manner that builds on top of the behaviors enabled by the Headphone Remote and Mic feature.

32.1 HID Headphone Remote Requirements

All accessories that implement a HID Headphone Remote component must be a headphone (see Headphones (page 378)) that connects to the Apple device via either the Lightning connector or Bluetooth. Also, they must implement only one headphone remote component.

Headphone remote components must support the HID (Human Interface Device) (page 407) feature and comply with all the requirements listed in HID Requirements (page 407).

Headphone remotes that send a Volume Down or Volume Up HID button press usage must immediately follow that usage with the corresponding button release usage even if the user has not released the button yet. The result is that repeated volume ramping must not occur when the volume up or down buttons are pressed and held, matching the behavior of Apple headphone remotes.

The HID report descriptor for a headphone remote must declare support for the HID Consumer and/or Telephony pages and only send usages from the following tables:

Table 32-1 HID Consumer Page controls for use by headphone remotes

Usage ID	Usage Name	Apple Function
0x00B5	Scan Next Track	Transport Right
0x00B6	Scan Previous Track	Transport Left
0x00B9	Random Play	Shuffle
0x00BC	Repeat	Repeat
0x00E2	Mute	Mute
0x00E9	Volume Increment	Volume Up
0x00EA	Volume Decrement	Volume Down

parwu@ubeidi.com.tw Ubeidi Parwu Ltd

Usage ID	Usage Name	Apple Function
0x025B	Promote	iTunes Radio "Play More Like This"
0x025C	Demote	iTunes Radio "Never Play This Song"
0x0262	Add to Cart	iTunes Radio "Add to iTunes Wish List"

Table 32-2 HID Telephony Page controls for use by headphone remotes

Usage ID	Usage Name	Apple Function
0x0021	Flash	Center

32.2 HID Headphone Remote Examples

32.2.1 Headphone Remote Example HID Report Descriptor (Telephony)

The following sample HID descriptor demonstrates how to implement the same controls found on the Apple Headphone Remote that can be used with all apps, including telephony.

```
USAGE_PAGE (Consumer Devices) 05 0C
USAGE (Consumer Control)      09 01
COLLECTION (Application)      A1 01
  LOGICAL_MINIMUM (0)         15 00
  LOGICAL_MAXIMUM (1)         25 01
  REPORT_SIZE (1)             75 01
  REPORT_COUNT (2)            95 02
  USAGE (Volume Increment)    09 E9    // Volume Up
  USAGE (Volume Decrement)    09 EA    // Volume Down
  INPUT (Data,Var,Abs)        81 02
  USAGE_PAGE (Telephony)      05 0B
  REPORT_COUNT (1)            95 01
  USAGE (Flash)               09 21    // Center
  INPUT (Data,Var,Abs)        81 02
  REPORT_SIZE (5)             75 05
  REPORT_COUNT (1)            95 01
```

Paul Wu
Universal Audio LTD
paulwu@universal-audio.com.tw

```
  INPUT (Cnst, Var, Abs)      81 03
END_COLLECTION                C0
```

Each report is one byte, and each bit corresponds to one of the functions. For instance, the following sample reports communicate that the referenced button has been pressed:

- Volume Up is `0x01`
- Volume Down is `0x02`
- Center is `0x04`

32.2.2 Headphone Remote Example HID Report Descriptor (Media Playback)

The following sample HID descriptor demonstrates how to implement media playback controls, including iTunes Radio buttons.

```
USAGE PAGE (Consumer Devices) 05 0C
USAGE (Consumer Control)      09 01
COLLECTION (Application)      A1 01
  LOGICAL MINIMUM (0)         15 00
  LOGICAL MAXIMUM (1)         25 01
  REPORT SIZE (1)             75 01
  REPORT COUNT (8)            95 08
  USAGE (Scan Next Track)     09 B5    // Next Track
  USAGE (Scan Previous Track) 09 B6    // Previous Track
  USAGE (Mute)                09 E2    // Mute
  USAGE (Shuffle)             09 B9    // Shuffle
  USAGE (Repeat)              09 BC    // Repeat
  USAGE (Promote)             0A 5B 02 // iTunes Radio 'Play More Like This'
  USAGE (Demote)              0A 5C 02 // iTunes Radio 'Never Play This Song'
  USAGE (Add to Cart)         0A 62 02 // iTunes Radio 'Add to Wish List'
  INPUT (Data,Var,Abs)        81 02
END COLLECTION                C0
```

Each report is one byte, and each bit corresponds to one of the functions. For instance, the following sample reports communicate that the referenced button has been pressed:

- Next Track is `0x01`

Tanya Wu tanyawu@uniland.com.tw Uniland Industrial Co., Ltd.

- Previous Track is `0x02`
- Mute is `0x04`

32.2.3 Headphone Remote Example HID Report Descriptor (Telephony and Media Playback)

The following sample HID descriptor demonstrates how to implement all possible media playback controls along with the same controls found on the Apple Headphone Remote.

```
USAGE_PAGE (Consumer Devices) 05 0C
USAGE (Consumer Control)      09 01
COLLECTION (Application)      A1 01
  LOGICAL_MINIMUM (0)         15 00
  LOGICAL_MAXIMUM (1)         25 01
  REPORT_SIZE (1)             75 01
  REPORT_COUNT (10)           95 0A
  USAGE (Scan Next Track)     09 B5
  USAGE (Scan Previous Track) 09 B6
  USAGE (Mute)                09 E2    // Mute
  USAGE (Shuffle)             09 B9    // Shuffle
  USAGE (Repeat)              09 BC    // Repeat
  USAGE (Promote)             0A 5B 02 // iTunes Radio 'Play More Like This'
  USAGE (Demote)              0A 5C 02 // iTunes Radio 'Never Play This Song'
  USAGE (Add to Cart)         0A 62 02 // iTunes Radio 'Add to Wish List'
  USAGE (Volume Increment)    09 E9    // Volume Up
  USAGE (Volume Decrement)    09 EA    // Volume Down
  INPUT (Data,Var,Abs)        81 02
  USAGE_PAGE (Telephony)      05 0B
  REPORT_COUNT (1)            95 01
  USAGE (Flash)               09 21    // Center
  INPUT (Data,Var,Abs)        81 02
  REPORT_SIZE (5)             75 05
  REPORT_COUNT (1)            95 01
  INPUT (Cnst, Var, Abs)      81 03
END COLLECTION                C0
```

Party Virtual Library LTD
partyvirtuallibrary.com.tw

Each report is two bytes. The bits are assigned top-to-bottom (from scan next track to select). For instance, the following sample reports communicate that the referenced button has been pressed:

- Next Track is `0x0100`
- Previous Track is `0x0200`
- Mute usage is `0x0400`
- Add To Wish List is `0x8000`
- Volume Up is `0x0001`
- Volume Down is `0x0002`
- Center is `0x0004`

Pary Wu
Uniqrand LTD
paryu@uniqrand.com.tw

33. HID Keyboard

iOS apps may accept user input from accessories that implement HID Keyboard components in place of the onscreen keyboard.

33.1 HID Keyboard Requirements

Accessories that implement HID Keyboard components must support the HID (Human Interface Device) (page 407) feature and comply with all the requirements listed in HID Requirements (page 407).

> **Note:** Accessory keyboards must not identify themselves as Apple-branded keyboards, e.g. use the Apple Vendor ID and/or Product IDs.

Accessory keyboard keys that exhibit any of the following behaviors are explicitly prohibited:

- Send anything other than a 'key pressed' or 'key released' for the key that has physically been pressed
- Emulate combinations of other keys (e.g. macros like Command-C for Paste)
- Emulate timed user actions such as 'press-and-hold'
- Send different HID usages depending on the state of another control surface

Each HID component that declares itself as a keyboard must correspond to a physical keyboard on the accessory that the user uses for tasks that might have otherwise been performed using the Apple device's onscreen keyboard. All HID usages sent from the keyboard accessory must occur in response to direct user action, i.e. pressing a key on the keyboard.

Conversely, any accessory that has physical or virtual user controls that appear to be a general purpose keyboard must implement a keyboard HID component and send corresponding HID usages. Mapping those controls to other iAP2 messages is grounds for failure to pass self certification.

Keyboards may integrate an LED to indicate the status of Caps Lock. Apple devices do not support any other keyboard status LEDs. Keyboards must not incorporate any other keyboard status LEDs, with one exception. Bluetooth keyboards may include a connection status LED.

Mechanical key layout must be based on the ISO/IEC 9995-2, ANSI-INCITS 154-1988, or JIS X 6002-1980 standards.

Part Number@Unlicensed LED

If the HID keyboard component is implemented using the HID Native Transport (USB Host Mode), its USB HID descriptor must set the *bCountryCode* field to the appropriate country code as defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor*.

If the HID keyboard component is implemented using iAP2, the *LocalizedKeyboardCountryCode parameter* in the StartHID (page 599) message must be set to the appropriate country code as defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor*.

The HID report descriptor for a keyboard must declare support for the HID Keyboard/Keypad Page. For efficiency, the keyboard may declare support for Keyboard/Keypad page usages that it does not send.

Accessory keyboards must implement individual keys that emit the following HID Keyboard/Keypad page usages:

Table 33-1 Required HID Keyboard/Keypad Page controls for use by keyboard components

Usage ID	Usage Name	Apple Function
0x0004	a and A	a and A
0x0005	b and B	b and B
0x0006	c and C	c and C
0x0007	d and D	d and D
0x0008	e and E	e and E
0x0009	f and F	f and F
0x000A	g and G	g and G
0x000B	h and H	h and H
0x000C	i and I	i and I
0x000D	j and J	j and J
0x000E	k and K	k and K
0x000F	l and L	l and L
0x0010	m and M	m and M
0x0011	n and N	n and N
0x0012	o and O	o and O

Paly Wu
Paly@unigrand.com.tw
Unigrand LTD

Usage ID	Usage Name	Apple Function
0x0013	p and P	p and P
0x0014	q and Q	q and Q
0x0015	r and R	r and R
0x0016	s and S	s and S
0x0017	t and T	t and T
0x0018	u and U	u and U
0x0019	v and V	v and V
0x001A	w and W	w and W
0x001B	x and X	x and X
0x001C	y and Y	y and Y
0x001D	z and Z	z and Z
0x001E	1 and !	1 and !
0x001F	2 and @	2 and @
0x0020	3 and #	3 and #
0x0021	4 and $	4 and $
0x0022	5 and %	5 and %
0x0023	6 and ^	6 and ^
0x0024	7 and &	7 and &
0x0025	8 and *	8 and *
0x0026	9 and (	9 and (
0x0027	0 and)	0 and)
0x0028	Return/Enter	Return
0x002A	Delete/Backspace	Delete
0x002B	Tab	Tab

Party Wu Digital LTD
Unauthorized Distribution

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Usage ID	Usage Name	Apple Function
0x002C	Spacebar	Spacebar
0x002D	- and _	- and _
0x002E	= and +	= and +
0x002F	[and {	[and {
0x0030	] and }	] and }
0x0031	\and |	\and |
0x0033	; and :	; and :
0x0034	' and "	' and "
0x0035	Grave Accent and Tilde	` and ~
0x0036	, and <	, and <
0x0037	. and >	. and >
0x0038	/ and ?	/ and ?
0x0039	CapsLock	Caps Lock
0x004F	RightArrow	Right Arrow
0x0050	LeftArrow	Left Arrow
0x0051	DownArrow	Down Arrow
0x0052	UpArrow	Up Arrow
0x00E1	LeftShift	Left Shift
0x00E2	LeftAlt	Left Option and Alt
0x00E3	LeftGUI	Left Command and ⌘
0x00E5	RightShift	Right Shift
0x00E6	RightAlt	Right Option and Alt
0x00E7	RightGUI	Right Command and ⌘

Party Wu LTD taiwu@unigrandtech.com

JIS keyboards must also implement additional keys found on the Apple Wireless Keyboard (Japanese). Non-JIS keyboards must not implement these keys.

Table 33-2 Required HID Keyboard/Keypad Page controls for use by JIS keyboard components

Usage ID	Usage Name	Apple Function
0x0032	Non-US # and ~	see Apple Wireless Keyboard (Japanese)
0x0087	Keyboard International1	see Apple Wireless Keyboard (Japanese)
0x0089	Keyboard International3	see Apple Wireless Keyboard (Japanese)
0x0090	LANG1	Switch to Previous Language
0x0091	LANG2	Switch to Next Language

Accessory keyboards may implement individual keys that emit the following HID Keyboard/Keypad page usages:

Table 33-3 Optional HID Keyboard/Keypad Page controls for use by keyboard components

Usage ID	Usage Name	Apple Function
0x0029	Escape	Escape
0x00E0	LeftControl	Left Control
0x00E4	RightControl	Right Control
0x004A	Home	Home
0x004D	End	End
0x0054	Keypad /	Keypad /
0x0055	Keypad *	Keypad *
0x0056	Keypad -	Keypad -
0x0057	Keypad +	Keypad +
0x0058	Keypad Enter	Keypad Enter
0x0059	Keypad 1 and End	Keypad 1
0x005A	Keypad 2 and Down Arrow	Keypad 2

Taiwan Unigrand LTD
Wu Tai-Yu
taiwu@unigrand.com

Usage ID	Usage Name	Apple Function
0x005B	Keypad 3 and PageDn	Keypad 3
0x005C	Keypad 4 and Left Arrow	Keypad 4
0x005D	Keypad 5	Keypad 5
0x005E	Keypad 6 and Right Arrow	Keypad 6
0x005F	Keypad 7 and Home	Keypad 7
0x0060	Keypad 8 and Up Arrow	Keypad 8
0x0061	Keypad 9 and PageUp	Keypad 9
0x0062	Keypad 0 and Insert	Keypad 0
0x0063	Keypad . and Delete	Keypad .
0x0067	Keypad =	Keypad =

Accessory keyboards may implement individual keys that emit the following HID Consumer page usages:

Table 33-4 HID Consumer Page controls for use by keyboard components

Usage ID	Usage Name	Apple Function
0x0030	Power	Lock
0x0040	Menu	Home
0x006F	Display Brightness Increment	Brighter
0x0070	Display Brightness Decrement	Dimmer
0x00B5	Scan Next Track	Transport Right
0x00B6	Scan Previous Track	Transport Left
0x00CD	Play/Pause	Play/Pause
0x00CF	Voice Command	Siri
0x00E2	Mute	Mute
0x00E9	Volume Increment	Louder

Paul Wu Digital Audio LTD
paul@digitalaudio.com.tw

Usage ID	Usage Name	Apple Function
0x00EA	Volume Decrement	Softer
0x01AE	AL Keyboard Layout	Toggle Onscreen Keyboard
0x0221	AC Search	Spotlight
0x025B	Promote	iTunes Radio "Play More Like This"
0x025C	Demote	iTunes Radio "Never Play This Song"
0x0262	Add to Cart	iTunes Radio "Add to iTunes Wish List"

33.2 HID Keyboard Examples

33.2.1 Keyboard Example HID Report Descriptor

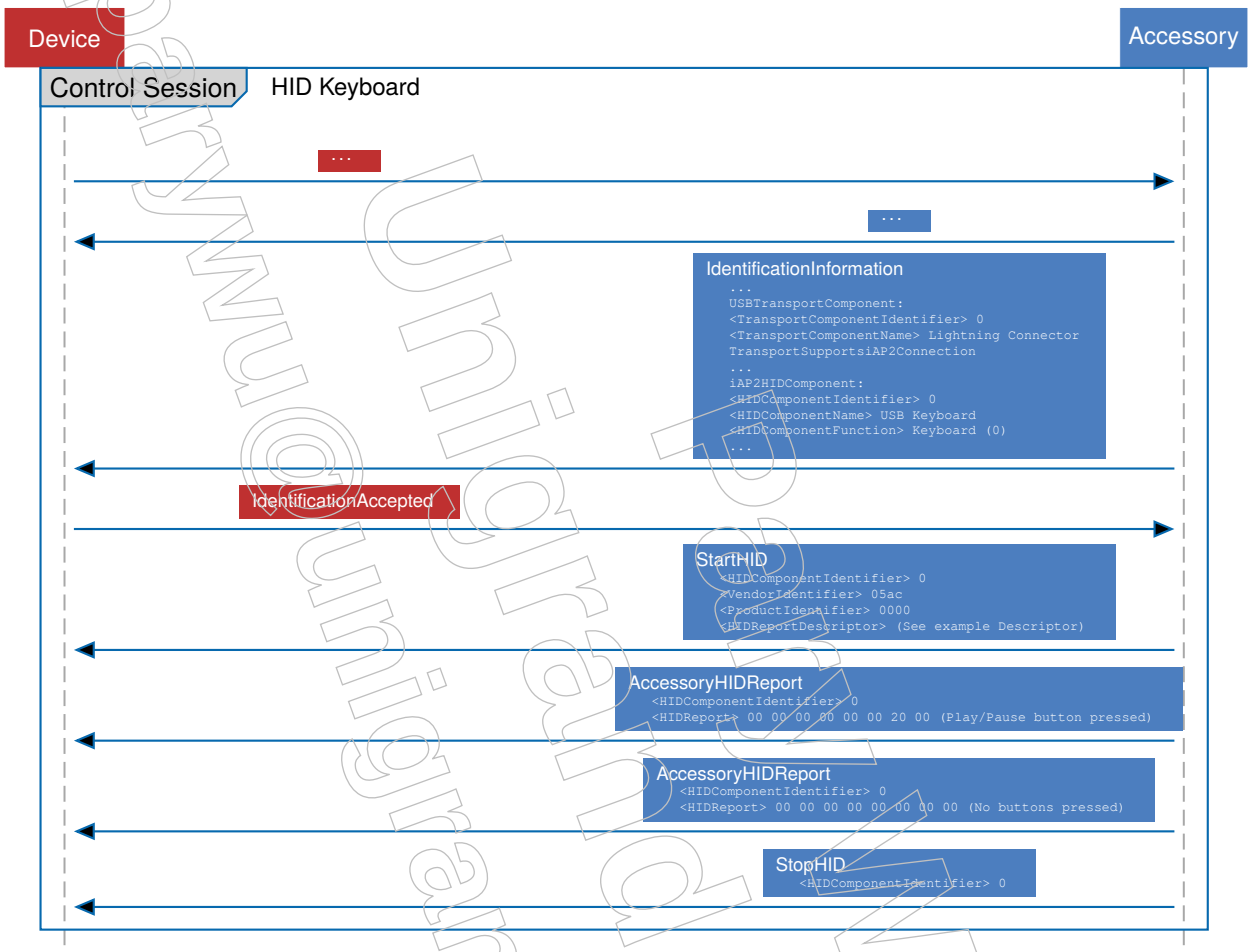
```
USAGE PAGE (Generic Desktop)                05 01
USAGE (Keyboard)                            09 06
COLLECTION (Application)                    A1 01
  USAGE PAGE (LEDs)                         05 08
  LOGICAL MINIMUM (0)                       15 00
  LOGICAL MAXIMUM (1)                       25 01
  USAGE (Caps Lock)                         09 02
  REPORT SIZE (1)                           75 01
  REPORT COUNT (1)                          95 01
  OUTPUT (Data,Var,Abs)                     91 02
  REPORT SIZE (7)                           75 07
  REPORT COUNT (1)                          95 01
  OUTPUT (Cnst,Var,Abs)                     91 03
  USAGE PAGE (Keyboard)                     05 07
  USAGE MINIMUM (Keyboard Left Control)     19 E0
  USAGE MAXIMUM (Keyboard Right GUI)        29 E7
  REPORT SIZE (1)                           75 01
  REPORT COUNT (8)                          95 08
  INPUT (Data,Var,Abs)                      81 02
```

Party Wu Universal LTD
tobywu@universal.com

```
  LOGICAL MINIMUM (0)                   15 00
  LOGICAL MAXIMUM (255)                 26 FF 00
  USAGE MINIMUM (0)                     19 00
  USAGE MAXIMUM (255)                   2A FF 00
  REPORT SIZE (8)                       75 08
  REPORT COUNT (5)                      95 05
  INPUT (Data,Ary,Abs)                  81 00
  USAGE PAGE (Consumer Devices)         05 0C
  LOGICAL MINIMUM (0)                   15 00
  LOGICAL MAXIMUM (1)                   25 01
  USAGE (Menu)                          09 40
  USAGE (AC Search)                     0A 21 02
  USAGE (AL Keyboard Layout)            0A AE 01
  USAGE (Scan Previous Track)           09 B6
  USAGE (Play/Pause)                    09 CD
  USAGE (Scan Next Track)               09 B5
  USAGE (Mute)                          09 E2
  USAGE (Volume Down)                   09 EA
  USAGE (Volume Up)                     09 E9
  USAGE (Power)                         09 30
  REPORT SIZE (1)                       75 01
  REPORT COUNT (10)                     95 0A
  INPUT (Data,Var,Abs)                  81 02
  REPORT SIZE (6)                       75 06
  REPORT COUNT (1)                      95 01
  INPUT (Cnst,Var,Abs)                  81 03
END COLLECTION                          C0
```

Pary Wu
Unigrand LTD
Pary.wu@unigrand.com.tw

33.2.2 Keyboard Usage Example



33.3 Test Procedures

33.3.1 General Requirements

33.3.1.1 Product Design

Verify that the accessory:

1. Implements a physical keyboard that the user uses for tasks that might have otherwise been performed using the Apple device's onscreen keyboard.
2. Implements individual keys that emit the HID Keyboard/Keypad page usages listed in Table 33-1 (page 447).

Copyright © 2014 Apple Inc. All Rights Reserved.

3. Implements individual keys that emit the LANG1 and LANG2 HID Keyboard/Keypad page usages listed in Table 33-2 (page 450) if the accessory is a JIS keyboard.
4. Does not incorporate any other keyboard status LEDs other than Caps Lock.
5. Has a mechanical key layout based on the ISO/IEC 9995-2, ANSI-INCITS 154-1988, or JIS X 6002-1980 standards.

33.3.1.2 HID Component

Verify that the accessory:

1. Supports the HID (Human Interface Device) (page 407) feature.
2. Complies with the requirements listed in HID Requirements (page 407).
3. Sets the *bCountryCode* field to the appropriate country code as defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor* if the HID component is implemented over a HID Native Transport (USB Host Mode).
4. Sets the *LocalizedKeyboardCountryCode* parameter in the StartHID (page 599) message to the appropriate country code as defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor* if the HID component is implemented over iAP2.

33.3.1.3 Direct User Action

Verify that the accessory:

1. Only sends HID usages in response to direct user action, i.e. pressing a key on the keyboard.
2. Does not send anything other than 'key pressed' or 'key released' for the key that has physically been pressed.
3. Does not emulate combinations of other keys (e.g. macros like Command-C for Paste).
4. Does not emulate timed user actions such as 'press-and-hold'.
5. Does not send different HID usages depending on the state of another control surface.

Uploaded by Wu
taryw@chuanglid.com.tw

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

34. HID Media Playback Remote

Accessories that implement HID Media Playback Remote components must support the HID (Human Interface Device) (page 407) feature and comply with all the requirements listed in HID Requirements (page 407).

34.1 HID Media Playback Remote Requirements

Each HID component that declares itself as a media playback remote must correspond to a physical set of media playback control surfaces. All HID usages sent from the media playback remote accessory must occur in response to direct user action, i.e. pressing one of the control surfaces. There are only three exceptions:

- If a media playback remote wishes to start playback immediately upon connection with the Apple device, it may send the `Play` HID usage.
- If a media playback remote wishes to start playback immediately upon changing audio sources to the Apple device, it may send the `Play` HID usage.
- If a media playback remote wishes to stop playback immediately upon changing audio sources from the Apple device, it may send the `Pause` HID usage.

> **Note:** Use of the `Play/Pause` HID usage for the exceptions listed above is not allowed.

Conversely, any accessory that has physical or virtual user controls that map to the media playback remote HID component must implement a media playback remote HID component and send corresponding HID usages. Mapping those controls to other iAP2 messages is grounds for failure to pass self certification.

The HID report descriptor for a media playback control must declare support for the HID Consumer Page and only send usages from the following table:

Table 34-1 HID Consumer Page controls for use by media playback remote components

Usage ID	Usage Name	Apple Function
0x00B0	Play	Play
0x00B1	Pause	Pause

Confidential Property of Apple Inc. Do not distribute

Usage ID	Usage Name	Apple Function
0x00B5	Scan Next Track	Transport Right
0x00B6	Scan Previous Track	Transport Left
0x00B9	Random Play	Shuffle
0x00BC	Repeat	Repeat
0x00BE	Tracking Normal	Reset audiobook playback speed to default
0x00CA	Tracking increment	Increase audiobook playback speed
0x00CB	Tracking Decrement	Decrease audiobook playback speed
0x00CD	Play/Pause	Play/Pause
0x00CF	Voice Command	Siri
0x00E2	Mute	Mute
0x00E9	Volume Increment	Louder
0x00EA	Volume Decrement	Softer
0x025B	Promote	iTunes Radio "Play More Like This"
0x025C	Demote	iTunes Radio "Never Play This Song"
0x0262	Add to Cart	iTunes Radio "Add to iTunes Wish List"

If a user presses and holds the accessory control surface corresponding to the Scan Next Track or Scan Previous Track HID usage, Apple devices may scrub forwards or backwards within the current playing media item. Accessories must not present a separate 'Fast-Forward' or 'Reverse' control surface to the user for the same feature.

34.2 HID Media Playback Remote Examples

34.2.1 Media Playback Remote Example HID Report Descriptor

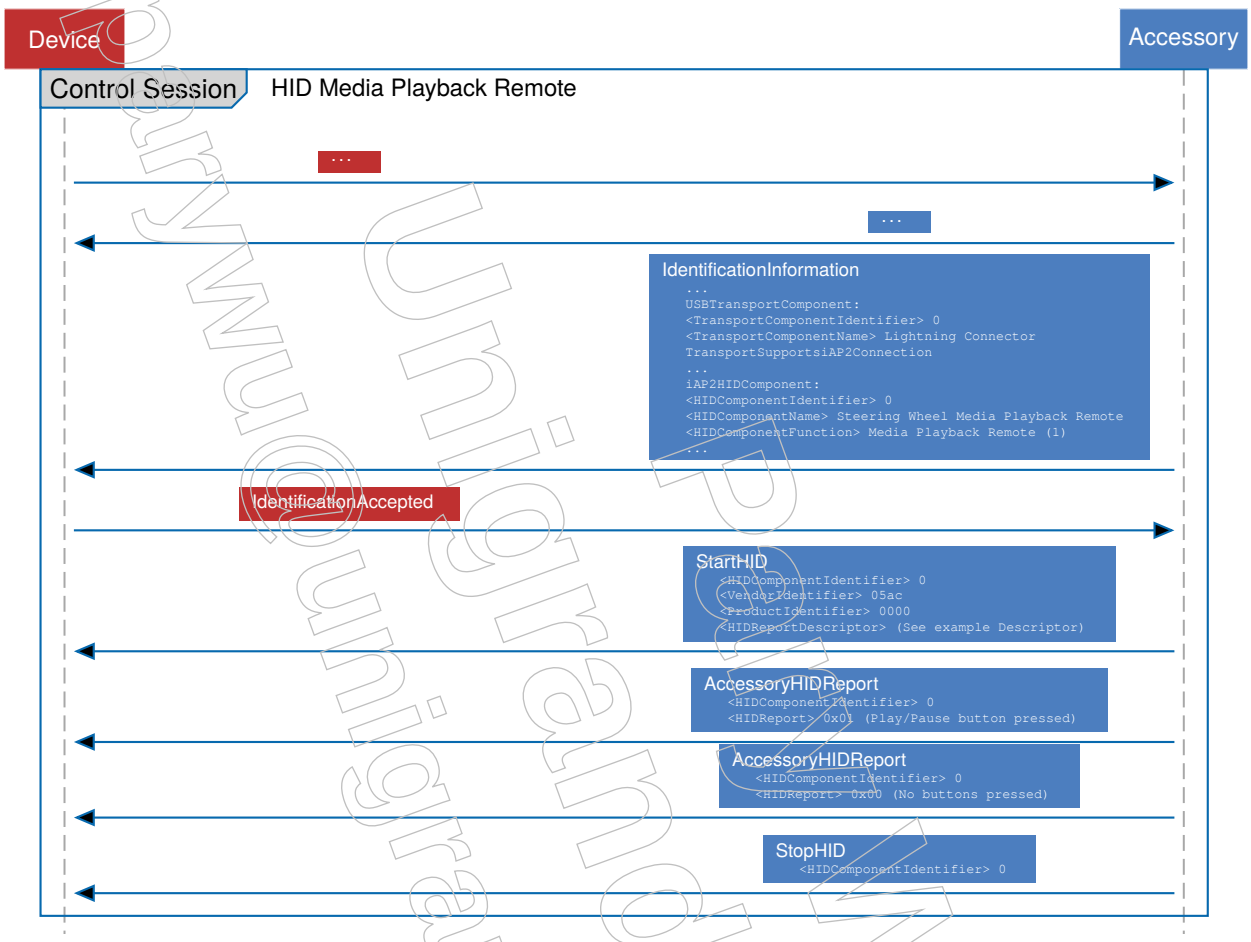
```
USAGE PAGE (Consumer Devices) 05 0C
USAGE (Consumer Control)      09 01
```

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

```
COLLECTION (Application)      A1 01
  LOGICAL MINIMUM (0)         15 00
  LOGICAL MAXIMUM (1)         25 01
  REPORT SIZE (1)             75 01
  REPORT COUNT (5)            95 05
  USAGE (Play/Pause)          09 CD
  USAGE (Scan Next Track)     09 B5
  USAGE (Scan Previous Track) 09 B6
  USAGE (Volume Up)           09 E9
  USAGE (Volume Down)         09 EA
  INPUT (Data,Var,Abs)        81 02
  REPORT SIZE (3)             75 03
  REPORT COUNT (1)            95 01
  INPUT (Cnst,Var,Abs)        81 03
END COLLECTION                C0
```

Party Wu
Courtland LTD
party@courtland.com.tw

34.2.2 Media Playback Remote Example Usage



Copyright © Unigrand LTD Yu-Hsiang Wu
yuwu@unigrand.com.tw

35. Location Information

The location feature provides compatible accessories with a means to provide external Global Navigation Satellite System (GNSS) information and other sensor data (like speed, gyro) in the form of National Marine Electronics Association (NMEA) sentences to Apple devices. Apple devices that support this feature can take advantage of this additional information to augment their built-in location services. For example, some external Location-enabled accessories can provide more accurate or more frequent position updates. Additionally, Apple devices can conserve power by using location information from a self-powered external accessory instead of using on-board positioning technologies.

All NMEA sentences provided to the Apple device must be NULL terminated.

35.1 Location Information Requirements

All accessories that support the Location feature via iAP2 must send or receive the following iAP2 control session message(s):

StartLocationInformation (page 601)

LocationInformation (page 602)

StopLocationInformation (page 602)

Accessories must provide location information to the Apple device as follows:

- Accessories with GNSS suitable for use in navigation must implement Global Navigation Satellite System (GNSS) Mode (page 461).
- Accessories without GNSS suitable for use in navigation must implement Sensors Mode (page 462).

Table 35-1 Supported NMEA Sentences and Maximum/Recommended Rates

Sentence Type	Maximum Rate	Recommended Rate
GPGGA	4 sentences per second	1 sentence per second
GPRMC	4 sentences per second	1 sentence per second
GPGSV	12 sentences per second	3-4 sentences per second

Part Wu (pwu@ford.com) Ford Confidential

Sentence Type	Maximum Rate	Recommended Rate
GPHDT	4 sentences per second	2 sentences per second
PASCD	50 sentences per second	1-4 sentences per second
PAGCD	50 sentences per second	1-4 sentences per second
PAACD	50 sentences per second	1-4 sentences per second

Accessories must provide location information to the Apple device in the form of NMEA comma-separated sentences at a minimum 1 Hz update rate.

> **Note:** NMEA sentences supplied to the Apple device must be live data. Accessories that simulate data are not permitted.

Three custom sentences in NMEA format are available: PASCD, PAGCD, and PAACD. For these sentences, the recommended sensor update rate is 10-50Hz. Data may be accumulated in a buffer for up to 1 second.

35.1.1 Global Navigation Satellite System (GNSS) Mode

Accessories that implement GNSS Mode must meet the requirements in this section.

Accessories must provide:

- GPGGA
- GPRMC

Accessories may additionally provide:

- GPGSV
- GPHDT
- PASCD

In this mode, it is recommended that accessories make use of inertial sensor and vehicle sensor technology and that the GPGGA and GPRMC information provided is derived from GNSS coupled with data from those sensors (e.g. wheel speed and gyro).

In this mode, the minimum update rate is 1 Hz.

Paywu Digital LTD
paywu@uuband.com

Latitude data from the accessory must fall within the range of -90 to +90 degrees, and longitude data from the accessory must fall within the range of -180 to +180 degrees.

NMEA Sentence Fields (page 462) lists the fields of supported NMEA sentences.

35.1.2 Sensors Mode

Accessories that implement Sensors Mode must meet the requirements in this section.

Accessories must provide:

- PASCD
- PAGCD (at least yaw rate)

Accessories may additionally provide:

- GPHDT
- PAACD

In this mode, sensor data provided by the accessory enables the Apple device's GNSS to provide position and velocity data.

In this mode, the minimum sensor update rate is 10 Hz.

For PAGCD & PAACD, the vehicle coordinate frame is defined as follows:

- x-axis pointing in the forward along-track direction of the vehicle
- z-axis pointing down (positive yaw rate is clockwise)
- y-axis defined by a right-handed convention; positive pitch angle means the vehicle is driving up-hill

NMEA Sentence Fields (page 462) lists the fields of supported NMEA sentences.

35.1.3 NMEA Sentence Fields

This section lists the fields of supported NMEA sentences. Commas must be used to delimit fields when constructing NMEA sentences. See Location Information Examples (page 469).

35.1.3.1 Standard NMEA Sentences

Fields listed as Required must be populated when the sentence is transmitted; other fields may be left empty.

Parv Wu
Unigrand LTD
parv@unigrand.com.tw

Table 35-2 GPGGA Sentence Fields

Field	Required	Description
1	yes	Sentence ID: $GPGGA
2	yes	UTC time of position (hhmmss.ss)
3	yes	Latitude of position (ddmm.mmmmmm)
4	yes	Latitude of position (N or S)
5	yes	Longitude of position (ddmm.mmmmmm)
6	yes	Longitude of position (E or W)
7	yes	Fix Quality indicator (0=invalid, 1=valid fix, 2=differential valid fix)
8	no	Number of satellites in view (xx)
9	no	Horizontal dilution of precision (x.x)
10	yes	Antenna altitude above mean sea level (x.x)
11	yes	Units of antenna altitude, must be meters (M)
12	no	Geoidal separation (x.x), recommended
13	yes	Units of geoidal separation, must be meters (M)
14	no	Age of differential GPS data in seconds (x.x)
15	no	Differential reference station ID (xxxx)
16	yes	Checksum, two characters: the bitwise exclusive OR of all characters in the sentence between $ and *

Note: If the GPGGA Fix Quality indicator is 'invalid', the sentence will be discarded by the Apple device.

Table 35-3 GPRMC Sentence Fields

Field	Required	Description
1	yes	Sentence ID: $GPRMC
2	yes	UTC time of position (hhmmss.ss)

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Field	Required	Description
3	yes	Data status (A=OK, V=Navigation receiver warning)
4	yes	Latitude of position (ddmm.mmmmmm)
5	yes	Latitude of position (N or S)
6	yes	Longitude of position (ddmm.mmmmmm)
7	yes	Longitude of position (E or W)
8	yes	Ground speed in knots
9	yes	Track made good in degrees True
10	yes	UTC date
11	no	Magnetic variation in degrees
12	no	Magnetic variation (E or W)
13	no	Mode indicator (A=Autonomous, D=Differential, E=Approximation, N=Invalid Data)
14	yes	Checksum, two characters: the bitwise exclusive OR of all characters in the sentence between $ and *

Table 35-4 GPGSV Sentence Fields

Field	Required	Description
1	yes	Sentence ID: $GPGSV
2	yes	Total number of sentences for this epoch of data
3	yes	Sentence Number
4	yes	Number of satellites in view
5	no	Satellite number
6	no	Elevation (degrees)
7	no	Azimuth (degrees)
8	no	Carrier to noise density ratio (dB-Hz)

Personal Parvwu@gmail.com Digital LTD

Field	Required	Description
9	no	Satellite number
10	no	Elevation (degrees)
11	no	Azimuth (degrees)
12	no	Carrier to noise density ratio (dB-Hz)
13	no	Satellite number
14	no	Elevation (degrees)
15	no	Azimuth (degrees)
16	no	Carrier to noise density ratio (dB-Hz)
17	no	Satellite number
18	no	Elevation (degrees)
19	no	Azimuth (degrees)
20	no	Carrier to noise density ratio (dB-Hz)
21	yes	Checksum, two characters: the bitwise exclusive OR of all characters in the sentence between $ and *

Table 35-5 GPHDT Sentence Fields

Field	Required	Description
1	yes	Sentence ID: $GPHDT
2	yes	Heading: the best estimate of the clockwise angle in degrees between True North (i.e. Geodetic North) and the forward along-track direction of the vehicle (range 0.0000-359.9999)
3	yes	Degrees true (always T)
4	yes	Checksum, two characters: the bitwise exclusive OR of all characters in the sentence between $ and *

Party Wu
Unauthorized LTD
taryw@unauthorized.com

Note: It is possible for the GPHDT angle to differ from the vehicle direction of travel as indicated in the GPRMC message. For example, if the vehicle is moving in reverse, these angles are different by 180 degrees.

35.1.3.2 Custom NMEA Sentences

All fields are required for sentences in this section unless otherwise specified.

Table 35-6 PASCD Sentence Fields

Field	Name	Type	Description
1	$PASCD	char	sentence ID for vehicle speed data
2	timestamp	float	Reference time in seconds with a resolution of 3 decimal places (for example, 12.345 seconds); timestamps increase monotonically but can roll over (for example, 0.000-86400.000 for daily rollover)
3	sensorType	char	C = combined left and right wheel speed sensors. Other types not supported.
4	transmissionState	char	U = unknown
			P = park
			R = reverse; this value should be used if reverse was engaged at any time within the interval of this data
			D = driving forward
			N = neutral
5	slipDetect	uint	1 = a wheel speed slippage was detected within the interval of this data sentence; 0 = no slip was detected
6	sampleCount	uint	the number of sensor samples included in this sentence (1-50)
7	timeOffset_i	float	the time offset in seconds of the ith sensor sample set from the reference timestamp, with a resolution of two decimal places (for example, 0.12 seconds)

Preliminary Draft Unigrand Confidential

Field	Name	Type	Description
8	speed_i	float	vehicle speed in meters/second at timeOfffset_i, with a minimum resolution of 3 decimal places (for example, 10.123 m/s)
9	Checksum	hex	Two characters: the bitwise exclusive OR of all characters in the sentence between $ and *.

Table 35-7 PAGCD Sentence Fields

Field	Name	Type	Description
1	$PAGCD	char	sentence ID for vehicle gyro data
2	timestamp	float	Reference time in seconds with a resolution of 3 decimal places (for example, 12.345 seconds); timestamps increase monotonically but can roll over (for example, 0.000-86400.000 for daily rollover)
3	sampleCount	uint	the number of sensor samples included in this sentence (1-50)
4	timeOffset_i	float	the time offset in seconds of the ith sensor sample set from the reference timestamp, with a resolution of two decimal places (for example, 0.12 seconds)
5	xAxisSample_i	float	ith x-axis sensor (degrees/second) with a minimum resolution of 4 decimal places (for example, 33.1234 deg/s); this field can be left empty if not available
6	yAxisSample_i	float	ith y-axis sensor (degrees/second) with a minimum resolution of 4 decimal places (for example, 33.1234 deg/s); this field can be left empty if not available
7	zAxisSample_i	float	ith z-axis sensor (degrees/second) with a minimum resolution of 4 decimal places (for example, 33.1234 deg/s); this field is required
8	Checksum	hex	Two characters: the bitwise exclusive OR of all characters in the sentence between $ and *.

Table 35-8 PAACD Sentence Fields

Field	Name	Type	Description
1	$PAACD	char	sentence ID for vehicle accelerometer data

Apple Confidential
taywu@quantatw.com

Field	Name	Type	Description
2	timestamp	float	Reference time in seconds with a resolution of 3 decimal places (for example, 12.345 seconds); timestamps increase monotonically but can roll over (for example, 0.000-86400.000 for daily rollover)
3	gValue	float	reference gravity magnitude in meters per second per second with a resolution of 5 decimal places (for example, 9.81001 m/s2)
4	sampleCount	uint	the number of sensor samples included in this sentence (1-50)
5	timeOffset_i	float	the time offset in seconds of the ith sensor sample set from the reference timestamp, with a resolution of two decimal places (for example, 0.12 seconds)
6	xAxisSample_i	float	ith x-axis sensor gravity reading with a minimum resolution of 4 decimal places (for example, 0.1234 g); this field can be left empty if not available
7	yAxisSample_i	float	ith y-axis sensor gravity reading with a minimum resolution of 4 decimal places (for example, 0.1234 g); this field can be left empty if not available
8	zAxisSample_i	float	ith z-axis sensor gravity reading with a minimum resolution of 4 decimal places (for example, 0.1234 g); this field can be left empty if not available
9	Checksum	hex	Two characters: the bitwise exclusive OR of all characters in the sentence between $ and *.

35.2 Location Information Usage

The accessory must enumerate what types of location information it can send during Accessory Identification. For more information, see Table 52-21 (page 585).

Once the accessory has successfully identified itself, the device will be aware that an external source of location data is available. When the device is ready to accept location information, it will send a StartLocationInformation (page 601) message to the accessory detailing what types of location data it will accept.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Note: The StartLocationInformation (page 601) message used to have a MinimumIntervalInMilliseconds parameter. That parameter is deprecated. Accessories must ignore it if it is present in received StartLocationInformation (page 601) messages.

Once StartLocationInformation (page 601) has been received, the accessory must send LocationInformation (page 602) messages containing one or more of the specified NMEA sentences. Bundling multiple sentences into one LocationInformation (page 602) message is recommended. The sentences must be sent to the device according to the rate specifications in Table 35-1 (page 460).

Once the device no longer needs external location data, it will send a StopLocationInformation (page 602) message to the accessory, and the accessory must cease sending LocationInformation (page 602) messages. The accessory must be prepared to start and stop providing location data to the device at any time.

35.3 Location Information Examples

35.3.1 GPGGA Sentence Example

```
$GPGGA,225833.0,3719.951324,N,12201.808796,W,1,09,0.6,69.8,M,-27.0,M,,*5F
```

35.3.2 GPRMC Sentence Example

```
$GPRMC,225833.0,A,3719.951324,N,12201.808796,W,0.0,26.1,060813,0.0,E,A*2F
```

35.3.3 GPGSV Sentence Example

```
$GPGSV,3,1,11,32,80,095,21,20,61,310,36,31,42,046,15,01,39,226,35*76
$GPGSV,3,2,11,23,38,285,32,16,17,147,15,11,17,210,34,13,12,268,33*74
$GPGSV,3,3,11,48,45,198,,51,44,156,,46,39,143,*4C
```

35.3.4 GPHDT Sentence Example

```
$GPHDT,0.0000,T*05
$GPHDT,359.9999,T*0A
```

Larry Wu (larry@quantaqct.com.tw)

35.3.5 PASCD Sentence Examples

10 Hz sensor data sent at 1 Hz while accelerating forward from 0 to 10 m/s (1 sentence, 139 characters long):

```
$PASCD,1000.001,C,D,0,10,0.00,0.123,0.10,1.123,0.20,2.123,0.30,3.123,0.40,
4.123,0.50,5.123,0.60,6.123,0.70,7.123,0.80,8.123,0.90,9.123*41
```

10 Hz sensor data sent at 1 Hz while accelerating in reverse from 0 to 10 m/s (1 sentence, 139 characters long):

```
$PASCD,1000.001,C,R,0,10,0.00,0.123,0.10,1.123,0.20,2.123,0.30,3.123,0.40,
4.123,0.50,5.123,0.60,6.123,0.70,7.123,0.80,8.123,0.90,9.123*57
```

10 Hz sensor data sent at 1 Hz while accelerating in reverse from 0 to 10 m/s with wheel slippage detected (1 sentence, 139 characters long):

```
$PASCD,1000.001,C,R,1,10,0.00,0.123,0.10,1.123,0.20,2.123,0.30,3.123,0.40,
4.123,0.50,5.123,0.60,6.123,0.70,7.123,0.80,8.123,0.90,9.123*56
```

50 Hz sensor data sent at 2 Hz while accelerating forward from 0 to 10 m/s (2 sentences, 304 characters each):

```
$PASCD,1000.001,C,D,0,25,0.00,0.123,0.02,0.323,0.04,0.523,0.06,0.723,0.08,
0.923,0.10,1.123,0.12,1.323,0.14,1.523,0.16,1.723,0.18,1.923,0.20,2.123,0.22,
2.323,0.24,2.523,0.26,2.723,0.28,2.923,0.30,3.123,0.32,3.323,0.34,3.523,0.36,
3.723,0.38,3.923,0.40,4.123,0.42,4.323,0.44,4.523,0.46,4.723,0.48,4.923*77

$PASCD,1000.501,C,D,0,25,0.00,5.123,0.02,5.323,0.04,5.523,0.06,5.723,0.08,
5.923,0.10,6.123,0.12,6.323,0.14,6.523,0.16,6.723,0.18,6.923,0.20,7.123,0.22,
7.323,0.24,7.523,0.26,7.723,0.28,7.923,0.30,8.123,0.32,8.323,0.34,8.523,0.36,
8.723,0.38,8.923,0.40,9.123,0.42,9.323,0.44,9.523,0.46,9.723,0.48,9.923*73
```

35.3.6 PAGCD Sentence Examples

10 Hz yaw-rate only (single axis) sensor data sent at 1 Hz, while turning at 12.1234 deg/s (1 sentence, 174 characters long):

```
$PAGCD,20000.001,10,
0.00,,,12.1234,0.10,,,12.1234,0.20,,,12.1234,0.30,,,
```

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

```
12.1234,0.40,,,12.1234,0.50,,,12.1234,0.60,,,12.1234,0.70,,,12.1234,0.80,,,
12.1234,0.90,,,12.1234*7C
```

50 Hz yaw-rate only (single axis) sensor data sent at 1 Hz, while turning at 12.1234 deg/s (2 sentences, 399 characters each):

```
$PAGCD,20000.002,25,0.00,,,12.1234,0.02,,,12.1234,0.04,,,12.1234,0.06,,,
12.1234,0.08,,,12.1234,0.10,,,12.1234,0.12,,,12.1234,0.14,,,12.1234,0.16,,,
12.1234,0.18,,,12.1234,0.20,,,12.1234,0.22,,,12.1234,0.24,,,12.1234,0.26,,,
12.1234,0.28,,,12.1234,0.30,,,12.1234,0.32,,,12.1234,0.34,,,12.1234,0.36,,,
12.1234,0.38,,,12.1234,0.40,,,12.1234,0.42,,,12.1234,0.44,,,12.1234,0.46,,,
12.1234,0.48,,,12.1234*43

$PAGCD,20000.502,25,0.00,,,12.1234,0.02,,,12.1234,0.04,,,
12.1234,0.06,,,12.1234,0.08,,,12.1234,0.10,,,12.1234,0.12,,,12.1234,0.14,,,
12.1234,0.16,,,12.1234,0.18,,,12.1234,0.20,,,12.1234,0.22,,,12.1234,0.24,,,
12.1234,0.26,,,12.1234,0.28,,,12.1234,0.30,,,12.1234,0.32,,,12.1234,0.34,,,
12.1234,0.36,,,12.1234,0.38,,,12.1234,0.40,,,12.1234,0.42,,,12.1234,0.44,,,
12.1234,0.46,,,12.1234,0.48,,,12.1234*46
```

35.3.7 PAACD Sentence Example

10 Hz x-axis and z-axis only sensor data sent at 1 Hz, (1 sentence, 236 characters long):

```
$PAACD,40000.001,9.80665,10,0.00,0.0340,,1.0000,0.10,-0.0106,,
1.0000,0.20,0.0283,,1.0000,0.30,0.0298,,1.0000,0.40,0.0412,,
1.0000,0.50,-0.0302,,1.0000,0.60,-0.0165,,1.0000,0.70,0.0268,,
1.0000,0.80,-0.0222,,1.0000,0.90,0.0054,,1.0000*7B
```

Bluetooth Pairing Wu (wu@bluetooth.com.tw)

36. Media Library Access

The media library feature allows accessories to download the metadata contents of an Apple device's media libraries (not the media items themselves) and request playback of media items. The feature is divided into three subfeatures: information, updates, and playback. Media library information informs the accessory about available media libraries on the device. Media library updates provide an accessory with an updated view of the contents of a particular media library. Media library playback allows the accessory to request playback of one or more items from the media library.

36.1 Media Library Access Requirements

Accessories that use this feature must have sufficient computational and storage resources to store and continuously update their mirror of the Apple device's media libraries. Accessories may choose not to receive certain types of media item metadata to minimize storage and computational requirements. If an accessory runs out of storage capacity while receiving media library updates it must not use this feature until sufficient storage capacity has been freed to resume receiving media library updates.

Additionally, accessories must allow the user to access and request playback of items from multiple media libraries, such as the device's main music library and the library that corresponds to iTunes Radio.

36.1.1 Media Library Information Requirements

All accessories that support the Media Library Information feature via iAP2 must send or receive the following iAP2 control session message(s):

- StartMediaLibraryInformation (page 603)
- MediaLibraryInformation (page 603)
- StopMediaLibraryInformation (page 604)

36.1.2 Media Library Updates Requirements

All accessories making use of Media Library Updates must support Media Library Information.

The accessory must implement and declare an iAP2 file transfer session as specified in File Transfer Session (page 568).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Additionally, the accessory must not request a full database update using the StartMediaLibraryUpdates (page 604) message more than once during an iAP2 connection. The accessory must be capable of processing incremental media library updates from the device and persist its media library mirror across connections to minimize the occurrence of full updates.

All accessories that support the Media Library Updates feature via iAP2 must send or receive the following iAP2 control session message(s):

StartMediaLibraryUpdates (page 604)

MediaLibraryUpdate (page 606)

StopMediaLibraryUpdates (page 610)

36.1.3 Media Library Playback Requirements

All accessories making use of Media Library Playback must support Media Library Updates.

All accessories that support the Media Library Playback feature via iAP2 may also send or receive the following iAP2 control session message(s):

PlayMediaLibraryCurrentSelection (page 610)

PlayMediaLibraryItems (page 610)

PlayMediaLibraryCollection (page 611)

PlayMediaLibrarySpecial (page 612)

PlayMediaLibraryCurrentSelection (page 610) must only be sent by the accessory in response to a direct user action, such as pressing a button on the accessory labeled 'iPod' or 'iPhone'. Similarly, PlayMediaLibraryItems (page 610), PlayMediaLibraryCollection (page 611), and PlayMediaLibrarySpecial (page 612) must only be sent by the accessory in response to a direct user action such as selecting a specific group of media items, or a media collection, and pressing a 'Play' or 'Select' button on the accessory.

All accessories must not assume that the playback queue is persistent on accessory detach.

Note: Accessories must not send Media Library Playback messages in response to any direct user actions that map to Media Remote Control HID usages (see HID Media Playback Remote Requirements (page 456)). For example, if an accessory has a 'Next Track' button, the accessory must generate a 'Next Track' HID usage when that button is pressed, not a PlayMediaLibraryItems (page 610) message.

Apple Confidential and Restricted

Autonomous generation of these messages without being preceded by direct user action is grounds for failure to pass self certification. If an accessory wishes to resume media playback upon initial connection to the Apple device, it must send a `Play` HID usage as specified in HID (Human Interface Device) (page 407).

Note that `Play` must not be sent upon connection when the accessory and the Apple device both support the CarPlay feature. See Session Establishment (page 256).

36.2 Media Library Access Usage

36.2.1 Media Library Information Usage

Accessories must send StartMediaLibraryInformation (page 603) to start receiving MediaLibraryInformation (page 603) messages. Each MediaLibraryInformation (page 603) message contains information about all available media libraries on the Apple device. The `MediaLibraryName` for each library may be used to populate the accessory display, and the accessory must use the assigned `MediaLibraryUniqueIdentifier` if it wants to receive updates or request playback for that media library.

The accessory must send StopMediaLibraryInformation (page 604) when it stops displaying or making use of media library information. For example, if an accessory supports multiple audio sources, including an Apple device, the accessory must stop receiving media library information from the Apple device if the user takes action to switch to another audio source.

36.2.2 Media Library Updates Usage

To make use of media library updates, accessories must send a StartMediaLibraryUpdates (page 604) message enumerating the media item and/or media playlist properties that they are interested in for a particular media library. MediaLibraryUpdate (page 606) messages will be sent by the device when the contents of the active media library change relative to the last database version sent to the accessory. Updates will be sent only for properties for which the accessory specifically requested updates.

Accessories must be prepared to receive duplicate MediaLibraryUpdate (page 606) messages. In such situations, the accessory does not need to apply the duplicate update. Additionally, non-actionable update messages, such as a message that instructs the accessory to delete a media item that is not present in the accessory's media library, must be ignored by the accessory. Other actions taken by the user, other devices owned by the user, or other accessories may result in those situations; the accessory must not treat them as error states.

Preliminary Draft Not to be Distributed

Accessories that register for media item and/or media playlist property updates via the StartMediaLibraryUpdates (page 604) message must process media item and/or media playlist deletions when present in received MediaLibraryUpdate (page 606) messages. Such deletions must be processed immediately and reflected in any user-visible displays. Accessory selection of a deleted media item or media playlist for playback after notice of the deletion has been received is grounds for failure to pass self certification.

The accessory must send StopMediaLibraryUpdates (page 610) when it stops displaying or making use of media library updates. For example, if an accessory supports multiple audio sources, including an Apple device, the accessory must stop receiving media library updates from the Apple device if the user takes action to switch to another audio source.

Regardless of whether the accessory sends StopMediaLibraryUpdates (page 610) or disconnects from the Apple device, the accessory must store the last received `MediaLibraryRevision` parameter value. When the accessory next sends StartMediaLibraryUpdates (page 604) to the same Apple device it must set `LastKnownMediaLibraryRevision` to that value to avoid having to retrieve a complete media library update. If the accessory has limited storage capacity it may choose to only retain a media library snapshot from one Apple device at a time.

Accessories may receive a percent completion of the current set of media library updates from the Apple device. This may be useful for informing the user of progress when there are a large number of updates from previous revision. For more information, see the `MediaLibraryUpdateProgress` parameter in StartMediaLibraryUpdates (page 604) and MediaLibraryUpdate (page 606).

If an accessory does not recognize the media library type (see Table 52-64 (page 603)), the media library must be ignored.

Accessories may receive media library updates indicating whether the Apple device's onscreen UI is not presenting remotely stored media items (such as content from iTunes in the Cloud) in certain situations like Airplane Mode. Additionally, accessories may choose to be informed whether a particular media item is resident on the Apple device or not. For more information, see the `MediaItemPropertyIsResidentOnDevice` and `MediaLibraryIsHidingRemoteItems` parameters in StartMediaLibraryUpdates (page 604), MediaLibraryUpdate (page 606), Table 52-70 (page 607), and Table 52-67 (page 604). Accessories must not assume that any particular media item is resident on the Apple device until a MediaLibraryUpdate message has been received from the device explictily indicating that it is resident. If a media library item is resident on the device, the accessory may choose to indicate that to the user it may be played in situations where the Apple device may be unable to play other non-resident media library items, such as when the device has poor network connectivity.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

36.2.3 Media Library Playback Usage

Accessories may make use of Media Library Playback to either play a media library item collection (such as a playlist) or even an arbitrary set of media items that has been assembled by the accessory. Every media item in that set must be obtained from Media Library Updates.

To resume playback of a library's current selection, the accessory must send PlayMediaLibraryCurrentSelection (page 610).

To start playback of specific media items or a media library collection, the accessory must send PlayMediaLibraryItems (page 610) or PlayMediaLibraryCollection (page 611), respectively. The library's current selection will be replaced with the new items or collection contents.

Music, Podcast, Audiobooks and iTunesU media items should not be mixed in the list of items to play using PlayMediaLibraryItems (page 610).

The number of items that can be specified for playback in the PlayMediaLibraryItems (page 610) message is bounded by the maximum possible size of an iAP2 message. Accessories should keep the list of media items to a reasonable length.

To start playback of all songs in the media library, the accessory should use the PlayMediaLibrarySpecial (page 612) message using the `AllSongs` parameter. The library's current selection will be replaced.

Additionally, accessories that request playback of media items must not anticipate or assume corresponding state changes by the Apple device. For further explanation, see Presentation of Apple Device Updates (page 49).

Apple Inc. Confidential
Authorized for Wu Chi-Hsiang, Pegatron Technology (Shanghai) LTD
Wu_Chi-Hsiang@pegatron.com.tw

37. Musical Instrument Digital Interface (MIDI)

Starting with iOS 4.2.1, some Apple devices provide support for USB Host Mode MIDI. Compatible accessories can interface directly with iOS apps that make use of the Core MIDI framework.

To support MIDI, accessories must follow the requirements in *Universal Serial Bus Definition for MIDI Devices*, Release 1.0, available at www.usb.org. Developers should test their accessory designs against the latest OS X MIDI driver; the application Audio MIDI Setup, in the OS X Applications/Utilities folder, can be used to verify accessory compatibility.

Every accessory that supports the USB Host Mode MIDI must implement a MIDI Streaming IN endpoint.

Party Wu
Unigrand LTD
party.wu@unigrand.com.tw

38. Now Playing Updates

The Now Playing feature enables an accessory to display information about the current "Now Playing" media source and media item on an Apple device. Media sources include both the built-in Music and Video apps on Apple devices and certain third-party iOS apps that support the generation of Now Playing metadata (see *MPNowPlayingInfoCenter* in the iOS SDK documentation). Accessories must be prepared for the Now Playing media source and media item to change at any time, whether the accessory requested the change or not.

38.1 Now Playing Updates Requirements

All accessories that support the Now Playing feature via iAP2 must send or receive the following iAP2 control session message(s):

StartNowPlayingUpdates (page 612)

NowPlayingUpdate (page 614)

StopNowPlayingUpdates (page 617)

All accessories that support this feature must demonstrate correct handling of Now Playing information from the following Apple-developed iOS apps during self certification:

- Music
- Videos
- Podcasts
- iBooks
- iTunes U

If the accessory will register for album artwork associated with the now playing media item, the accessory must implement and declare a file transfer session as specified in File Transfer Session (page 568). All album artwork is transmitted to the accessory in JPEG format.

Party Wu Digital and LTD
party@digitalandltd.com.tw

38.2 Now Playing Updates Usage

A StartNowPlayingUpdates (page 612) message from the accessory to the device starts the generation of NowPlayingUpdate (page 614) messages from the device. The accessory must declare what types of Now Playing information changes are desired in StartNowPlayingUpdates (page 612). There are two major types of Now Playing information changes: changes in the now playing media item, and changes in the state of the media playback engine. In StartNowPlayingUpdates (page 612), the accessory must specify exactly which media item attributes and playback engine state changes it is interested in to receive those updates.

The very first NowPlayingUpdate (page 614) message will be sent immediately after the Now Playing feature is started, and will contain a complete list of media item attributes and playback engine states. All subsequent messages will be sent only when one or more attributes or states change, and the message will only contain the new values. Accessories must assume the last reported attribute or state value remains constant until it is overwritten by a NowPlayingUpdate (page 614) message.

Accessories that interact with iTunes Radio must handle all associated media item attributes (see Table 52-83 (page 615)).

On average, NowPlayingUpdate (page 614) messages will be sent by the Apple device once a second. However, this interval can vary and must not be relied on to drive accessory behavior such as display updates.

The accessory must send a StopNowPlayingUpdates (page 617) message to the device once it no longer needs these updates.

Pay Wu
Beyond LTD
beyond@ms1.hinet.net

39. Power

The Power feature enables accessories to:

- Report their power characteristics.
- Receive Apple device power status.
- Provide power to an Apple device.
- Draw power from an Apple device.

Accessories are defined as:

- Self-powered, if capable of providing their own power (e.g., from a battery or directly from an external power source).
- Device-powered, if capable of drawing power from the Apple device.
- Power-providing, if capable of providing power to the Apple device.

39.1 Power Requirements

All accessories must support the Power feature.

Accessories that provide power to or draw power from an Apple device must connect to the Apple device either through an integrated Lightning connector or a cable as described in Cables with USB Connectors (page 50) or Cables with Non-USB Connectors (page 51).

All accessories that do not have the potential for data communication with the Apple device must use a USB Device Mode connector configuration and provide power to the Apple device. See Connector Pad/Pin Configuration (page 125) for more details on the USB Device Mode connector configuration. See Providing Power to the Apple Device (page 481)

Additionally, certain accessory interface features require the accessory to provide power to the Apple device.

Otherwise, accessories may provide power to the Apple device (Providing Power to the Apple Device (page 481)), draw power from the Apple device (Drawing Power From the Apple Device (page 486)), do both (in certain situations), or do neither (Neither Providing Power nor Drawing Power (page 481)).

Part Number Unigrand Limited

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Apple strongly recommends providing power to the Apple device whenever possible for the best user experience.

39.1.1 Neither Providing Power nor Drawing Power

Accessories that do not provide power to or draw power from an Apple device (as in the case of a wireless accessory) must declare this during Accessory Identification, see Accessory Identification of Power Capabilities (page 207).

These accessories may implement the device power status feature.

All accessories that support the Device Power Status feature via iAP2 may also send or receive the following iAP2 control session message(s):

- StartPowerUpdates (page 617)
- PowerUpdate (page 618)
- StopPowerUpdates (page 619)

39.1.2 Providing Power to the Apple Device

Accessories that provide power to an Apple device must implement an accessory power source.

Accessory power sources must provide power at all times unless direct user action (button press, power switch, menu item selection, detach of external power supply, etc.) is taken to put the accessory into an 'off' state. Failure to provide power at all times may result in the accessory being unable to charge an Apple device whose battery level is too low for the device to boot and is grounds for failure to pass self certification.

Accessory power sources must establish an iAP2 connection over the Lightning connector and declare their capability to provide power (see IdentificationInformation (page 579) the `PowerProvidingCapability`, `USBHostTransportComponent`, `USBDeviceTransportComponent`, and `SerialTransportComponent` parameters).

All accessories that support the Accessory Power Source feature via iAP2 must send or receive the following iAP2 control session message(s):

- StartPowerUpdates (page 617)
- PowerUpdate (page 618)
- StopPowerUpdates (page 619)
- PowerSourceUpdate (page 619)

Partially Unsigned iAP2 Document - Unauthorized Copy - Property of Apple Inc. - Do Not Distribute

39.1.2.1 External Power Source

If the accessory can accept power from an external power source, it must report the available current.

See Integrated USB Receptacles (page 51) and User Supplied Cables and Power Supplies (page 52).

39.1.2.2 Power Adapter Converter Switching Frequencies

To be compatible with the frequency-hopping touch sensors in Apple devices, every AC or DC power adapter design must conform to the following requirements for its converter switching frequencies:

- To avoid interference with audio output, the switching frequency must always be greater than the audio band (that is, more than 22 kHz) for all loads greater than 5 mA.
- The switching frequency must always be above 60 kHz, and preferably above 450 kHz, for all loads greater than 20 mA.

39.1.2.3 Power Adapter Noise Reduction Using a YCAP AC Capacitor

AC adapter control switching frequencies are much higher than power line frequencies. They or their harmonics can easily interfere with the touch sensor modulation frequencies in an Apple device. It is strongly recommended that any AC adapter design for an Apple device include a YCAP AC capacitor (up to 1000 pF) between the primary and secondary sections of the adapter's transformer to reduce common-mode noise at these higher switching frequencies.

39.1.2.4 Power Adapter Impedance Stability

The diodes used in an AC adapter's full-wave bridge rectifier can be a major source of abrupt changes in the adapter's series impedance. To reduce unwanted touch sensor output oscillations, the AC adapter circuit should be designed such that its series impedance does not change abruptly.

If the AC adapter bridge diodes have large inherent reverse capacitance (greater than 100 pF, as many large power diodes do), then the net impedance change due to diode switching may be acceptably small; it will not adversely affect the touch sensor output. In more compact IC designs, however, the chip area of each diode may be reduced in size and its reverse capacitance may become correspondingly smaller.

To stabilize the impedance of bridge diodes with unacceptably low reverse capacitance, follow the example shown in Figure 39-1 (page 483). In this example, capacitors C1, C2, C3, and C4 have been placed in parallel with diodes D1, D2, D3, and D4 to stabilize the bridge impedance. Their values are larger than the inherent reverse capacitances of the diodes.

Resistors R1, R2, R3, and R4 are optional; if included, they can block noise at very high frequencies, which can help with EMI compatibility. The suggested values of R1, R2, R3, and R4 shown were chosen to have trivial levels of impedance relative to the impedances of C1, C2, C3, and C4 at power line frequencies.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 39-1 Typical AC adapter diode bridge circuit

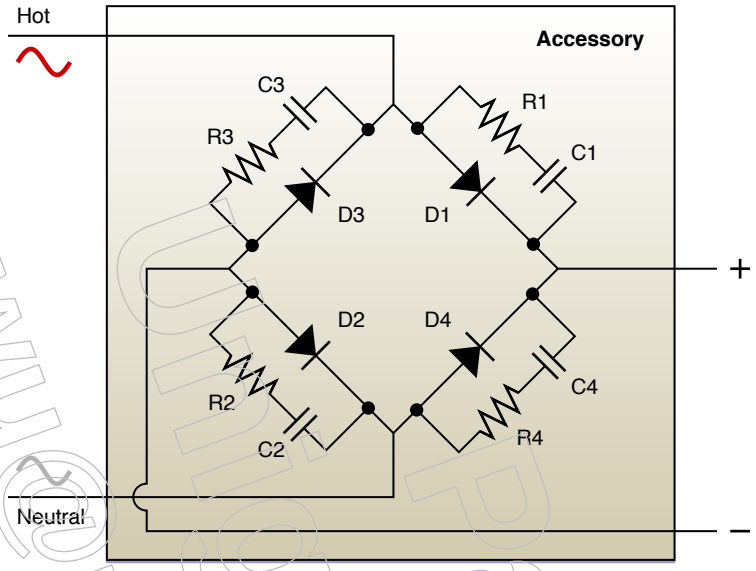


Table 39-1 Typical component values for an AC adapter diode bridge circuit

Component	Value
C1, C2, C3, C4	47 pF
R1, R2, R3, R4	2 kΩ

39.1.2.5 Fuse Protection

A fuse should be present at the input of the power supply accessory circuitry to protect it under any fault condition.

39.1.2.6 Short Circuit Response

The output of the power supply accessory should drop or fold back if its output is shorted to the secondary common, and no damage should result. A short circuit is defined as less than 10 milliohms resistance.

39.1.2.7 Lightning Connectors on Power-Providing Accessories

All Lightning connectors on a power-providing accessory must meet the following requirements under load:

- The Lightning connector must supply 2100 mA at 4.55 V to claim compatibility with iPad and iPad mini. 2400 mA at 5.0 V is recommended.
- The Lightning connector must supply 1000 mA at 4.70 V to claim compatibility with iPhone and iPod. 2100 mA at 5.0 V is recommended.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

39.1.2.8 Non-Lightning Connectors on Power-Providing Accessories

All power-providing accessory connectors (such as a USB Type A receptacle) designed for use with a separate cable that terminates in a Lightning connector must meet the following requirements, regardless of whether the cable is included with the accessory or provided by the end user:

- The connector must supply 2100 mA at 4.97 V to claim compatibility with iPad and iPad mini. 2400 mA at 5.2 V is recommended.
- The connector must supply 1000 mA at 4.90 V to claim compatibility with iPhone and iPod. 2100 mA at 5.2 V is recommended.
- The connector must meet or exceed all applicable USB-IF specifications, both mechanical and electrical (or only electrical if the connector is not a standard USB connector).

39.1.2.9 Cables Terminating in Lightning Connectors

All accessory cables that terminate in a Lightning connector must meet the following requirements:

- The cables must not share or split power between the Lightning connector and any other connector.
- The cables must meet or exceed all applicable USB-IF specifications.
- The cables must have DC resistances below the limits shown in Table 39-2 (page 484).
- The cables must exhibit no more inductance than a simple ferrite bead on every circuit.
- Every cable shield must be connected to at least one of the ground pads on the Lightning connector.
- All captive cables must short the shields to the accessory's ground path.

Table 39-2 USB cable maximum DC resistances

Specification	Maximum DC resistance
Round-trip V_{BUS} with shield shorted to GND at each end	200 mΩ ($R_{GND\|\|SHIELD}$)
V_{BUS} conductor alone	160 mΩ (R_{VBUS})
Ground conductor alone	140 mΩ (R_{GND})

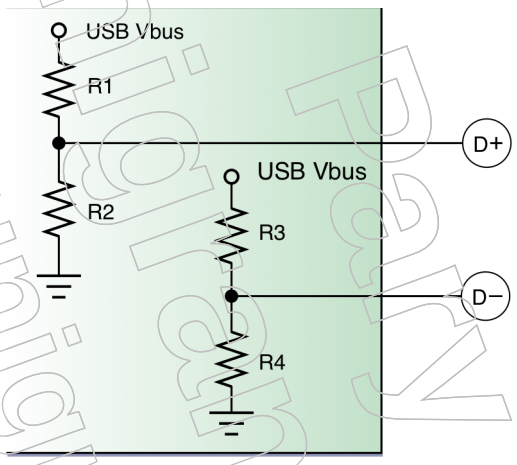
Taiwan Apple Users Group
applehelp@applehelp.com.tw

39.1.2.10 Connectors That Do Not Implement iAP2

All accessory connectors that are capable of providing power to the Apple device, but do not implement an iAP2 connection, must connect the USB D+ and USB D- pins to resistor networks as shown in Figure 39-2 (page 485). These resistors must be present on the D+ and D- pins at the time the Apple device is connected without requiring user action such as moving a switch or pressing a button.

Additionally, all such connectors must supply between 4.9 V to 5.25 V on the VBUS (Device Power) pin when there is no current being drawn.

Figure 39-2 USB D+/D- resistor networks for power-providing accessory connectors that do not implement iAP2



Note: All resistors used to implement the networks specified in Figure 39-2 (page 485) must have a tolerance of 1% or better.

Table 39-3 USB D+/D- resistor values for power-providing accessory connectors that do not implement iAP2

Current	R1	R2	R3	R4
2400 mA	43.2 kΩ	49.9 kΩ	43.2 kΩ	49.9 kΩ
2100 mA	43.2 kΩ	49.9 kΩ	75.0 kΩ	49.9 kΩ
1000 mA	75.0 kΩ	49.9 kΩ	43.2 kΩ	49.9 kΩ

Samuel@unigrandi.com Unigrandi Peiyu Wu LTD

39.1.2.11 Multiple Connectors

Note: Every Apple device-compatible connector on an accessory that uses D+/D- resistors must have its own set of resistors. The accessory must be capable of supplying the total current required when all connectors are in use, regardless of whether the ports are compatible with Apple devices or not.

If the accessory has standard USB type-A connectors supplying 500 mA that could be used to provide power to an Apple device in addition to Apple device-compatible USB type-A connectors, then the following labeling requirements apply:

- The standard USB type-A connectors supplying 500 mA must be labeled using the USB icon.
- USB type-A connectors capable of identifying themselves to an Apple device as supplying 1000 mA must be labeled, singly or in groups, with the text 'iPod/iPhone'.
- USB type-A connectors capable of identifying themselves to an Apple device as supplying 2100 mA or 2400 mA must be labeled, singly or in groups, with the text 'iPad'.

If the accessory has multiple Lightning connectors with different device compatibilities, then the iPad-compatible connectors must be labeled with the text 'iPad' unless it is physically impossible to connect an iPad to the iPod/iPhone compatible connectors.

39.1.2.12 Overcurrent Protection

For safety reasons, an accessory providing power to an Apple device must detect any nontransient current drain of more than 2.9 A. The accessory must immediately cut off its power supply, after which it may perform a power-up reinitialization. This over-current detection and shut-off circuitry must reset itself without mechanical intervention.

39.1.3 Drawing Power From the Apple Device

Any accessory that draws power from the Apple device at any time is considered to be a device-powered accessory, even if it contains an accessory power source that can provide power to the Apple device. All device-powered accessories must implement both Low Power and Intermittent High Power operational modes.

Pad 5 (Accessory Power) on the Lightning connector can supply up to 100 mA at 2.9-3.4 V ±5% (2.71 V to 3.57 V).

The current drawn by the Apple Authentication Coprocessor does not count towards the current limits. The Authentication Coprocessor must be put into a sleep state if it is not being used for accessory authentication.

Apple Confidential
Licensed for Wei Qian (weiqian@ftdichip.com)

All device-powered accessories must establish an iAP2 connection over the Lightning connector and declare their maximum power draw during Accessory Identification (see IdentificationInformation (page 579) `MaximumCurrentDrawnFromDevice`, `USBHostTransportComponent`, `USBDeviceTransportComponent`, and `SerialTransportComponent` parameters).

All accessories that support the Device Powered Accessory feature via iAP2 must send or receive the following iAP2 control session message(s):

StartPowerUpdates (page 617)

PowerUpdate (page 618)

StopPowerUpdates (page 619)

39.1.3.1 Low Power Mode Requirements

The following table specifies how much current an accessory in Low Power Mode may draw from an Apple device, depending on its iAP2 transport component:

Table 39-4 Maximum allowable Low Power Mode current draw

iAP2 Transport	Maximum Current Draw
None	0 mA
Serial	5 mA
USB Device Mode	0 mA
USB Host Mode (Full Speed)	10 mA
USB Host Mode (High Speed)	30 mA
Bluetooth	0 mA

39.1.3.2 Intermittent High Power Mode Requirements

When an accessory is in Intermittent High Power mode it may draw additional current from the Apple device. An accessory in Intermittent High Power mode must not ever draw more than 100 mA from the Apple device.

Party Wu LTD
Unofficial@unofficial.com.tw

39.2 Power Usage

Accessories must send a StartPowerUpdates (page 617) message with at least one parameter if they will draw power from or provide power to the Apple device and must react to any received PowerUpdate (page 618) messages.

The Apple device may send PowerUpdate (page 618) messages at any time.

Accessories may send a StopPowerUpdates (page 619) message only if they are no longer providing power to or drawing power from the Apple device.

39.2.1 Providing Power to the Apple Device

If the accessory can provide power to the Apple device from its own power source, it must send the IdentificationInformation (page 579) message with a `PowerProvidingCapability` parameter value of `Advanced`.

The accessory must send a PowerSourceUpdate (page 619) message to the device:

- Within 1 second after Accessory Identification (page 205) has completed successfully. The first PowerSourceUpdate (page 619) message sent after identification must contain the `AvailableCurrentForDevice` and `DeviceBatteryShouldChargeIfPowerIsPresent` parameters.
- Whenever the accessory power source state changes, for example:
  - When power is not available or available at a reduced level (e.g., when the user unplugs the accessory from the wall socket).
  - When power is once again available or the level has increased (e.g., when the user plugs the accessory back into a wall socket).

The `DeviceBatteryShouldChargeIfPowerIsPresent` parameter in PowerSourceUpdate (page 619) messages must be `true` for the Apple device's UI to indicate to the user that it is charging its internal battery from the accessory power source.

The device may send PowerUpdate (page 618) messages to the accessory in response to PowerSourceUpdate (page 619) messages indicating how much current the device intends to draw from the accessory power source. If the accessory power source is providing more current than the device is drawing then the accessory may use the extra power for its own purposes.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

39.2.2 Drawing Power from the Apple Device

All device-powered accessories must identify their maximum possible current draw in Intermittent High Power Mode via the `MaximumCurrentDrawnFromDevice` parameter in IdentificationInformation (page 579). Also, they must start operation in Low Power Mode.

Accessories supporting Intermittent High Power Mode must include the `AccessoryPowerMode` parameter when sending the StartPowerUpdates (page 617) message.

39.2.2.1 Entering Intermittent High Power Mode

An accessory may enter Intermittent High Power Mode (maximum 100 mA current draw) if it receives a PowerUpdate (page 618) message from the device with a corresponding `AccessoryPowerMode` parameter. Accessories that provide Location updates to an Apple device may be permitted to enter Intermittent High Power Mode when the Apple device requests those updates. Similarly, accessories that implement an iAP2 EA Session may enter Intermittent High Power Mode when the Apple device opens an EA session with the accessory.

When an Apple device is in USB Host Mode, an attached accessory in Low Power mode may also enter Intermittent High Power mode upon receipt of any of the following USB events:

Table 39-5 USB Events that permit Intermittent High Power Mode

USB Device Type	Event
Audio	The Apple device selects a nonzero bandwidth interface setting
EA Native Transport	The Apple device selects a nonzero bandwidth interface setting
MIDI	The Apple device starts polling a MIDI Streaming IN endpoint

If an accessory receives permission to enter Intermittent High Power Mode via multiple mechanisms, the highest power limit overall applies.

39.2.2.2 Exiting Intermittent High Power Mode

An accessory must exit Intermittent High Power Mode and re-enter Low Power Mode within 1 second if it receives a PowerUpdate (page 618) message from the device with a corresponding `AccessoryPowerMode` parameter.

Similarly, if an Apple device is in USB Host Mode, an attached accessory in Intermittent High Power Mode must re-enter Low Power Mode within 1 second after all of the USB events that permitted Intermittent High Power Mode are negated by one or more of the following USB events:

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 39-6 USB Events that exit Intermittent High Power Mode

USB Device Type	Event
Audio	The Apple device selects a zero bandwidth interface setting
EA Native Transport	The Apple device selects a zero bandwidth interface setting
MIDI	The Apple device stops polling a MIDI Streaming IN endpoint

39.3 Test Procedures

39.3.1 Accessory Power Sources

39.3.1.1 Output Voltage Regulation

Test conditions: At an ambient temperature of 25°C, the load on the power supply accessory's output must be increased in steps to produce the following output currents: 0, 250, 500, 700, 1000, 1500, 2100, and 2400 mA. The accessory should be tested at all currents up to and including the maximum current it supports. If the accessory is a car charger, the test must be repeated with input voltages of 10, 12, and 14 V DC. The output voltage must be noted for each combination of input voltage and output current.

Pass/Fail criteria: For every combination of input voltage and output current, the output measured at the dock connector of the power supplies listed in Table 39-7 (page 490) must remain between the minimum and maximum voltages shown.

Table 39-7 Power Supply Voltage Outputs

Supply rating	Minimum output	Maximum output
500 mA or 1 A	4.70 V	5.25 V
2.1 A delivering up to 1.0 A	4.70 V	5.25 V
2.1 A delivering 1.0 to 2.1 A	4.55 V	5.25 V
2.4 A delivering 1.0 to 2.4 A	4.97 V	5.25 V

Paul Wu
LDU
paulwu@pudu.com.tw

39.3.1.2 Ripple and Noise

Test conditions: Ripple voltage and noise must be measured at each output pin of the Lightning connector while the output is decoupled by a high-frequency 0.2 uF capacitor. The measurement bandwidth must be 0 to 20 MHz.

Pass/Fail criteria: Periodic and Random Deviation (PARD) on each output pin must be less than 100 mV while the power supply accessory is loaded to produce the following output currents: 0, 250, 500, 700, 1000, 1500, 2100, and 2400 mA. The accessory should be tested at all currents up to and including the maximum current it supports.

39.3.1.3 Dynamic Load Response

Test conditions:

- The power supply accessory must be subjected to 500 mA load increases and decreases.
- Power supplies that provide at least 1 A must also be subjected to load increases and decreases between 80 mA and 1 A.
- Power supplies that provide at least 2.1 A must be subjected to increases and decreases from 0 to 500 mA, 500 to 1000 mA, 1000 to 1500 mA, and 2100 mA.
- Power supplies that provide at least 2.4 A must be subjected to increases and decreases from 0 to 500 mA, 500 to 1000 mA, 1000 to 1500 mA, and 2100 to 2400 mA.
- In all tests the load slew rate must not be less than 100 mA/usec.

The frequency of change must be set to give the most readable deviation and settling times. If the accessory is a car charger, the input voltage must be 12 V DC. No load capacitors may be used to stabilize the output.

Pass/Fail criteria: With a decoupling capacitor of 10 uF and an ambient temperature of 25° C, the power supply accessory output, as measured at the dock connector, must not undershoot below the minimum output voltage or overshoot above the maximum output voltage for each type of power supply listed in Table 39-7 (page 490)

39.3.1.4 Overload Protection

Overloads applied to the power supply output, at a rate faster than 100 mA/µsec, should cause the output to disconnect before they cause damage to the accessory or the Apple device, and the output should remain disconnected until the overload is removed. This design goal for overload protection also applies to any single fault condition. It must not be possible for a power supply to provide more than 2.5 A rms in any overload situation.

Paul Unigard Pauluni@unigard.com

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

39.3.1.5 Overvoltage Protection

Taking into account the delay time of the overvoltage protection circuit, no single-point fault should be able to cause a sustained overvoltage condition on the power supply output. The power supply should provide a latch-mode overvoltage protection circuit that resets itself within 30 seconds. Optionally, a power off/on cycle could restore normal operation. An overvoltage fault can be simulated by opening the feedback loop that regulates the output voltage. A voltage greater than 6.3 V must not be possible during a single-point failure.

39.3.1.6 Car Charger Tests

39.3.1.6.1 Input Voltage Surge

Test conditions: While the charging accessory is operating at both maximum load and minimum load, the line voltage must be switched 5 times to the surge voltage in a 50% duty cycle, as follows:

1. From 12 V DC to 40 V DC; hold for 16 msec.
2. Back to 12 V DC; hold for 16 msec.
3. Repeat 5 times.

Pass/Fail criteria: The accessory's power supply must survive the repeated applications of the input voltage surge with no component damage. Voltages and logic signals must remain within specification limits during and after the line transients (making it a "transparent surge"). Loss of function or performance must not occur. Permanent damage must not occur. A fuse opening is a failure.

39.3.1.6.2 Dynamic Line Response

Test conditions: The car charger must be subjected to instantaneous ±2 V DC input variations. The frequency of change must be set to give the most readable deviation and settling times. Output loads must be chosen to produce worst-case conditions. No load capacitors may be used to stabilize the output.

Pass/Fail criteria: The car charger output, as measured at the dock connector, must not undershoot below 4.75 V or overshoot above 5.25 V.

39.3.1.6.3 Turn-On and Turn-Off Characteristics

Test conditions: 12 V DC input voltage must be applied to and removed from the car charger input, with 5 µF capacitive loading on the output.

Pass/Fail criteria: The following measurements must all be within the limits stated:

1. **Turn-on delay:** The car charger output must settle to a value between 4.75 and 5.25 V within 4 seconds after the DC input is applied.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

2. **Output rise time:** The time between 10% and 90% of the nominal output voltage must be ≤ 20 ms.
3. **Overshoot:** The output voltage overshoot upon application or removal of the input voltage must be less than 10% of the nominal voltage.
4. **Voltage ramp:** There must be a smooth and continuous ramp of the DC output voltage from 10% to 90% of its final value within a range of 4.75 to 5.25 V, while the output is loaded as specified above under "Test conditions." Voltages of opposite polarity must not be present on the output at any time during turn-on or turn-off.

39.3.1.7 USB Receptacle Load Test

This test is applicable to accessories that provide power to an Apple device via a USB receptacle.

The following equipment is required:

- Programmable DC load
- USB break out board
- Digital multimeter

To perform the load test:

1. Set the programmable DC load to the appropriate load.
   - For accessories that claim compatibility with iPod/iPhone, set the DC load to 1.0 A.
   - For accessories that claim compatibility with iPad, set the DC load to 2.1 A.
2. Connect the USB break out board to the accessory's USB type A receptacle.
3. Configure the digital multimeter to measure voltage, and connect it to the USB break out board at VCC and GND.
4. Connect the DC load to the USB break out board at VCC and GND.
5. Read the voltage from the digital multimeter.
   - For accessories that claim compatibility with iPod/iPhone, the voltage must be higher than 4.90 V.
   - For accessories that claim compatibility with iPad, the voltage must be higher than 4.97 V.

Party Unigrand LTD
party@unigrand.com.tw

40. Serial

The Serial transport may make sense for low-cost accessories that do not need to transmit audio, or for device-powered accessories which emphasize low power draw over high data transmission rates. Otherwise, the USB Host Mode transport is recommended.

Apple devices support serial communication based on the RS-232 serial specification with the exception of signaling levels. For Apple devices, a mark is 2.500 V through 3.57 V and a space is 0 V through 0.8 V.

Table 40-1 Serial transport mark and space levels for Apple devices

Description	Conditions	MIN	MAX	Units
Accessory TX Voltage High		2.500	3.57	V
Accessory TX Voltage Low		0.000	0.800	V
Accessory TX Current	Accessory TX Voltage = 0 V or 3.0 V		±30	µA
Device TX Voltage High	Accessory TX Voltage High = 100 µA	2.500	3.57	V
Device TX Voltage Low	Accessory TX Voltage Low = -100 µA	0.000	0.500	V

Accessories must communicate over the serial transport at 57600 bps and maintain their baud rates within ±2% of the chosen rate over the entire temperature range of the accessory. Accessories must not change baud rate once a Serial communication has started.

The accessory must not rely on the Apple device to hold the Device TX level to the mark state when the Serial transport is idle. When idle, Device TX line must be pulled up to Accessory Power through a 100 kΩ resistor.

Accessories that use an open-collector or open-drain UART driver on Accessory TX must include a pull-up resistor to Accessory Power. The pull-up resistor value must be chosen to meet the baud rate requirements described above. The maximum mark voltage must be measured with the Apple device not connected, and mark signals must be sent only after Accessory Power goes high.

Paul Pilgrim (pilgrim@unimelb.edu.au)

All serial communications use 8 data bits, no parity bits, and one stop bit (8-N-1). Serial hardware flow controls (RTS/CTS and DTR/DSR) are not used and will be ignored by the Apple device. In addition, the accessory must not use software flow control (XON/XOFF). The accessory must not use bit averaging to produce a mean bit rate not directly achievable from its system clock. All bits transmitted by the accessory must have the same nominal duration.

Pary Wu
Unigrand LTD
pary.wu@unigrand.com.tw

41. Siri

41.1 Enabling Custom Siri Commands

Every accessory that works with Siri must support HFP Command AT+XAPL (page 233). The Apple device will use the information sent by this command to enable and disable custom commands related to Siri.

To receive Siri status events, the accessory must send the AT+XAPL command after making a successful HFP Service Level Connection (SLC) to the Apple device. The accessory must send an AT+XAPL command first, before sending any of the additional Siri-specific commands described below.

41.2 Obtaining Siri Availability Information

After establishing an HFP profile connection, an accessory can determine if Siri is available and enabled on an Apple device. It can also receive notifications of changes in Siri status. If Siri is disabled, Voice Control will be activated instead.

41.2.1 Obtaining Status Information at Connection

The accessory must send the following command after making a successful HFP profile (SLC) connection and sending an AT+XAPL command.

41.2.1.1 HFP Command AT+APLSIRI?

Description: AT command to retrieve Siri status information.

Initiator: accessory

Format: AT+APLSIRI?

Response: +APLSIRI:*value*

Defined Values:

- 0 = Siri is not available on this platform.
- 1 = Siri is available and enabled.
- 2 = Siri is available but not enabled.

Patent Engine LTD Harvey Wu (harvey@patentengine.com.tw)

Example: `+APLSIRI:1` (Siri is available and enabled)

41.2.2 Receiving Siri Availability Updates from the Apple Device

After initialization has been completed, the Apple device will send the accessory the following notification if there is a change in Siri status. This notification will be provided only if the accessory has requested Siri status (by sending `AT+APLSIRI?`) at least once after connection, and if the Apple device has reported that Siri is available and enabled.

41.2.2.1 HFP Command +APLSIRI

Description: Unsolicited event indicating a change in Siri status.

Initiator: Apple device

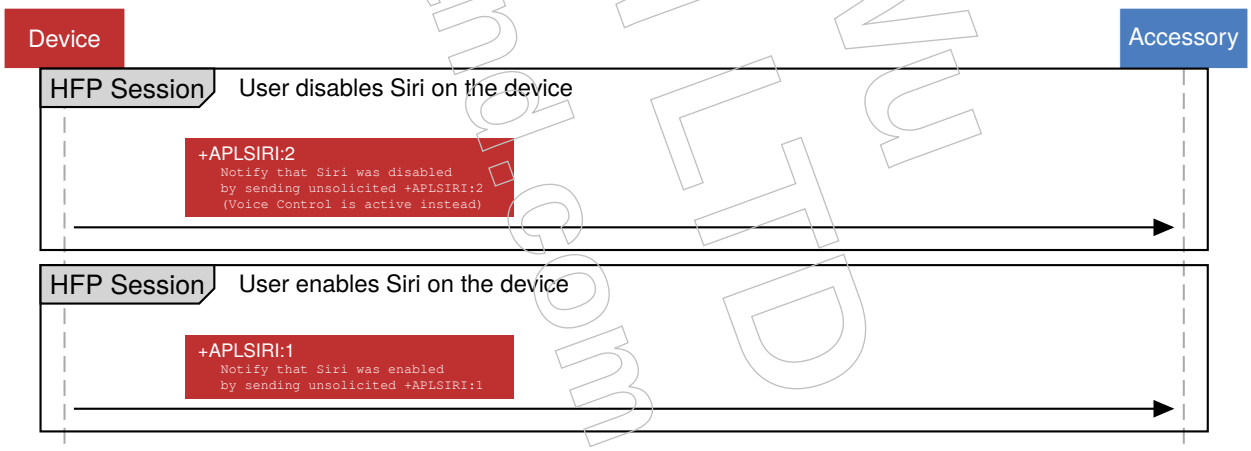
Format: `+APLSIRI:`*value*

Defined Values:

- 1 = Siri is available and enabled.
- 2 = Siri is available but not enabled.

Example: `+APLSIRI:2` (Siri is available but not enabled)

Figure 41-1 Siri is Disabled/Enabled from the Apple Device's Settings



41.3 Initiating a Siri Session

Once support for Siri is established on both the accessory and the Apple device, a Siri session can be started from either one.

Party Wu LTD paul@unigrandinc.com.tw

41.3.1 Initiating a Session from the Accessory

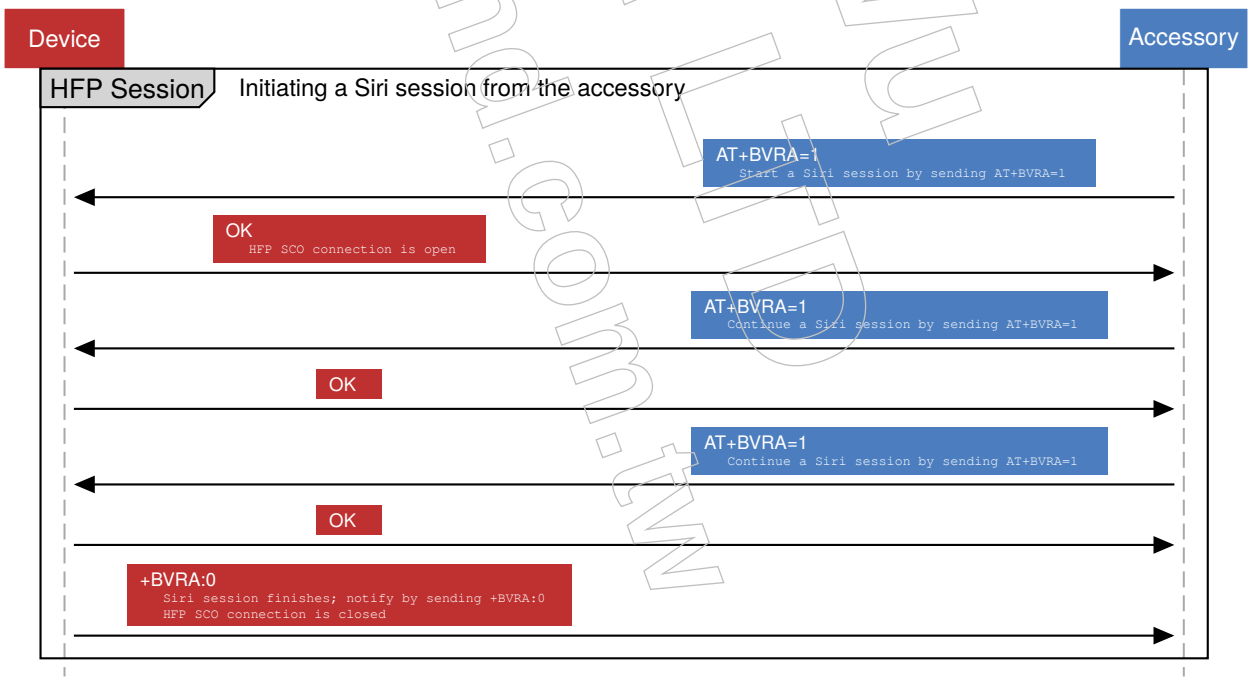
To initiate a Siri session, the accessory must use the voice recognition command AT+BVRA defined in the Bluetooth *Hands-Free Profile* specification. For further details, see the Bluetooth *Hands-Free Profile* 1.6 profile specification, section 4.25. The HFP profile must be connected and SLC must exist.

The accessory must use the following command sequence:

- The accessory sends an AT+BVRA=1 command to the Apple device.
- The Apple device sends an OK response.
- The Apple device launches a Siri session and creates a Synchronous Connection (SCO) for the audio.
- If the Siri session is not finished, the accessory must send AT+BVRA=1 to continue the conversation. This may need to happen multiple times.
- When the Siri session is finished, the Apple device sends a +BVRA:0 result code to the accessory.
- The Apple device disconnects the SCO connection.

While a Siri session is active, the accessory must let the user continue the conversation and ask follow up questions within the current context. In order to do so, the accessory must be able to send an AT+BVRA=1 command to the Apple device even after Siri has been already activated and before +BVRA:0 is received. Figure 41-2 (page 498) shows an overview of the interaction when Siri is triggered from the accessory, the running session was continued twice and once Siri was finished, the device dismissed the session.

Figure 41-2 Initiating a Siri Session from the Accessory



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

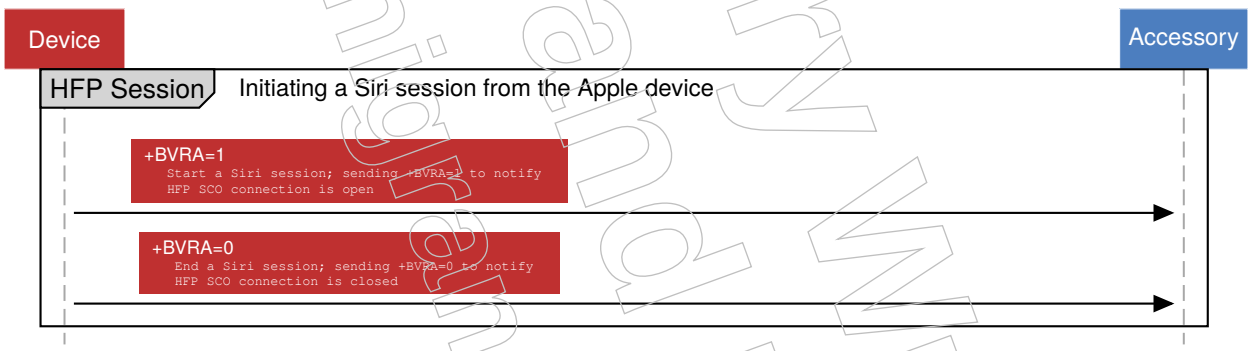
41.3.2 Initiating a Session from the Apple Device

If the accessory supports voice recognition commands, the Apple device sends a +BVRA event to indicate the start of a Siri session. The accessory must enable support for voice recognition and indicate it in its feature response as described in the Bluetooth *Hands-Free Profile* 1.6 specification, section 4.34.1, "Bluetooth Defined AT Capabilities." Specifically, the HFP profile must be connected, SLC must exist, and voice recognition activation (bit 3) must be enabled in the AT+BRSF command. The Apple device will not use virtual call functionality for the Siri session if voice recognition activation is supported by the accessory.

The accessory must expect the following command sequence:

- The Apple device sends a +BVRA:1 event to the accessory.
- The Apple device launches a Siri session and creates a SCO connection for the audio.
- When the Siri session is finished, the Apple device sends a +BVRA:0 result code to the accessory.
- The Apple device disconnects the SCO connection.

Figure 41-3 Initiating a Siri Session from the Apple Device

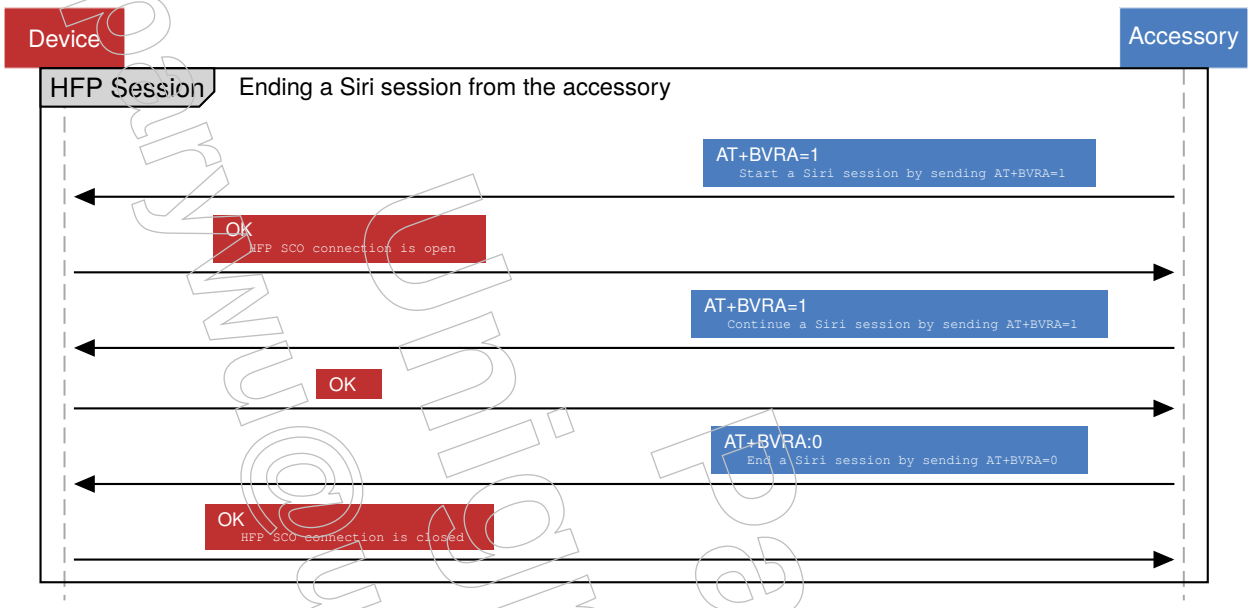


41.3.3 Ending a Session from the Accessory

Once a Siri session is running the accessory must be capable of ending the session by sending an AT+BVRA=0 command to the Apple device. Figure 41-4 (page 500) shows an example of ending a running Siri session from the accessory. The accessory must only end an active session as a direct result of a user action.

Confidential
Property of Pay Wu
pay.wu@ginlong.com.tw

Figure 41-4 Ending a Siri Session from the Accessory



41.4 Siri Eyes Free Mode

Siri Eyes Free mode is a feature to control Siri responses that include display information and can be enabled or disabled as needed. In Eyes Free mode, the user experience is tailored towards a driving scenario and interactions with Siri are done primarily via voice to minimize the need for the user to look at a screen. Siri Eyes Free mode is supported only for Bluetooth-enabled vehicle entertainment systems and must not be used by any other accessories.

The Apple device will listen for the HFP AT command `AT+APLEFM` to enable or disable Eyes Free mode.

This command is used by the Apple device to modify Siri responses that contain visual information or require user interaction. Suitable audio feedback and voice commands will be available to the user based on the Siri use case that was initiated.

Eyes Free mode is disabled by default. Once the accessory has enabled Eyes Free mode, it remains enabled for all subsequent Siri sessions initiated from the accessory until the accessory disables it or the Bluetooth connection is disconnected.

41.4.1 HFP Command AT+APLEFM

Description: An accessory sends this command to notify an Apple device of the preferred state of Eyes Free mode.

Initiator: accessory

Property of Universal Electronics Inc. Unauthorized use or distribution prohibited

Format: `AT+APLEFM=`*value*

Response: `OK`

Defined Values:

- 0x00 = Disable Eyes Free mode.
- 0x01 = Enable Eyes Free mode.
- 0x02-0xFF = reserved

Example: `AT+APLEFM=1`

41.5 Improving Voice Recognition

The microphone audio that the accessory sends to the Apple device during a Siri session must be suitable for voice recognition. Audio requirements for optimal voice recognition may differ from requirements for optimal human perception (such as during a cellular phone call).

Filtering of the audio signal to remove echoes or feedback noise is acceptable.

To provide the best possible audio quality as Siri input, the accessory must observe the following recommendations:

- **Echo cancellation and noise suppression (EC/NR)**: Directional microphones and linear beamforming with microphone arrays giving improved SNR are recommended. Linear echo cancellation for reducing unwanted audio sources (such as audio output from the system) without having any other effect on the speech signal are also recommended. However, single channel noise reduction methods (such as spectrum subtraction) must not be applied, as they will be detrimental to the speech recognition accuracy. Similarly, automatic gain control, residual echo suppression and attempts to blank out non-speech periods in the waveform must not be applied.
- **Signal gain**: When adjusting signal levels, the accessory must avoid artifacts, dropouts, and clipping in all circumstances. Automatic Gain Control is not recommended. If the accessory adjusts signal gain, the gain must be held constant across each spoken utterance. A recommended target is to maintain speech level at A-weighted -26 dBFS RMS.
- **Signal-to-noise ratio (SNR)**: An average SNR greater than 20 dB is recommended. Below 20 dB, recognition rates will be impacted.
- **Reverberation**: Maintaining RT60 time at less than 200 msec is recommended.

Apple Confidential
Unauthorized Distribution Prohibited

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

41.5.1 Wide Band Speech Support

An accessory using Siri must support 16 kHz wide band speech audio for better audio quality and voice recognition performance. See the Bluetooth *Hands-Free Profile* 1.6 specification for details about wide band speech audio. Narrow band audio signal (8 kHz) is supported but not recommended.

41.6 Optimizing the Siri Experience

For best results in using Siri, the accessory design must follow these guidelines:

- The start of a Siri session must not be accompanied by local beeps or verbal indications (such as an announcement of "...voice dialing...") from the accessory. When a Siri session becomes active, the Apple device sends two beeps indicating that Siri is ready to receive instructions. Adding extra audible notifications only inserts delays in the system.
- The accessory must wait for the Apple device to end each Siri session. The accessory must not send an `AT+BVRA=0` command unless it is prompted to do so by user interaction.
- The Apple device expects that the accessory is capable of rendering audio as soon as the SCO connection is active. This is necessary to make sure that the user always hears the Siri introductory beeps with minimum delay. The delay should be within 200 ms.

41.7 Common Siri Applications

Siri can send messages, find points of interests, place phone calls, and much more. As Siri capabilities are constantly growing, additional use cases may become available after the initial integration. In Eyes Free mode, some of these use cases may not be accessible as the user experience is tailored towards a driving scenario.

41.7.1 Initialization Procedure After Connection is Established

Figure 41-5 (page 503) outlines the sequence the accessory has to trigger to be able to use Siri on an Apple device. After establishing an HFP profile connection, the accessory must first enable the custom Siri commands by sending `AT+XAPL` and provide the features it supports. After a confirmation is received from the device, the accessory must determine Siri's availability with `AT+APLSIRI?`.

Vehicles with Bluetooth-enabled infotainment systems can also enable Siri Eyes Free Mode during initialization. This is detailed in Figure 41-6 (page 503).

Licensed to Wu Yongqiang (wuyongqiang@hotmail.com)

Figure 41-5 Siri Initialization Procedure

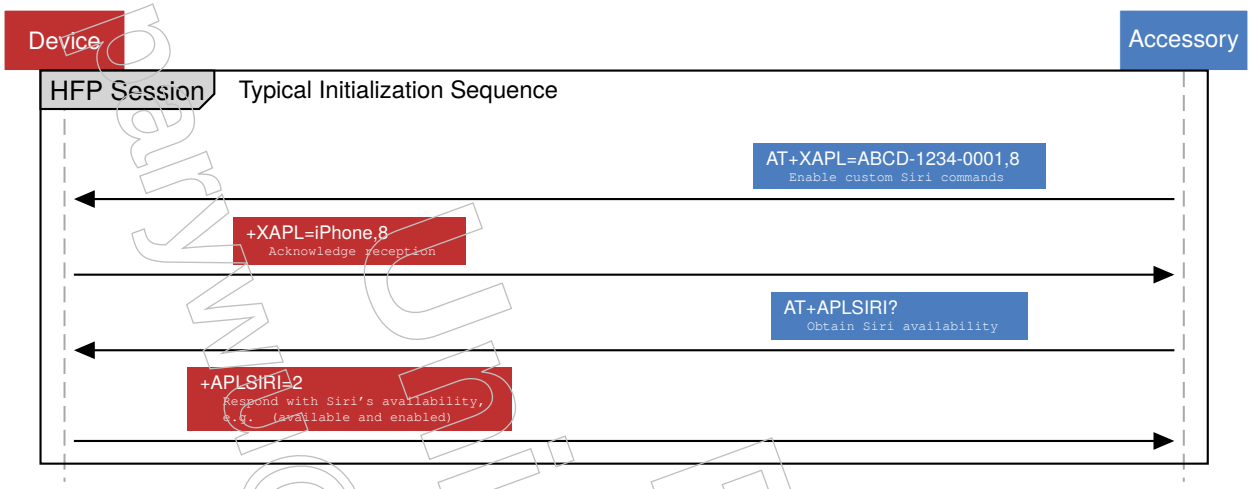
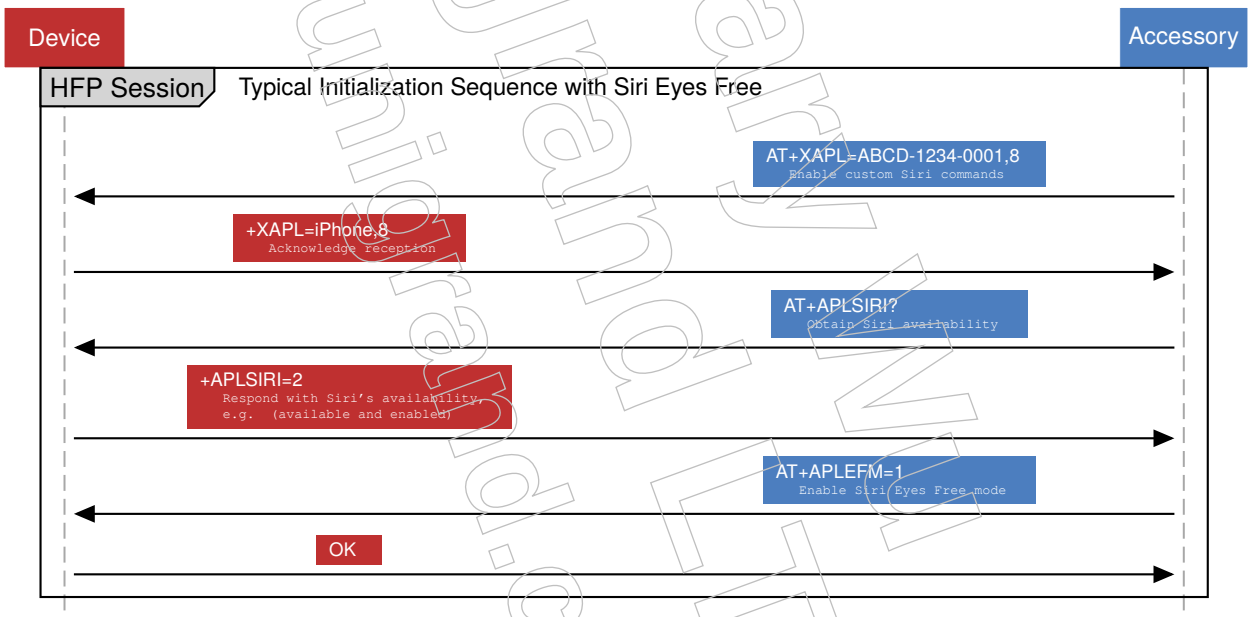


Figure 41-6 Siri Initialization Procedure with Siri Eyes Free



41.7.2 Phone Dialing Using Siri

Upon the user's request Siri can initiate an outgoing phone call. The Apple device will initiate HFP call signaling to establish a phone call as described in Bluetooth (page 216). The accessory must be able to transition to Hands-Free dialing at any time during or after a Siri session when signaled by the Apple device.

Party Unlimited
party@unlimited.com.tw

41.7.3 Audio Routing and Media Playback Using Siri

Siri can control the media playback on an Apple device, and if Siri determines that the user wants to play or pause music it will either start, pause or resume media playback. The Apple device will send a notification to the accessory indicating a change in playback state and any associated track information. The accessory must respond to those notifications, start or stop the music playback as requested, as well as update the correct playback state (e.g. shuffle, repeat).

The accessory must not force a change in the playback state after a Siri session is ended. If music was playing before Siri was started, it must continue playing, if it was paused, it must remain paused.

After Siri starts music playback the accessory must set its current audio route to match the audio source, depending on how audio is being received from the Apple device (via Bluetooth or by a wired connection).

The available media playback notifications depend on the audio route being used:

- For a Bluetooth audio route, use the approach described in Notifications (page 222).
- For a wired audio route, use the iPod Accessory Protocol, specifically the iAP2 Now Playing Updates (page 478) feature, iAP1 Display Remote Lingo, or iAP1 Extended Interface Mode.

41.7.4 Turn-By-Turn Directions Using Siri

Siri can initiate active route guidance that will play turn-by-turn directions. In case the Apple device is the active source and is already playing music, turn-by-turn directions will be mixed in as part of the audio stream. In case the Apple device is not playing music, the accessory must be able to mix in turn-by-turn directions with the active audio source.

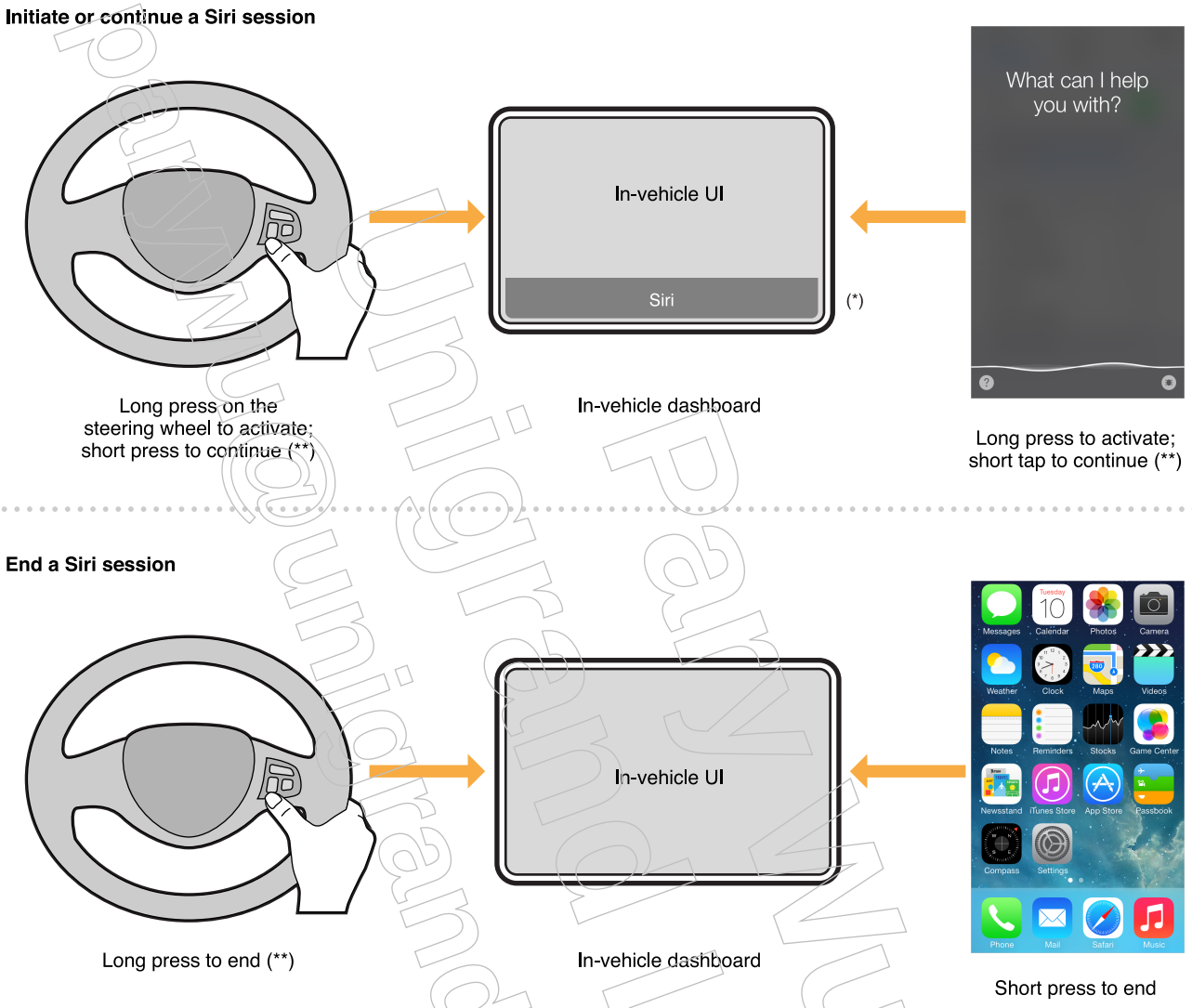
The Apple device will notify the accessory to play turn-by-turn directions only over Bluetooth. Detailed information on how to distinguish between music playback and turn-by-turn notifications is available in Notifications (page 222).

41.8 User Interaction with Siri Eyes Free in a Vehicle

A vehicle that uses Siri Eyes Free mode must integrate the Siri experience with the existing in-vehicle entertainment system and controls. The vehicle must provide a convenient interface to initiate, continue, and end a Siri session. Once a Siri session is running, the vehicle must display a visual cue that voice recognition is in process. Figure 41-7 (page 505) outlines how a Siri interaction must be designed.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 41-7 Siri Eyes Free User Interaction



(*) If the accessory wishes to indicate that Siri is active, it must do one of the following:

- Display the word 'Siri' (as capitalized) with no additional text or icon.
- Use generic text or icon that does not resemble the Siri microphone icon and does not infringe Apple trademarks.

(**) If the vehicle is equipped with steering wheel controls, a dedicated button or a long-press on a button on the vehicle's steering wheel is required to start, continue and end a Siri session. If no steering wheel controls are available, a soft button must be available within the in-vehicle user interface to start, continue or end a Siri session.

Apple Confidential and Proprietary Information

When a vehicle enables Siri Eyes Free mode, the Apple device will not display any onscreen Siri content. If the device was locked at the time the Siri session was activated from the vehicle, it will remain locked and the screen will not turn on. If the user unlocks or manually activates the device while in an Eyes Free Session there will be a notification that the device is in an active Siri session but there will be no visual Siri content displayed.

41.9 Enabling/Disabling Siri from the Apple Device

The user has the ability to disable or enable Siri from the Settings menu on the Apple device. When Siri is disabled, Voice Control becomes the recognition engine on the device and will be triggered by default. The accessory may choose to either launch Voice Control with no further changes, in the same way Siri is launched (shown in Figure 41-8 (page 506)) or display a warning message and not send an activation command to the Apple device (Figure 41-9 (page 506)).

Figure 41-8 Siri is Deactivated - Launching Voice Control

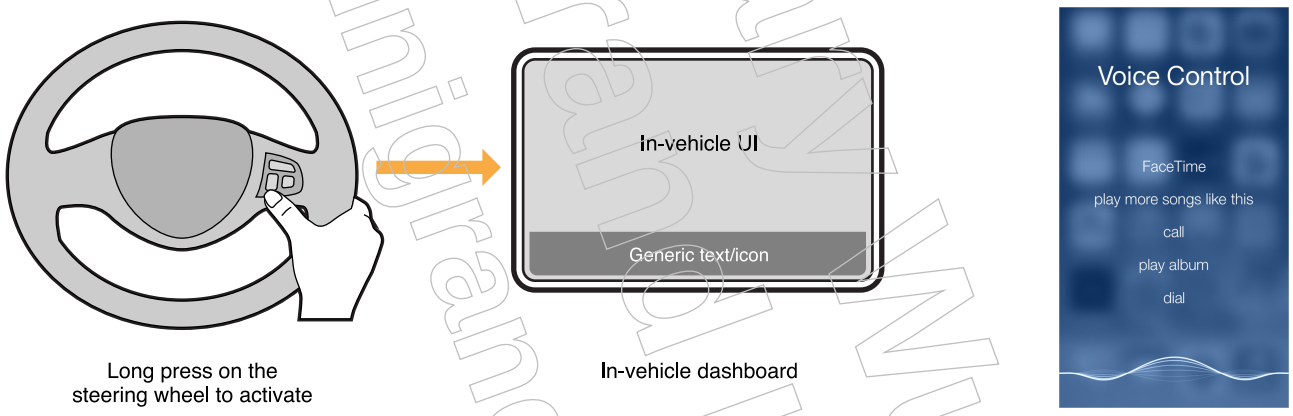
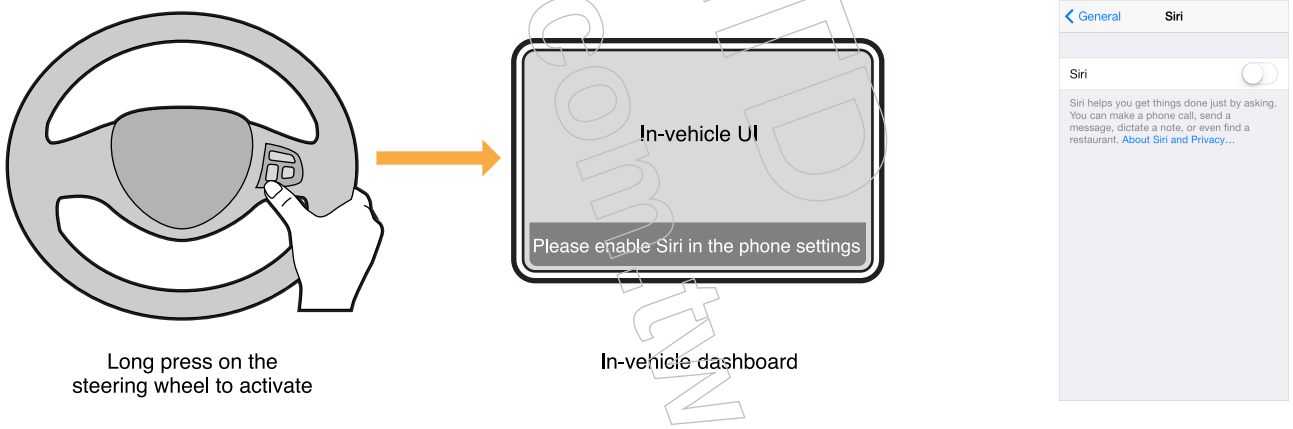


Figure 41-9 Siri is Deactivated - Displaying a Warning Message



Tony Wu (tonywu@ideashower.com) Ideas Ltd.

41.10 Test Procedures

41.10.1 Siri Eyes Free

The following test procedures are applicable to accessories that interact with Siri Eyes Free.

For the following spoken tests, the speaker should ideally be a native speaker of North American English. If the tester's native language is not English, set Siri to your native language and translate the phrases to be spoken below into your native language.

41.10.1.1 General

1. Pair and establish a Bluetooth Hands Free Profile (HFP) connection between the iPhone and the head unit. Activate Siri from the vehicle steering wheel button (e.g. by press hold):
   a. Observe that the iPhone screen remains inactive after a Siri session has started (a purple status bar will be visible on the Apple device if the screen is activated manually).
   b. Ensure that Siri opening chime is heard completely through the vehicle speakers.
   c. Observe a visual notification in the in-car User Interface (UI) that a Siri session is active (textual notification, on-screen UI, etc.).
2. Activate Siri from the vehicle steering wheel button and say "Send a message to Peter. How are you?". While still saying the message press the vehicle steering wheel button to cancel Siri (e.g. by press hold):
   a. Ensure that the Siri closing chime is heard completely through the vehicle speakers.
   b. Ensure the iPhone screen remains inactive (if manually activated, the purple bar on the phone will disappear).
   c. Verify that the in-car UI for Siri interaction dismisses and the head unit resumes the state before Siri's interaction.
3. Activate Siri from the vehicle steering wheel button and say "How is the weather in San Francisco?". Wait for Siri to respond with the weather forecast. Once the weather forecast is complete, resume Siri from the vehicle steering wheel button and say "What about New York?":
   a. Observe that the purple bar is still active on the phone.
   b. Listen for the Siri opening chime and closing chime.
   c. The vehicle UI should be displaying an on-going Siri session.
   d. Verify that Siri responds with the weather forecast for New York.
4. In case the vehicle UI offers on-screen controls to activate/cancel/resume Siri, repeat steps (1) to (3) for all on-screen controls.
5. Activate Siri from the steering wheel button and say "What's the time". Listen to the current time and do not interact with Siri or the iPhone. After 5 seconds have expired:

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

a. Observe that the purple bar on the phone is no longer visible.

b. The in-car UI for Siri interaction should be dismissed.

c. The head unit should resume the state before Siri's interaction.

6. Listen to FM radio from the car speakers (no A2DP streaming active). Press and hold phone's home button to activate Siri from the phone:

a. Ensure that the Siri closing chime is heard completely through the vehicle speakers.

b. Observe a visual notification in the in-car UI that a Siri session is active (textual notification, on-screen UI, etc.).

c. Observe Siri's interaction on the phone's screen and ask "What's the time?"

d. After Siri has responded, lock the phone again to dismiss the Siri session by pressing the phone's sleep/wake button.

7. On the phone go to Settings and turn off Siri. Activate Siri from the head unit. Observe one of the following depending on the actual implementation (a) Voice Control starts instead of Siri (b) The head unit displays a warning that Siri Eyes Free is not available.

8. On the phone go to Settings and turn Siri back on. Verify that Siri can be activated/cancelled from the head unit and from the Home button on the phone.

9. Turn Bluetooth off from the Settings on the phone. Verify that Siri cannot be started.

10. Turn Bluetooth back on from the Settings on the phone. Verify that Bluetooth HFP profile reconnects and that Siri can be activated/cancelled from the head unit and from the Home button on the phone.

11. Make sure there is no accessory battery status level indicator icon displayed on the phone's status bar.

41.10.1.2 Siri Dialog

1. Activate Siri from the vehicle's steering wheel button and say "Send a text message to **insert contact name**". When Siri prompts for "what would you like it to say", dictate a short message. After Siri has read back your dictated message, say "Review it". After Siri has read back the message again, say "Review it" again. Repeat this cycle ~5 times to ensure that the head unit is able to handle a long interaction with Siri. At the end say "Send it" and verify that the message is sent. Verify that all opening and closing beeps are audible and the message is sent. After the Siri session is closed, the audio playback should go back to the state it was before Siri was started, i.e. if it was paused remains paused, if it was playing remains playing.

2. Start Siri from the vehicle's steering wheel button and ask for directions. Follow up through the dialog until the navigation is started. Verify that the Siri session is closed and that the audio playback goes back to the state it was before Siri was started, e.g. if it was paused remains paused, if it was playing remains playing.

Apple Confidential
Preliminary Draft

3. Start Siri from the vehicle's steering wheel button and say "Search the web for polar bears". Verify that Eyes Free mode is on and that this use case is blocked by Siri. Note: In some implementations the vehicle has to be in motion before Eyes Free is activated by the car kit.
4. Start Siri from the vehicle's steering wheel button and say "What is the current time in Munich?". After Siri has answered but before ~5 seconds have elapsed, resume Siri (e.g. by a short press on the steering wheel button) and verify that Siri is initiated again. Say "What about San Francisco?". Repeat (with a different city) and verify that this can continue indefinitely as long as you short press on the steering wheel button within 5 seconds of the last response.

41.10.1.3 Bluetooth HFP A2DP Music

1. Establish a Bluetooth A2DP connection and switch to Bluetooth audio source on the head unit. Activate Siri and ask "Next track". Verify that track advances and that audio is playing through vehicle speakers. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
2. Activate Siri and say "Pause the music". Verify that audio remains paused after Siri has been dismissed. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
3. Pause music playback on the head unit (via AVRCP command). Activate Siri and ask "What time is it?". Verify that the music playback remains paused after the Siri session has been dismissed. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
4. Switch to FM radio on the head unit. Activate Siri and ask "Play me a song". Verify that head unit is able to automatically switch to BT audio and iPhone music starts playing. Verify that the beginning of the selected track is heard, e.g. there is no skipping of audio packets. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
5. Activate Siri and ask "Shuffle all songs". Verify that head unit correctly updates NowPlaying track information. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
6. Activate Siri and ask to play a specific artist or title. Verify that the Siri session is dismissed after the music starts. Make sure the correct metadata is displayed on the screen. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

41.10.1.4 Call

1. Activate Siri and call a contact with more than one phone number (home and mobile). Wait for Siri's response to ask for which phone number to call. Answer with "home". Verify that call transition is handled correctly by the head unit and any Siri UI displayed on the vehicle screen is dismissed.
2. While iPhone music is playing, activate Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback resumes after the call has been answered and terminated on the far end. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

3. While iPhone music is playing, start Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback resumes after the call has been answered and terminated on the near end (i.e. on the head unit). Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
4. While in a Siri session, receive an incoming call on the head unit. Verify that head unit handles call-signaling correctly and transitions to phone UI once the call has been accepted. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

41.10.1.5 Bluetooth + Wired iAP

1. Connect Apple device to head unit via 30-pin or Lightning connector (iPhone 4s / iPhone 5 respectively). Switch to iPod music and verify that audio is playing. Activate Siri and say "Next track". Verify that track advances and head unit displays track metadata correctly. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
2. From the head unit UI, select a playlist with a single song and start playing it. Start Siri from the vehicle steering wheel and say "Play ..... **make sure to select a song to play that is (a) not in the same album as the single-track playlist and (b) not song track index 0 of its album**". Verify that the new song starts playing and that the head unit displays the track metadata for this new song correctly. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
3. Turn Shuffle off on the head unit UI. Then start Siri and say "Shuffle all songs". Verify that the shuffle indicator on the head unit UI is updated and the correct track metadata for the new now playing song is displayed correctly Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
4. Switch to FM radio on the head unit. Activate Siri and say "Play me a song". Verify that head unit is able to automatically switch to iPOD audio source and that audio starts playing through the speakers. Verify that there is no skipping of audio at the beginning of the selected track. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
5. Pause music playback on the head unit (via iAP commands). Activate Siri and ask "What time is it?". Verify that music playback remains paused after Siri session has been dismissed. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
6. While iPhone music is playing, start Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback resumes after the call has been answered and terminated on the far end. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
7. While iPhone music is playing, start Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback resumes after the call has been answered and terminated on the near end (i.e. on the head unit). Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

Confidential Property of Apple Inc.

8. Pause music playback on the head unit (via iAP commands). Start Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback remains paused after the call has been answered and terminated on the far end. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

Pary Wu
Unigrand LTD
paryWu@unigrand.com.tw

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

42. Telephony

The Telephony feature permits accessories to receive telephony and call status updates from Apple devices that can make and/or receive voice calls.

42.1 Telephony Requirements

All accessories that support the Telephony feature via iAP2 must send or receive the following iAP2 control session message(s):

StartTelephonyCallStateInformation (page 621)

TelephonyCallStateInformation (page 621)

All accessories that support the Telephony feature via iAP2 may also send or receive the following iAP2 control session message(s):

StopTelephonyCallStateInformation (page 622)

StartTelephonyUpdates (page 623)

TelephonyUpdate (page 623)

StopTelephonyUpdates (page 625)

42.2 Telephony Usage

Accessories supporting the Telephony feature must identify themselves as sending and receiving all required messages associated with the feature. If the device does not support the Telephony feature then it will reject all identification attempts that include Telephony messages.

Accessories may use the TelephonyCallStateInformation (page 621) message to be notified when the Apple device starts or ends a call. The StartTelephonyCallStateInformation (page 621) and StopTelephonyCallStateInformation (page 622) messages must be sent for the accessory to start and stop receiving call updates.

Party Wu parywu@uniband.com Uniband LTD

One initial TelephonyCallStateInformation (page 621) message will be sent by the Apple device to the accessory for each call that is active at the time when StartTelephonyCallStateInformation (page 621) is received. If there is no call in progress, a TelephonyCallStateInformation (page 621) message will be sent with a state of Disconnected and direction of Unknown as a placeholder.

The accessory may optionally indicate that it can process vCards in its StartTelephonyCallStateInformation (page 621) message. If so, the Apple device may send a vCard to the accessory upon connecting to an incoming call. The accessory must be capable of handling multiple simultaneous vCard transfers, and a transfer of zero length or an empty vCard may occur in cases when suitable information cannot be obtained for an incoming call.

The vCard will conform to the vCard 3.0 specification and contain one or more of the following properties:

- BEGIN
- VERSION
- PRODID
- N
- FN
- TEL
- END

Accessories may use the TelephonyUpdate (page 623) message to be notified when the status of the Apple device changes. The StartTelephonyUpdates (page 623) and StopTelephonyUpdates (page 625) messages must be sent for the accessory to start and stop receiving Telephony updates.

Telephony signal strength will always be reported as zero when the Apple device is in Airplane Mode.

Pary Wu
Unigrand LTD
paywu@unigrand.com.tw

43. USB

Apple devices support both USB Device Mode (see USB Device Mode (page 522)) and USB Host Mode (see USB Host Mode (page 527)) transports. USB Host Mode is recommended. Most, if not all, future development of accessory interface features will occur on the USB Host Mode transport.

43.1 USB Host Mode Passthroughs

Accessories must not facilitate the connection of arbitrary USB peripherals to an Apple device in USB Host Mode.

43.2 USB Signal Integrity

USB Host Mode accessories and USB Device Mode accessories must pass the USB signal integrity test procedures (see Signal Integrity (page 514)).

Accessories should use tunable USB PHYs to maximize their chances of passing the tests.

Accessories that support USB High-Speed must:

- Support test packet mode.
- Detect the test Vendor ID (0x1A0A) and Product ID (0x0104) and trigger test packet generation.

43.3 Test Procedures

Test procedures for both USB Device Mode and USB Host Mode accessories are contained in this section.

43.3.1 Signal Integrity

The purpose of this signal integrity (SI) procedure is to provide clear information on logic levels, jitter budget, rise time, and fall time for USB signals transmitted by embedded hosts at the near end test point (at USB A connector).

This test procedure must be performed using the configuration of a typical user. It must not be performed:

Parvus@unigrand.com.tw Unigrand LTD Parvus

- on an incomplete system,
- on development hardware,
- using custom connectors,
- using short cables,
- using a boosted signal.

This test procedure addresses the following:

- High-Speed USB signal logic levels, rise time, fall time, and jitter budget requirements
- Full-Speed USB signal logic levels, rise time, fall time, and jitter budget requirements
- Tools required for testing
- Setup, test procedures and complete system test bed definitions

These tests must be run by product developers or 3rd party test labs. The full test report must be submitted in order to obtain certifications.

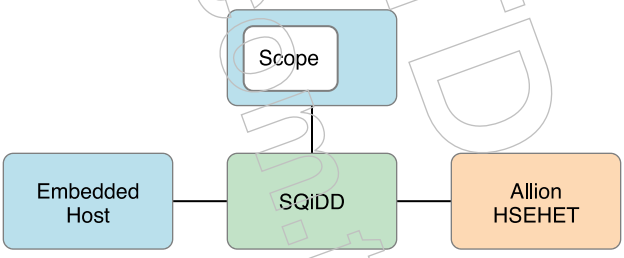
43.3.2 High-Speed USB

The following are required for validating USB host High-Speed (HS) signal integrity:

- Allion HSEHET board
- SQiDD board (TDSUSBF USB2.0 test fixture)
- USB Compliance scope with TDSUSB2 application

A proper interface to the SQiDD board must be used to take eye diagrams.

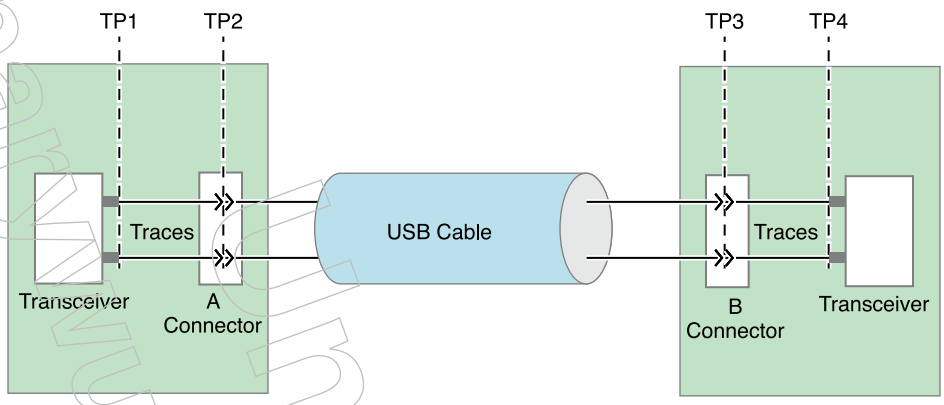
Figure 43-1 High-Speed Test Environment



All measurements must be done at the USB Type A connector at test point TP2 near the host as shown in Figure 43-2 (page 516). See the USB 2.0 specification.

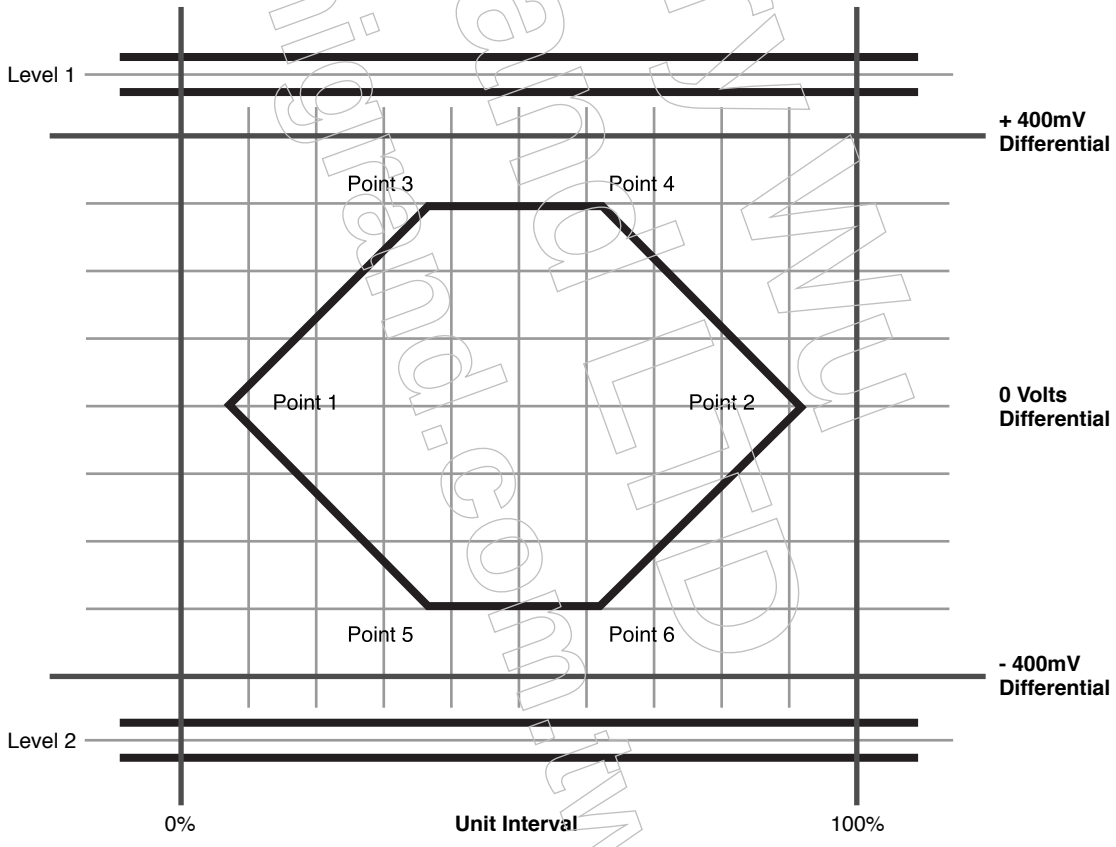
2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 43-2 USB Test Points



The transmit waveform from the USB host/hub at TP2 must meet the requirements in Figure 43-3 (page 516) and Table 43-1 (page 517) (see the USB 2.0 specification). Table 43-2 (page 517) is a summary of pass/fail criteria for a given host under test.

Figure 43-3 Eye Diagram Template at TP2



Parwuier@uniandand.com

Table 43-1 Eye Diagram Template Specifications

	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition -475 in all others	N/A
Point 1	0 V	7.5% UI
Point 2	0 V	92.5% UI
Point 3	300 mV	37.5% UI
Point 4	300 mV	62.5% UI
Point 5	-300 mV	37.5% UI
Point 6	-300 mV	62.5% UI

Table 43-2 Pass/Fail Criteria for Host Under Test

Req. #	Requirement	Value
SI_HS_01	USBDP/DM Mask Levels	+/- 300 mV
SI_HS_02	Rise Time	500 ps
SI_HS_03	Fall Time	500 ps
SI_HS_04	Jitter	50 ps
SI_HS_05	Signal Rate	479.76 Mbps - 480.240 Mbps

The host must continuously transmit a test packet (see Table 43-3 (page 517)) within inter-packet gap limits of 88 bits and <125 usec.

Table 43-3 Test Packet Layout

NRZI Symbol (Fields)	NRZ Bit Strings	Number of NRZ Bits
{KJ * 15}, KK (SYNC)	{00000000 * 3}, 00000001	32
KKJKJKKK (DATA0 PID)	11000011	8
JKJKJKJK * 9	00000000 * 9	72
JJKKJJKK * 8	01010101 * 8	64

Paul Wu
Universal Serial Bus Interface Engineer
paulwu@unigraf.com

NRZI Symbol (Fields)	NRZ Bit Strings	Number of NRZ Bits
JJJJKKKK * 8	01110111 * 8	64
JJJJJJJJKKKKKKKK * 8	0, {111111S * 15}, 111111	97
JJJJJJJK * 8	S, 111111S, {0111111S * 7}	55
{JKKKKKKK * 10}, JK	00111111, {S0111111 * 9}, S0	72
JJJKKKJJKKKKJKK (CRC16)	0110110101110011	16
JJJJJJJJ (EOP)	01111111	8

43.3.2.1 Test Procedure (HS Signal Quality EL_2, EL_3, EL_6, EL_7)

Perform the following steps:

1. Connect Host to SQiDD board, set switch in "Init" mode.
2. Connect Scope differential probes to SQiDD board with proper orientation.
3. Set the SW2 on Allion HSEHET board to position "4" Test Packet PID (0x0104).
4. Connect the Allion HSEHET board to the SQiDD board. Wait 15 sec.
5. Set the switch on SQiDD board to apply terminations.
6. Run the TDSUSB application on scope with near end setting.
7. Capture waveform and analyze with TDSUSB application.
8. Save the HS eye diagram, waveforms and USBIF report.

Accessories supporting USB Role Switch must be tested in both USB Host Mode and USB Device Mode. The accessory should transmit test packets for 5 minutes, role switch, and transmit test packets for 5 minutes again. An indicator such as an LED can be helpful to indicate this change.

43.3.2.2 Naming Report

High-Speed accessory file name format:

```
HostName_Model#_FirmwareVersion_HSEHET_Date_PASS/FAIL_MFi_Report.html
```

Accessories supporting USB Role Switch must submit separate reports corresponding to USB Device Mode and USB Host Mode.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

43.3.2.3 Pass/Fail Criteria

The report generated by TDSUSB will indicate failure if there is a mask, timing or jitter failure. Submit the report as is without any modifications.

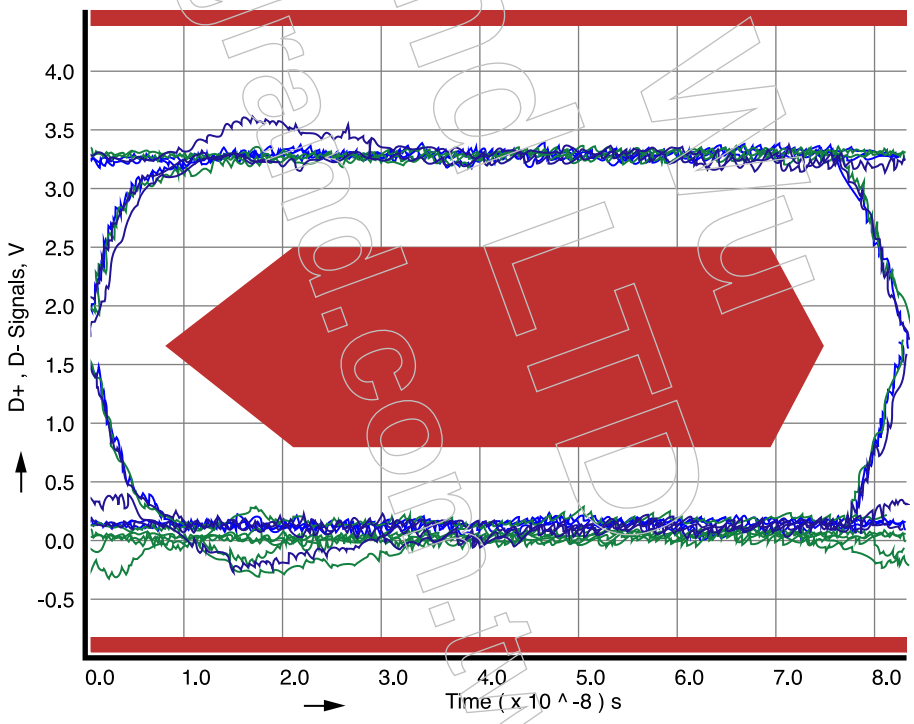
43.3.3 Full-Speed USB

The following are required for validating USB host Full-Speed (FS) signal integrity:

- FS Adjacent Trigger device (a FS thumb drive or HS thumb drive)
- Three single ended active probes
- TDSUSBF signal quality (SQiDD) board
- TDSUSB2 running on a compliance scope
- The USB host under test must enumerate the device and send SOF packets every 1 msec. See Start of Frame Packet (SOF) (page 520).

Testing must be done at TP2 at the point where host provides A-Type USB receptacle. The host must qualify the FS USB TDSUSB compliance mask. See Figure 43-4 (page 519) and Table 43-4 (page 520).

Figure 43-4 TP2 Eye Diagram Mask



Paul Quinn Payne Ltd. paul@quinnpayne.com

Table 43-4 Full-Speed Signal Quality Requirements

Req. #	Requirement	Value
SI_FS_01	Mask	0.8 V - 2.5 V
SI_FS_02	Rise Time	14 ns
SI_FS_03	Fall Time	5 ns
SI_FS_04	Source Jitter (next transition)	-3.5 to 3.5 ns
SI_FS_05	Signal Rate	12Mb/s +/- 0.25%

The FS eye diagram must be taken at TP2 while the FS transmitter is sending SOF packets to the device. See Start of Frame Packet (SOF) (page 520). The device must fully enumerate with the host for SOF packets to be present.

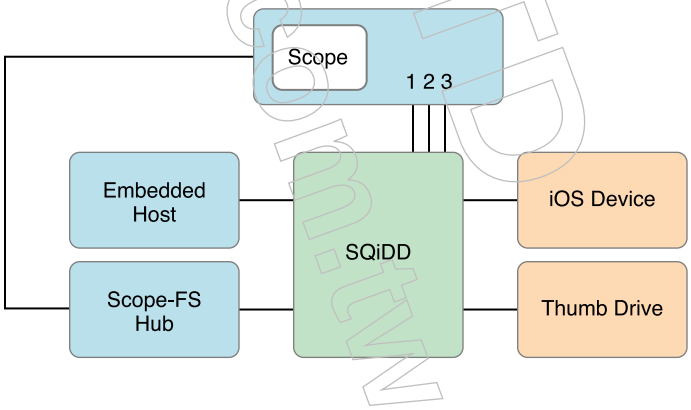
43.3.3.1 Start of Frame Packet (SOF)

Once the USB device is fully enumerated by the USB host, the host will continuously send SOF packets every 1 msec +/- 500ns. The SOF packet consists of an 11-bit frame number.

Sync	PID	Frame #	CRC5	EOP
	8 bits	11 bits	5 bits	

43.3.3.2 Test Environment

Figure 43-5 Full-Speed Test Environment



The USB host must be connected to the device and an FS adjacent trigger device must be connected to the scope USB bus. Channel connections are as follows:

Taiwu Wu Unilan Technology Ltd. Taiwu@unilan.com.tw

- Channel 1: Apple device USBDP
- Channel 2: Apple device USBDM
- Channel 3: FS adjacent trigger device USBDP

The TDSUSB2 application running on compliance scope is configured for above channel settings and full speed testing.

43.3.3.3 Test Procedure

Perform the following steps:

1. Connect accessory to SQ board, switch in "init" Position.
2. Connect Scope Ch1 and Ch2 with single ended probes to USBDP and USBDM with proper grounding.
3. Ensure the device gets enumerated and observe SOF packets every 1 msec on scope.
4. Connect a thumb drive via FS hub to scope.
5. Connect Ch3 to USBDP of adjacent trigger (use inrush current section of SQiDD board).
6. Run the TDSUSB application with FS settings.
7. Save the TDSUSB report including eye diagram.

43.3.3.4 Naming Report

Full-Speed accessory file name format:

```
HostName_Model#_FirmwareVersion_FSSQ_Date_PASS/FAIL_MFi_Report.html
```

43.3.3.5 Pass/Fail Criteria

The report generated by TDSUSB will indicate failure if there is a mask, timing or jitter failure. Submit the report as is without any modifications.

taywu@universal-lab.com.tw
Universal Lab LTD

44. USB Device Mode

When this transport is active, the accessory is a USB host and the Apple device is a USB device. USB Device Mode is not recommended for most accessories as the only features it supports are iAP2, Digital Audio (2 channel output only), and Power (to device only).

44.1 USB Embedded Host Implementation

All accessories that connect to an Apple device in USB Device Mode must comply with the requirements set forth for USB Embedded Hosts in the USB 2.0 specifications.

If the accessory integrates a USB Type A receptacle that in turn connects to the Apple device via a detachable cable, a valid Test ID issued by a USB-IF authorized lab (see http://www.usb.org/developers/compliance/labs/) must be submitted to Apple as part of the self-certification audit process.

44.2 Power

All accessories that connect to an Apple device in USB Device Mode must provide power to the device as specified in Power (page 480).

44.3 Enumeration

The initialization and configuration of an attached USB device is specified in the USB 2.0 specification. This specification does not cover this topic in detail, but instead provides information specific to Apple devices. To distinguish an Apple device in USB Device Mode, the accessory must check the device descriptor of attached USB devices for the following fields:

- Vendor ID = 0x05AC
- Product ID = 0x12*nn*

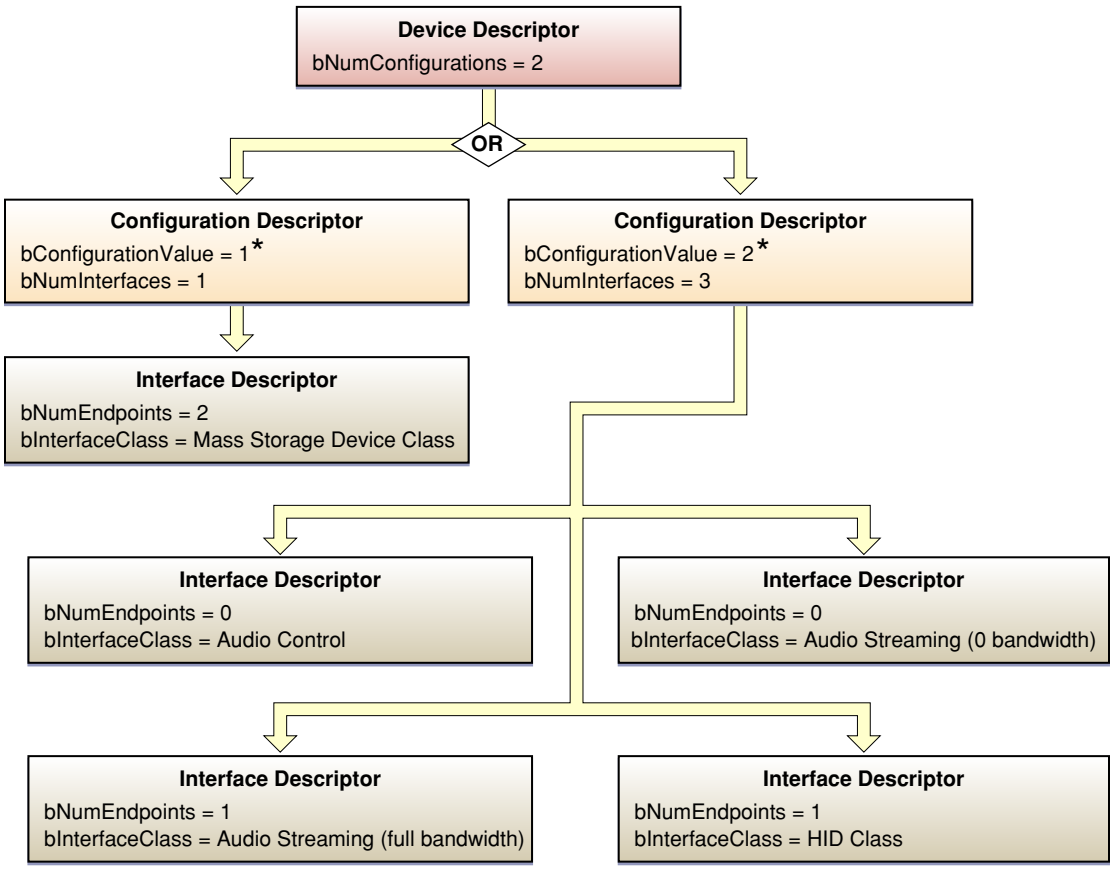
2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Note: Accessories must not attempt to distinguish between different types of Apple devices based on the least significant byte of the Product ID. Additionally, accessories must not use any other aspect of the device descriptor, such as number of configurations, interfaces, of endpoints, to determine whether an Apple device is connected. Use of additional parameters will cause the accessory to fail self certification immediately.

44.4 iAP2 Configuration

The iUI configuration enables iAP2 over the USB Device Mode transport and USB Device Mode Audio. For iAP2, a Human Interface Device (HID) interface is exposed to the accessory and uses two endpoints for communication: the control endpoint (endpoint number 0) is used for OUT data, while the HID interrupt endpoint is used for IN data.

Figure 44-1 USB Device Mode Interface Descriptor



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Note: The Apple device USB isochronous audio data endpoint descriptor bmAttributes field erroneously returns the Synchronization Type field (D3:2) as b00 (no synchronization) instead of the correct value, b11 (synchronous). Apple devices support synchronous data transfers, so accessories must override these attribute bits. The erroneous b00 value is retained for backwards compatibility with older Apple device accessories.

The Apple device HID interface utilizes several vendor-specific HID reports, some of which are used to transport data from the accessory (output reports) and some of which are used to transport data to the accessory (input reports). To send data to an Apple device, the accessory must choose one or more appropriately sized HID reports in which to embed the iAP2 link packet and send them to the Apple device HID interface using USB SetReport commands. The Apple device reassembles the iAP2 link packet and processes it. The process is repeated in reverse when the Apple device sends responses or iAP2 link packets to the accessory. In this case, the data is sent on an interrupt endpoint associated with the HID interface.

The different HID report sizes, endpoint requirements, and particulars are all described in the USB descriptors that accompany the interface.

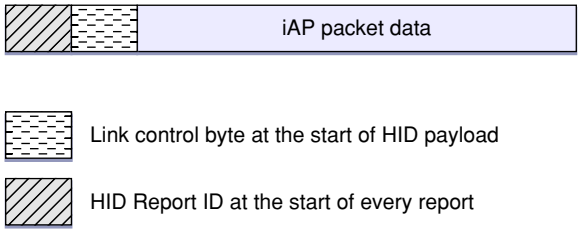
Note: Accessories must always request and parse the HID report descriptor each time an Apple device is connected or the accessory resets USB, because the HID Report ID and size descriptions may change.

44.5 HID Interface

As mentioned earlier, the HID interface breaks iAP packets up into a stream of vendor-specific HID reports and transports them across USB in either direction. To help manage this, it breaks this stream up into logical sets of reports, where a set of reports encompasses one or more complete iAP packets. For instance, a set could be a single HID report containing one iAP packet or a set of seven HID reports containing a total of three iAP packets. A vendor-specific HID report, as defined by the USB specification, consists of a Report ID followed by a payload of data that is specific to the vendor and its usage. The payload of this HID report is a link control byte (LCB), followed by iAP packet data.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 44-2 USB Device Mode Interface HID Report



The HID Report ID indicates the type of report and implies the size of the report. Every report of a given type is the same size. The Apple device specifies several different report types. The USB host must analyze the HID report descriptor of the Apple device at runtime to determine which Report ID corresponds to the most appropriate report type for each transfer. Note that the HID report descriptor may change in future Apple devices.

Usage of a HID Report ID is defined by the USB specification and is not specific to Apple devices, in contrast to the LCB and the rest of the payload.

The link control byte provides a mechanism for grouping sets of reports and is used by the HID interface to manage the data flow.

Table 44-1 Link control byte usage

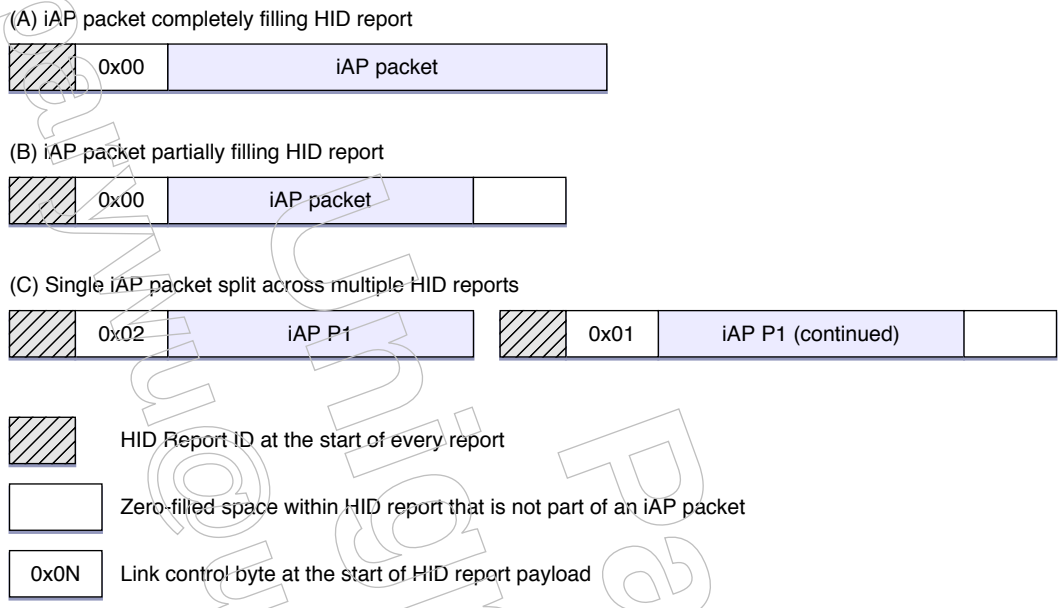
Bit	Name	Usage
Bit 0	Continuation	0 indicates that this HID report is the first in a set of one or more reports. This also implies that any previous sets are completed. Any incomplete iAP packets received prior to the arrival of this report are flushed and lost. 1 indicates that this report is not the start of a set, but is a continuing part of a set.
Bit 1	More to Follow	0 indicates that this report is the last in a set. Any following reports must be part of another set. 1 indicates that the current report set is not yet complete and there is at least one more report expected.
Bits 2-7	Reserved	Set to 0.

In general, iAP packets can be packed into HID reports in any manner, given the following limitations:

- All unused space within any HID report must be set to 0x00.
- If there is more than one iAP packet in the same HID report, there must be no unused space between them.
- If an iAP packet is split across multiple HID reports, all component reports must be in the same logical set of reports.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 44-3 USB Device Mode Interface Report Packing



Taiwan Unigrand LTD Party Wu taiwanunigrand.com.tw

45. USB Host Mode

USB Host Mode is recommended for all new accessory designs other than simple chargers and charge/sync cables. When this transport is active, the Apple device is a USB host.

Apple devices are capable of functioning as either a USB 2.0 High Speed or Full Speed host. Unless otherwise overridden in this specification, the accessory must comply with the USB 2.0 specification, plus any applicable device class-specific USB specifications that are available at http://www.usb.org/developers/docs/usb20_docs/.

The following additional USB implementation requirements apply:

- All USB descriptors (particularly the endpoint descriptors and the bMaxPower field of the configuration descriptors) must accurately represent the accessory's capabilities.
- Every USB device descriptor must declare a unique Vendor ID (VID) assigned by the USB-IF and a unique Product ID (PID) assigned by the accessory developer. The USB-IF Vendor ID must be assigned to the MFi licensee whose brand name appears on the accessory or its packaging. Re-using the Vendor ID of a silicon/component/reference design provider or contract manufacturer is specifically prohibited.
- The USB device descriptor must set the `Device Class`, `Device Subclass`, and `Device Protocol` fields to 0x00 to indicate that each interface descriptor will specify its own class code.
- All USB Device, Configuration, and Interface descriptors must be accompanied by human-readable String descriptors. Among these String descriptors, the following must match IdentificationInformation (page 579) message parameter values that the accessory passes to the Apple device:
  - USB Manufacturer String Descriptor must match the `Manufacturer` parameter value.
  - USB Product String Descriptor must match the `Name` parameter value.
  - USB Serial Number String Descriptor must match the `SerialNumber` parameter value.
- There is no USB 5 V VBUS supply from an Apple device; only Accessory Power is available.
- Upon detecting USB Suspend (the USB bus has been idle for 3.0 ms) the accessory must immediately enter Low Power Mode and remain there until it receives any non-idle signaling.
- The accessory must be capable of handling simultaneous bulk IN and bulk OUT transfers.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

45.1 iAP2 Interface Descriptor

Accessories that establish an iAP2 connection to an Apple device in USB Host Mode must declare a vendor-specific interface with one bulk IN endpoint and one bulk OUT endpoint. The accessory can determine that the Apple device has successfully entered USB Host Mode by detecting that it has been fully enumerated by the Apple device.

Table 45-1 USB Host Mode iAP2 interface descriptor

USB Descriptor	Value	Comments
Interface Number	0x00	
Interface Class	0xFF	Vendor-specific interface
Interface Subclass	0xF0	MFi accessory
Interface Protocol	0x00	
Interface String	'iAP Interface'	
Number of Endpoints	2	1 bulk IN and 1 bulk OUT endpoint descriptor must be specified

45.2 iAP2 Data Transfers

In USB Host Mode, the Apple device sends iAP2 data to the accessory using the bulk OUT endpoint. The accessory must return a USB ACK packet if the write operation is successful or return a USB NAK packet if it is not ready to accept data. If the accessory repeatedly returns a USB NAK packet for more than 1 second, the write operation will time out and information from the Apple device may be lost.

45.3 iAP2 Performance Optimization

Accessories should send the largest iAP2 link packets possible. This permits more data to be sent to the Apple device before a response is required. Sending small packets and waiting for the Apple device to respond introduces unnecessary delays and lowers data throughput.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

46. USB Role Switch

Some Apple devices can enter USB Host Mode from USB Device Mode after receiving a specific USB Vendor Request. Accessories that are normally USB hosts when not connected to an Apple device may choose to implement USB Role Switch and take advantage of features that are only available to USB Host Mode accessories while using a standard USB to Lightning cable.

46.1 USB Role Switch Requirements

The following requirements apply to accessories that support the USB role switch feature:

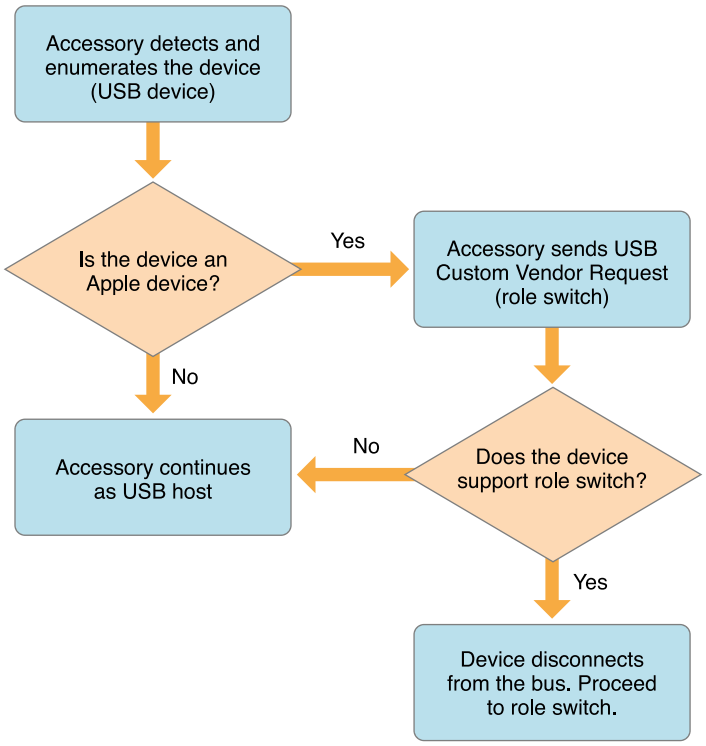
- The accessory must have a USB-A receptacle that is capable of functioning in both USB Host and USB Device roles.
- The receptacle must be labeled in accordance with the requirements specified in the Power feature. See Multiple Connectors (page 486) for details.
- The USB-A receptacle must provide power compatible with Apple devices as specified in the Power feature.
- The accessory must pass the USB test procedures (see Test Procedures (page 514)) in both USB Host Mode and USB Device Mode. Accessories should send test packets for 5 minutes in USB Device Mode, Role Switch, and send test packets for 5 minutes in USB Host Mode.

46.2 USB Role Switch Usage

The process of initiating a USB role switch using a custom vendor request is outlined in Figure 46-1 (page 530).

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 46-1 USB Role Switch Custom Vendor Request (Apple device is USB Device, Accessory is USB Host)



The sequence of events that takes place when an accessory is connected to an Apple device is as follows. For Steps 1-4, the accessory is in USB Host Mode and the Apple device is in USB Device Mode. For Steps 5-6, these roles are reversed.

1. The accessory (USB host) must detect and enumerate the Apple device (USB device).
2. The accessory must send the following USB Custom Vendor Request, which asks for a USB role switch:

Table 46-1 USB Vendor Request for Apple Device to Host Mode Switch

Field	Value	Comments
`bmRequestType`	`0x40`	Host-to-device request, vendor-defined type, device is recipient
`bRequest`	`0x51`	Vendor-defined USB role switch request
`wValue`	`0xnn`	See Table 46-2 (page 531)
`wIndex`	`0x00`	Reserved
`wLength`	`0`	No data transfer

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 46-2 USB Vendor Request for Apple Device to Host Mode Switch: Valid wValues

Value	Description
`0x00`	Default value
`0x01`	Accessory supports Apple CarPlay (see CarPlay (page 244))
All other values	Reserved

3. If the Apple device supports this role switch request, it disconnects itself from the bus. If it does not, it issues a STALL packet. See *On-The-Go Supplement, Revision 2.0, version 1.1a*, Section 5.2.1, Step B. If the accessory receives a STALL packet, it must assume that the Apple device does not support this feature and proceed accordingly. For example, the accessory may establish an iAP2 connection using USB Device Mode instead.
4. The accessory must detect the bus disconnect and turn on D+ pull-up. See *On-The-Go Supplement, Revision 2.0, version 1.1a*, Section 5.2.1, Step C.
5. The Apple device will assert a bus reset signal to start using the bus. See *On-The-Go Supplement, Revision 2.0, version 1.1a*, Section 5.2.1, Step D.
6. The accessory (USB device) must wait at least 1000 ms for the Apple device (USB host) to start enumeration of the accessory. See *On-The-Go Supplement, Revision 2.0, version 1.1a*, Section 7.1.5.
7. If 1000 ms passes without any traffic, the accessory must transition back to USB Host Mode.

Note: In previous versions of this specification, a USB Vendor Request with the `bRequest` field set to `0x50` was defined. That Vendor Request is now deprecated and must not be used in new accessory designs.

The following two events take place when the cable connecting the Apple device to the accessory's USB-A receptacle is unplugged from either or both ends:

- The Apple device detects loss of VBUS and switches back to USB Device Mode.
- The accessory detects a minimum of 200 ms of bus inactivity and transitions back to USB Host Mode. See *On-The-Go Supplement, Revision 2.0, version 1.1a*, Section 7.1.6.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

47. Vehicle Status

Accessories may use the Vehicle Status feature to send sensor data and other useful information from a vehicle to an Apple device.

The following types of information can be communicated:

- Engine type
- Estimated range remaining
- Outside temperature
- Low range indicator warning status

47.1 Vehicle Status Requirements

Accessories that implement the Vehicle Status feature must provide live information from a vehicle. Simulation of vehicle status information from other sources is not allowed.

All accessories that support the VehicleStatus feature via iAP2 must send or receive the following iAP2 control session message(s):

StartVehicleStatusUpdates (page 625)

VehicleStatusUpdate (page 625)

StopVehicleStatusUpdates (page 626)

47.2 Vehicle Status Usage

Accessories that implement this feature must identify one Table 52-18 (page 584) and one Table 52-20 (page 585).

If the Apple device can make use of some/all of the information, it will send a StartVehicleStatusUpdates (page 625) message to the accessory informing the accessory which information types send updates for in a VehicleStatusUpdate (page 625) message.

The accessory must only send vehicle status updates to the Apple device:

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- Between StartVehicleStatusUpdates (page 625) and StopVehicleStatusUpdates (page 626) messages.
- Once after receiving the StartVehicleStatusUpdates (i.e. providing an initial value).
- When the value has changed. The range and temperature values should change by at least 0.5 miles or 0.5° C respectively before an update is sent.
- At a maximum rate of 1 Hz.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

48. VoiceOver

VoiceOver is the screen reader feature built into both Apple's iOS and OS X operating systems. Users who are blind or have low vision can interact with and control Apple devices and Macs simply by moving their finger over the touchscreen or touch-sensitive trackpad. For more information on VoiceOver, see http://www.apple.com/accessibility/voiceover/. In addition, when VoiceOver is paired with certain accessories users are able to use VoiceOver even if they are unable to touch the display at all.

Most VoiceOver accessories interface with one or more switch-type inputs. Accessory developers should consider supporting VoiceOver if the accessory is intended for users who cannot see the display or touch the screen. If the user can see the display, consider supporting the AssistiveTouch or Assistive Switch Control features.

48.1 VoiceOver Requirements

All accessories supporting the VoiceOver feature must be targeted at users with special needs.

All accessories that support the VoiceOver feature via iAP2 must send or receive the following iAP2 control session message(s):

- StartVoiceOver (page 626)
- RequestVoiceOverMoveCursor (page 627)
- RequestVoiceOverActivateCursor (page 627)
- StartVoiceOverUpdates (page 629)
- VoiceOverUpdate (page 630)

All accessories that support the VoiceOver feature via iAP2 may also send or receive the following iAP2 control session message(s):

- StopVoiceOver (page 627)
- RequestVoiceOverScrollPage (page 628)
- RequestVoiceOverSpeakText (page 628)
- RequestVoiceOverPauseText (page 629)
- RequestVoiceOverResumeText (page 629)

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

RequestVoiceOverConfiguration (page 631)

StopVoiceOverUpdates (page 630)

StartVoiceOverCursorUpdates (page 631)

VoiceOverCursorUpdate (page 631)

StopVoiceOverCursorUpdates (page 633)

48.2 VoiceOver Usage

Accessories supporting VoiceOver must identify themselves as using one or more VoiceOver messages and have that configuration accepted by the Apple device. Apple devices that do not support VoiceOver accessories will reject accessory identifications that include VoiceOver messages.

The VoiceOver feature is started and stopped using the StartVoiceOver (page 626) and StopVoiceOver (page 627) messages. All compatible accessories must send the StartVoiceOver (page 626) message. Also, all VoiceOver accessories must receive and process VoiceOverUpdate (page 630) messages to be notified when the feature is enabled or disabled asynchronously. The StartVoiceOverUpdates (page 629) message will signal the device to start generating and sending VoiceOverUpdate (page 630) messages to the accessory.

It is possible (and not uncommon) for more than one VoiceOver-compatible accessory to be connected to a device at the same time. All of those accessories, or their users, may independently start or stop the VoiceOver feature at any time, and a VoiceOver accessory must handle these use cases correctly.

Several additional VoiceOver messages deal with manipulation and selection of user interface elements. Basic VoiceOver accessories will typically only send the RequestVoiceOverMoveCursor (page 627) and RequestVoiceOverActivateCursor (page 627) messages.

Some accessories may choose to also receive VoiceOverCursorUpdate (page 631) messages; this message provides additional information to the accessory about the currently selected user interface element if the app provides it.

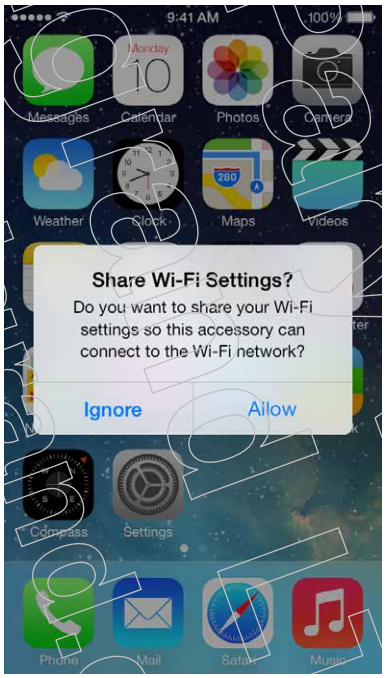
The accessory may send RequestVoiceOverConfiguration (page 631) to adjust the volume and/or speaking rate after the feature is started. Additionally, accessories can request that the speech synthesizer be used to render certain useful phrases specific to the accessory at key moments. The speech synthesizer volume is persistent and will be saved and restored by the Apple device when VoiceOver is stopped and started.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

49. Wi-Fi Information Sharing

Apple devices can share Wi-Fi configuration information with an accessory via an iAP2 connection. The accessory can initiate this process, but the user must grant permission for the device to share this information. The device can only share information about the currently connected Wi-Fi network, and this feature will not account for other router-configured access control mechanisms such as RADIUS or MAC address filtering.

Figure 49-1 Wi-Fi Information Sharing Alert



49.1 Wi-Fi Information Sharing Requirements

All accessories that support the Wi-Fi Information Sharing feature via iAP2 must send or receive the following iAP2 control session message(s):

RequestWiFiInformation (page 633)

WiFiInformation (page 633)

An accessory designed to use this feature must comply with the following requirements:

- It must let the user initiate Wi-Fi login sharing via either a physical button or an onscreen option.
- It must not be able to initiate Wi-Fi login sharing without an explicit user action via the means described above.
- It must notify the user, visibly and/or audibly, when it has received Wi-Fi connection information.
- It must notify the user, visibly and/or audibly, when it has successfully established a Wi-Fi connection.

49.2 Wi-Fi Information Sharing Usage

To request the Wi-Fi network information, the accessory must send the RequestWiFiInformation (page 633) message. A request must only be sent in response to direct user action such as pressing a button or selecting a menu item.

The device will send WiFiInformation (page 633) when the user has responded to the request. The Wi-Fi network SSID and passphrase are part of the WiFiInformation (page 633), and the network security information must be determined by examining the Access Point's beacon frames per the 802.11 specification.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

50. iAP2 Link

Every iAP2 connection starts with the establishment of a link between an accessory and a device over a supported transport. The link protocol provides a transport-agnostic mechanism for reliable and ordered delivery of packetized data belonging to one or more iAP2 sessions. The protocol is also configurable on a per-connection basis and can be optimized for best performance on any particular transport and accessory usage profile. Certain protocol features contribute towards these goals:

- Positive acknowledgement of received packets.
- Retransmissions only require the unacknowledged packets in a sequence to be resent.
- Explicit and efficient support for iAP2 sessions.

iAP2 is supported over the following transports:

- Bluetooth (see Bluetooth (page 216))
- Serial (see Serial (page 494))
- USB Device Mode (see USB Device Mode (page 522))
- USB Host Mode (see USB Host Mode (page 527))

All devices support one or more transports that can be used to establish an iAP2 connection with an accessory. Some transports are better suited for various features than others, so transport selection should be a high priority early in the accessory design process.

Sample source code for an iAP2 link layer implementation may be found on the MFi Portal.

50.1 Packet Structure

Every link packet must start with a fixed-size 9-byte header, including checksum, and is followed by an optional variable-length data payload. Both the header and payload have their own checksums. If there is no payload data, there is no payload checksum. An example can be found in iAP2 Link Packet Structure Example (page 559).

Table 50-1 iAP2 Link Packet Structure

Start of Packet MSB (`0xFF`)

Start of Packet LSB (0x5A)
Packet Length MSB
Packet Length LSB
Control Byte
Packet Sequence Number
Packet Acknowledgement Number
Session Identifier
Header Checksum
...
Payload Data
...
Payload Checksum

50.1.1 Start of Packet

The first two bytes of every iAP2 link packet are always 0xFF 0x5A. If these bytes are detected in the transport stream then both accessory and device must attempt to parse the following bytes as a valid packet.

50.1.2 Packet Length

The next two bytes denote the Packet Length in bytes and are always expressed as an unsigned 16-bit big-endian integer. A link packet with no Payload Data always has a Packet Length of 9 bytes (from Start of Packet to Header Checksum). Otherwise, the Packet Length is measured from the Start of Packet to the last byte of Payload Data including the Payload Checksum. For example, a link packet with 1 byte of Payload Data has a Packet Length of 11 bytes. Packets have a maximum Payload Data size of 65525 bytes; a packet with such a payload has a Packet Length of 65535 bytes.

50.1.3 Control Byte

The bits in the Control Byte indicate what is present in the packet.

- The SYN, EAK, and RST bits are mutually exclusive.
- The ACK bit may be combined with the SYN bit.

Pary Wu
wu@unigrandtech.com.tw

- The EAK bit is always set along with the ACK bit.
- The RST bit is never set in conjunction with any other bit.
- The SLP bit is never set in conjunction with any other bit.
- All unassigned bits must be set to 0.

iAP2 Session Payloads cannot be present when any of the SYN, EAK, or RST bits are set. Conversely, link packets with iAP2 session payloads always have the ACK bit set.

Table 50-2 iAP2 Link Control Byte Bits

Bit	Name	Meaning
7	SYN	Link Synchronization Payload is present
6	ACK	Packet Acknowledgement Number is valid, and iAP2 Session Payload may be present
5	EAK	Extended Acknowledgement Payload is present
4	RST	Link Reset
3	SLP	Device Sleep

In the rest of this specification, a link packet can be described in terms of its Control Byte bits and payload. For example, a SYN packet refers to a link packet with the SYN bit set in the Control Byte, and a SYN+ACK packet refers to a link packet with both the SYN and ACK bits set in the Control Byte.

50.1.4 Packet Sequence Number

Every link packet contains a Packet Sequence Number that uniquely identifies the packet among all other packets in transit. When a link is first created, both the device and accessory must randomly pick an initial sequence number before sending their first SYN/SYN+ACK packet. Each time a packet with iAP2 Session or Link Synchronization Payload Data is sent, the sequence number is incremented by 1. Otherwise, the sequence number does not increment when a packet is sent. Retransmissions of a previously sent packet must retain the same Packet Sequence Number. The sequence number wraps back to 0 upon reaching 255.

50.1.5 Packet Acknowledgement Number

The Packet Acknowledgment Number only has meaning if the ACK bit in the Control Byte is set. If ACK is not set, the Packet Acknowledgement Number must be set to 0 by the sender, and it must be ignored by the receiver.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

If ACK is set, the Packet Acknowledgment Number indicates to a sender the Packet Sequence Number of the last in-sequence packet that was received. For example, if the Packet Sequence Numbers 1, 2, 3, and 5 were received by the accessory, the accessory's next outgoing packet's Packet Acknowledgement Number would be 3, because 5 was received out of sequence.

50.1.6 Session Identifier

The Session Identifier only has meaning if the ACK bit in the Control Byte is set and there is an iAP2 session payload present. If those two conditions are met, the Session Identifier will be a nonzero number that specifies a particular session in an iAP2 connection. Otherwise, the Session Identifier must be set to 0.

50.1.7 Header Checksum

The Header Checksum is calculated by adding together all of the following packet bytes as signed 8-bit values, discarding any signed 8-bit overflow, and negating the sum to produce a signed 8-bit value. If this signed 8-bit value does not match the Header Checksum, the receiver must restart parsing of packets from the next detected Start of Packet sequence.

- Start of Packet MSB
- Start of Packet LSB
- Packet Length MSB
- Packet Length LSB
- Control Byte
- Packet Sequence Number
- Packet Acknowledgement Number
- Session Identifier

50.1.8 Payload Data

This section is optional, and its presence must match the state of the bits in the Control Byte. The maximum possible payload size is 65,525 bytes.

50.1.9 Payload Checksum

The Payload Checksum byte is present if and only if Payload Data is present. It is calculated by adding together all of the values of the Payload Data bytes as signed 8-bit values, discarding any signed 8-bit overflow, and then negating that sum. The resulting checksum is a signed 8-bit value. If this signed 8-bit value does not match the Payload Checksum, the receiver must restart packet parsing from the next detected Start of Packet sequence.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

50.2 Link Synchronization Payload

The Link Synchronization Payload (LSP) is used to establish a link and synchronize Packet Sequence Numbers between the device and accessory. It also contains the negotiable link parameters. An example can be found in iAP2 Link Synchronization Payload Example (page 560).

Table 50-3 Link Synchronization Payload (Version 1)

Link Version (`0x01`)
Maximum Number of Outstanding Packets
Maximum Packet Length MSB
Maximum Packet Length LSB
Retransmission Timeout MSB
Retransmission Timeout LSB
Cumulative Acknowledgement Timeout MSB
Cumulative Acknowledgement Timeout LSB
Maximum Number of Retransmissions
Maximum Cumulative Acknowledgements
iAP2 Session 1: Session Identifier
iAP2 Session 1: Session Type
iAP2 Session 1: Session Version
...
iAP2 Session N: Session Identifier
iAP2 Session N: Session Type
iAP2 Session N: Session Version

50.2.1 Link Version

- The version of the link being established. All packet payloads may vary depending on the Link Version.
- The only valid value of Link Version at this time is `1`.

Paryl Wu LTD
paryl@uniqland.com.tw

- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

50.2.2 Maximum Number of Outstanding Packets

- The maximum number of packets that may be sent without receiving an acknowledgement.
- Valid values are 1 to 127.
- This is not a negotiable parameter.
- The accessory and device may propose and use different values.
- The accessory must not send more than the device's proposed Maximum Number of Outstanding Packets without waiting for an acknowledgement from the device, and vice versa.

50.2.3 Maximum Packet Length

- The largest possible Packet Length in bytes.
- Valid values are 24 to 65535.
- This is not a negotiable parameter.
- The accessory and device may propose and use different values.

50.2.4 Retransmission Timeout

- The timeout value in milliseconds for retransmission of unacknowledged packets. This should be set to a value approximating the transmission time for a packet over the link transport.
- Valid values are 20 ms to 65535 ms.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

50.2.5 Cumulative Acknowledgement Timeout

- The timeout value in milliseconds after which an acknowledgment packet must be sent if another packet is not sent.
- Valid values are 10 ms to half of the Retransmission Timeout.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

Paiyu Electronics LTD
paiyu@eiechina.com

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

50.2.6 Maximum Number of Retransmissions

- The maximum number of packet retransmissions attempted before the link is considered to be broken.
- Valid values are `1` to `30`.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

50.2.7 Maximum Cumulative Acknowledgements

- The maximum number of received acknowledgments that may be accumulated before an acknowledgement packet must be sent if another packet is not sent.
- Valid values are `0` to `127` or the Maximum Number of Outstanding Packets, whichever is smaller.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

50.2.8 iAP2 Sessions

- The iAP2 sessions that the accessory will use to communicate with the device.
- Valid session types and versions are defined in Type (page 561) chapter. Session identifiers must be unique to each defined session. 0 is not a valid session identifier.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

50.3 iAP2 Session Payload

The iAP2 Session Payload is specified in iAP2 Sessions (page 561). The ACK bit in the Control Byte must be set whenever an iAP2 Session Payload is present.

50.4 Extended Acknowledgement Payload

The Extended Acknowledgement Payload is used to acknowledge packets that were received out of sequence. This payload has the following attributes:

- Both the EAK and ACK bits in the Control Byte must be set.

Party Wu LTD
Unauthorized use of this material is prohibited. Please contact Party Wu LTD at partywu@unlimited.com.tw

- The Packet Acknowledge Number contains the sequence number of the last packet that was received in sequence.
- The Payload Data section contains the sequence numbers of one or more packets that were received out of sequence. Note that these are not acknowledgements for those packets. An acknowledgement will be sent separately in a later packet with the appropriate number in the ACK field.

Table 50-4 EAK Packet Payload (Link v1)

1st Out of Sequence Acknowledgement Number
...
Nth Out of Sequence Acknowledgement Number

50.5 Reset

The RST bit in the Control Byte is used by the device to reset a connection. Such a link packet does not have a payload and may only be sent by the device.

50.6 Sleep

The SLP bit in the Control Byte is used by the device to signal that it is about to enter a sleep state. Such a link packet does not have a payload and may only be sent by the device.

The device will stop supplying Accessory Power once it enters the sleep state. If the device exits the sleep state, it will start supplying Accessory Power again.

If the accessory is capable of suspend and resume of the link layer, the accessory must wait 500 ms after Accessory Power returns before re-initializing the link. During this time, the accessory may receive an ACK from the device which signals a link layer resume and serves to synchronize the sequence number (last number sent from the device) and ACK number (the last number received by the device).

If the accessory is not capable of suspend and resume of the link layer, the accessory must wait 80 ms after Accessory Power returns before re-initializing the link.

Accessories using the Bluetooth transport will not receive a suspend packet when the Apple device enters a sleep state.

Paul Wu
Unigrand Ltd
paul@unigrand.com.tw

50.7 Operation

50.7.1 Record

All link implementations must store the following variables in a record that is unique to the link. These variables will be referred to in subsequent subsections describing iAP2 Link operation.

Table 50-5 iAP2 Link Operation Record Variables

Variable	Description
SentACKTimer	A timer that keeps track of the elapsed time (ms) since the last ACK packet was sent
NextSentPSN	The Packet Sequence Number of the next packet to be sent
OldestSentUnacknowledgedPSN	The Packet Sequence Number of the oldest unacknowledged packet
InitialSentPSN	The Packet Sequence Number used for the very first packet sent
LastReceivedInSequencePSN	The Packet Sequence Number of the last packet received correctly and in sequence
InitialReceivedPSN	The Packet Sequence Number of the very first packet received
ReceivedOutOfSequencePSNs[n]	An array of Packet Sequence Numbers that have been received and acknowledged out of sequence

50.7.2 Initialization

Once the transport connection is established, the accessory must confirm the presence of a device that supports iAP2 by sending the following byte sequence at 1 Hz (once a second) until a response is received from the Apple device:

```
FF 55 02 00 EE 10
```

If the device supports iAP2, the accessory will receive the same byte sequence in return. If the device is not compatible with this Specification but is compatible with the MFi Accessory Firmware Specification R46, the accessory will receive one of the following:

- An iAP1 General Lingo `iPodACK` packet with a command status of `0x04` (ERROR: Bad Parameter): `55 04 00 02 04 EE 08`

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

- An iAP1 General Lingo `RequestIdentify` packet: `55 02 00 00 FE`

In either case, the packet may or may not be preceded by an `0xFF` sync byte.

If the accessory is also backward compatible with the MFi Accessory Firmware Specification R46, it may proceed to initiate communication with the older device using iAP1. The StartIDPS command must be sent within 500 ms of receiving confirmation that the Apple device supports iAP1.

If the accessory receives this message but is not backward compatible, the accessory must send the following byte sequence to the device to indicate lack of backward compatibility:

```
FF 55 0E 00 13 FF FF FF FF FF FF FF FF FF FF FF FF EB
```

The accessory must treat any other response from the device as final; no retries or retransmissions are allowed.

50.7.3 Synchronization

One notable feature of the link is support for automatic negotiation of transmission parameters over any type of transport. This permits the link to scale according to the capabilities of both device and accessory. link initialization has 3 main goals:

- Determines mutually acceptable link configuration parameters for both device and accessory.
- Searches for errors in configuration parameters.
- Terminates the link if no agreement can be reached.

A link is initiated when an underlying transport connection and device compatibility have been established. When this occurs, the accessory first sends a `SYN` packet containing desired connection parameters in the Link Synchronization Payload, along with a randomly generated Packet Sequence Number. The device will respond with an `SYN+ACK` packet acknowledging receipt of the accessory's `SYN` packet, its own desired connection parameters, and its randomly generated Packet Sequence Number. If the two sets of connection parameters match, the accessory must send a final `ACK` packet of the most recent `SYN+ACK` from the device, and the connection is considered to be established. Otherwise, `SYN+ACK` packets will continue to be exchanged up to `10` times until connection parameters are agreed upon. If `10` exchanges occur without agreement on connection parameters then the device will stop responding to any further packets sent by the accessory.

Sometimes, a device will be unable to respond immediately to the receipt of a `SYN` packet from an accessory. In that situation, the accessory may resend the `SYN` packet with the same Payload Data and the same Packet Sequence Number. It may continue doing so every second until the device sends a `SYN+ACK` response acknowledging the previously sent Packet Sequence Number. Once this response has been received, the accessory must ignore any subsequent incoming `SYN+ACK` packets with the same Packet Sequence Number.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

The device will send a RST packet to reset the link if the accessory's final Link Synchronization Payload contains an invalid non-negotiable parameter.

During synchronization, the following default link configuration parameters are assumed until synchronization is complete:

Table 50-6 Default link parameters during synchronization

Parameter	Default Value
Maximum Number of Outstanding Packets	1
Maximum Packet Size	128 bytes
Retransmission Timeout	1000 ms
Cumulative Ack Timeout	10 ms
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	0

The following tables contain suggested link configuration parameters for some iAP2 transports. These values are just starting points for development and must not be used without verifying suitability for a particular accessory.

Table 50-7 Suggested link parameters for USB Host Mode transport (Full Speed)

Parameter	Suggested Value
Maximum Number of Outstanding Packets	5
Maximum Packet Size	4096 bytes
Retransmission Timeout	2000 ms
Cumulative Ack Timeout	22 ms
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	3

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 50-8 Suggested link parameters for USB Device Mode transport (Full Speed)

Parameter	Suggested Value
Maximum Number of Outstanding Packets	5
Maximum Packet Size	4096 bytes
Retransmission Timeout	2000 ms
Cumulative Ack Timeout	22 ms
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	3

Table 50-9 Suggested link parameters for Bluetooth transport

Parameter	Suggested Value
Maximum Number of Outstanding Packets	5
Maximum Packet Size	2048 bytes
Retransmission Timeout	1500 ms
Cumulative Ack Timeout	73 ms
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	3

Table 50-10 Suggested link parameters for 57.6 kbps serial transport

Parameter	Suggested Value
Maximum Number of Outstanding Packets	4
Maximum Packet Size	256 bytes
Retransmission Timeout	1000 ms
Cumulative Ack Timeout	133 ms
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	2

Purty Wui
purtyui@uniguardsolutions.com.tw

50.7.4 Acknowledgements

To guarantee delivery of packets, the link protocol uses both positive acknowledgement and retransmission of packets. Each packet with Payload Data and SYN packets are acknowledged when they are correctly received and accepted by the receiver. ACK packets are not acknowledged. Damaged packets are discarded and not acknowledged. Packets are retransmitted by the sender when they are not acknowledged by the receiver within the time specified by the Retransmission Timeout.

There are two types of packet acknowledgements. A cumulative acknowledgement is used to acknowledge receipt of all in-sequence packets up to a specified Packet Sequence Number. The extended acknowledgement allows the receiver to acknowledge packets out of sequence. The type of acknowledgement used is simply a function of the order in which packets arrive. Whenever possible, link implementations must use the cumulative acknowledgement and use the extended acknowledgement only when packets are received out of sequence.

Receivers must generate and send an extended acknowledgement as soon as one or more out-of-sequence packets are received.

50.7.5 Retransmissions

Packets can get lost in transmission because of loss or damage in the underlying transport. They can also be discarded by the receiver. The positive acknowledgement of packets requires the receiver to acknowledge a packet only when the packet has been correctly received and accepted.

To detect missing packets, the sender must use a retransmission timer for each outgoing packet. This timer is set to the negotiated Retransmission Timeout value. When an acknowledgement is received for a packet, the timer is cancelled for that packet. If the timer expires before an acknowledgement is received, that packet is retransmitted and the timer is restarted, up to the Maximum Number of Retransmissions negotiated during link synchronization.

50.7.6 Flow Control

The iAP2 link employs a simple flow control mechanism that is based on the number of unacknowledged packets sent and the Maximum Number of Outstanding Packets link configuration parameter. This parameter is specified by each side when the link is first created, and should be set based on the number and size of buffers that each side is willing to allocate to the link. Once it is set this parameter remains unchanged while the link is connected.

The link employs the concept of a sequence number window for acceptable Packet Sequence Numbers. The left edge of the window is the last in-sequence acknowledged Packet Sequence Number plus one. The right edge of the window is equal to the last in-sequence acknowledged Packet Sequence Number plus the Maximum

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Number of Outstanding Packets. The link sender sends packets until the receiver's Maximum Number of Outstanding Packets is reached; once this limit is reached, the sender may only send a new packet for each acknowledged packet.

When a received packet has a Packet Sequence Number that falls within the window, it is acknowledged. If the Packet Sequence Number is equal to the left edge (i.e., it is the next expected Packet Sequence Number), the packet is acknowledged with a cumulative acknowledgement (ACK), and the acceptance window's left and right edges are incremented by one. If the received packet's Packet Sequence Number is within the window but is out of sequence, it is acknowledged with an Extended Acknowledgement (EAK). The window is not adjusted, and the receipt of the out-of-sequence packet is recorded. Received link packets with Packet Sequence Numbers outside of the window must be discarded; they are a sign that the link is unstable and may need to be reset.

When a sender receives Extended Acknowledgements (EAK), the sender must not transmit beyond the acceptance window. This could occur if one packet is not acknowledged but all subsequent packets are received and acknowledged. This requirement will fix the left edge of the window at the sequence number of the unacknowledged packet. As additional packets are sent, the next Packet Sequence Number will approach and eventually overtake the right edge. At this point, no more packets may be sent until the unacknowledged packet is acknowledged.

50.7.7 Reset

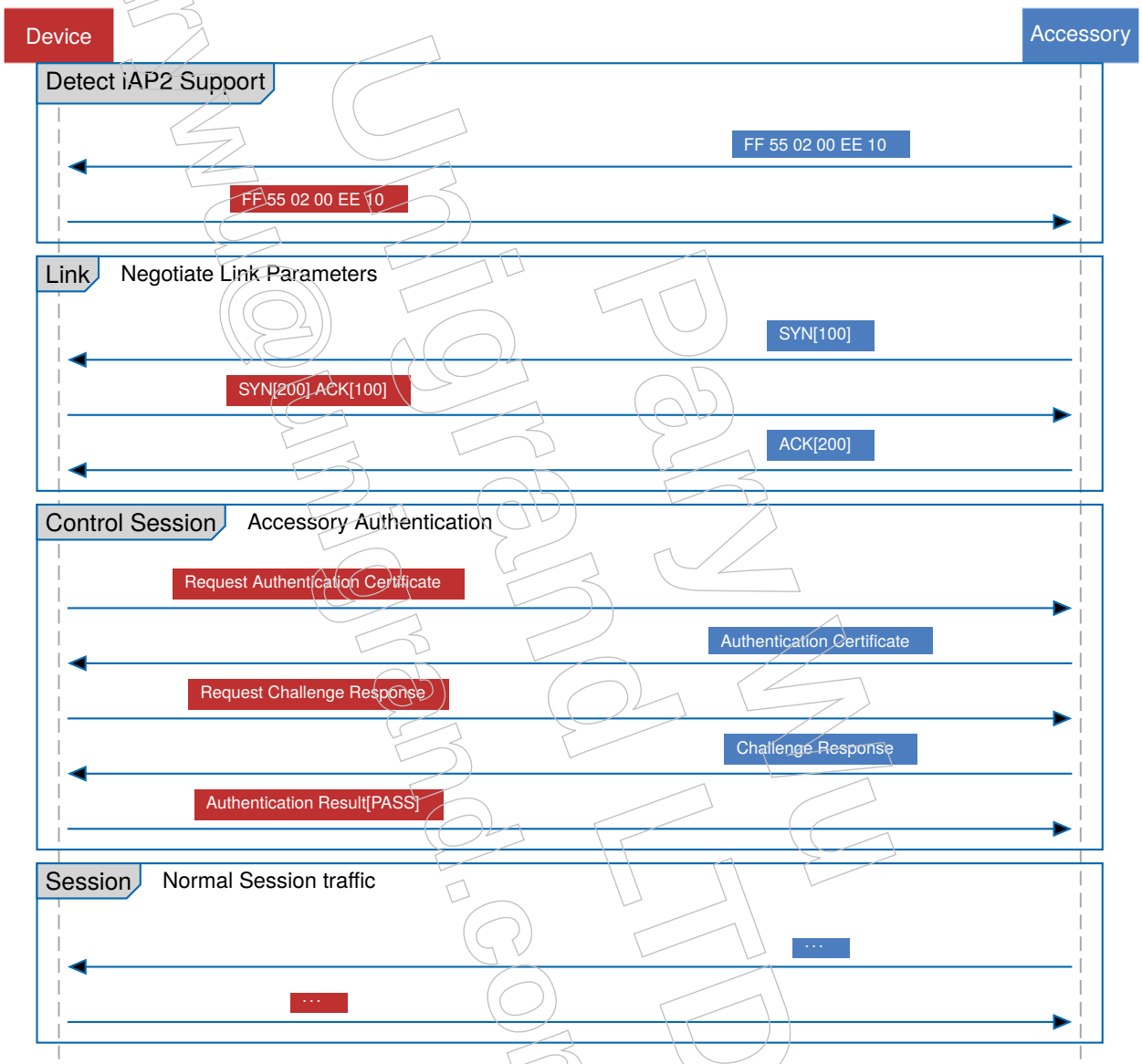
The device may send a RST packet to the accessory at any time. Upon receipt of this packet, the accessory must send a SYN packet to the device within 1 second of receiving the RST packet and proceed as specified in Synchronization (page 547).

Accessories must terminate the underlying transport connection and reconnect if they wish to reset the link. This is not possible when the serial transport is active without physically disconnecting and reconnecting the accessory with the device's Lightning connector. Having to restart the link is a sign of poorly negotiated link parameters and should not occur during typical operation.

Confidential Property of Wistron NeWeb Corp. (WNC) Unauthorized Copy Prohibited

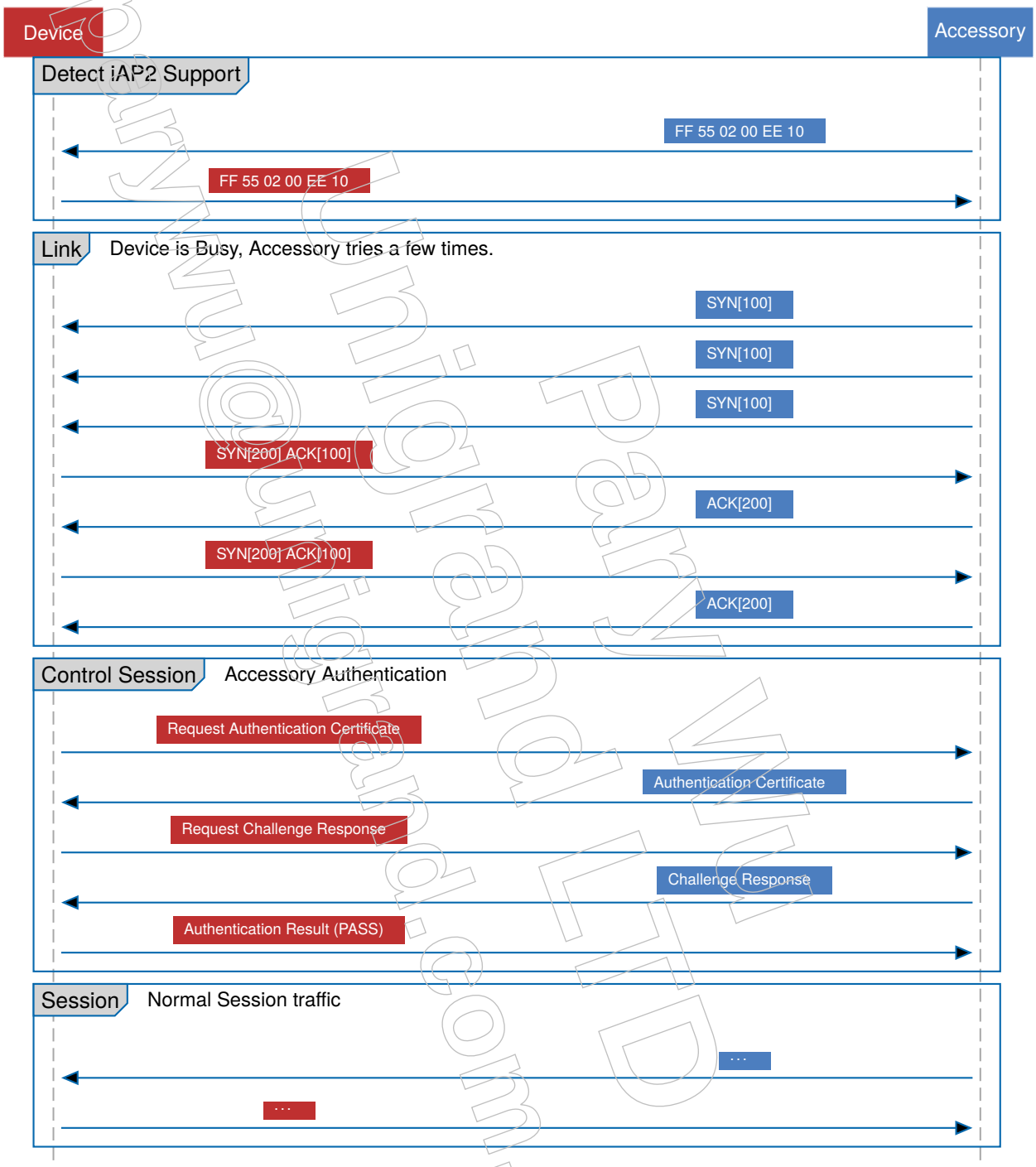
50.8 Examples

50.8.1 Typical Link Initialization



Uniquely Paired to iTunes Account
Paired with iTunes.com.tw

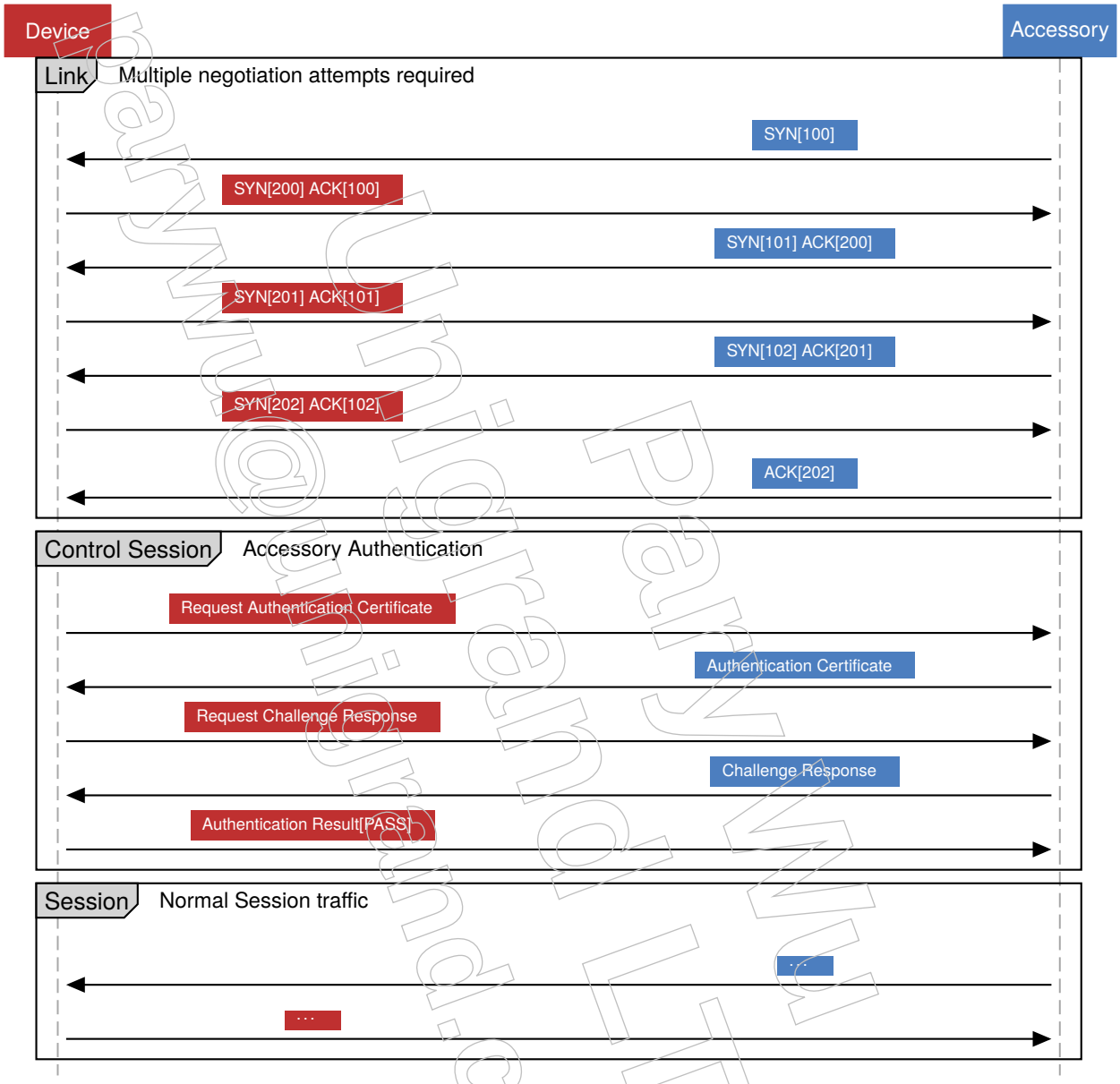
50.8.2 Connection Initialization When Device is Busy



50.8.3 Connection Requiring Multiple Negotiation Attempts

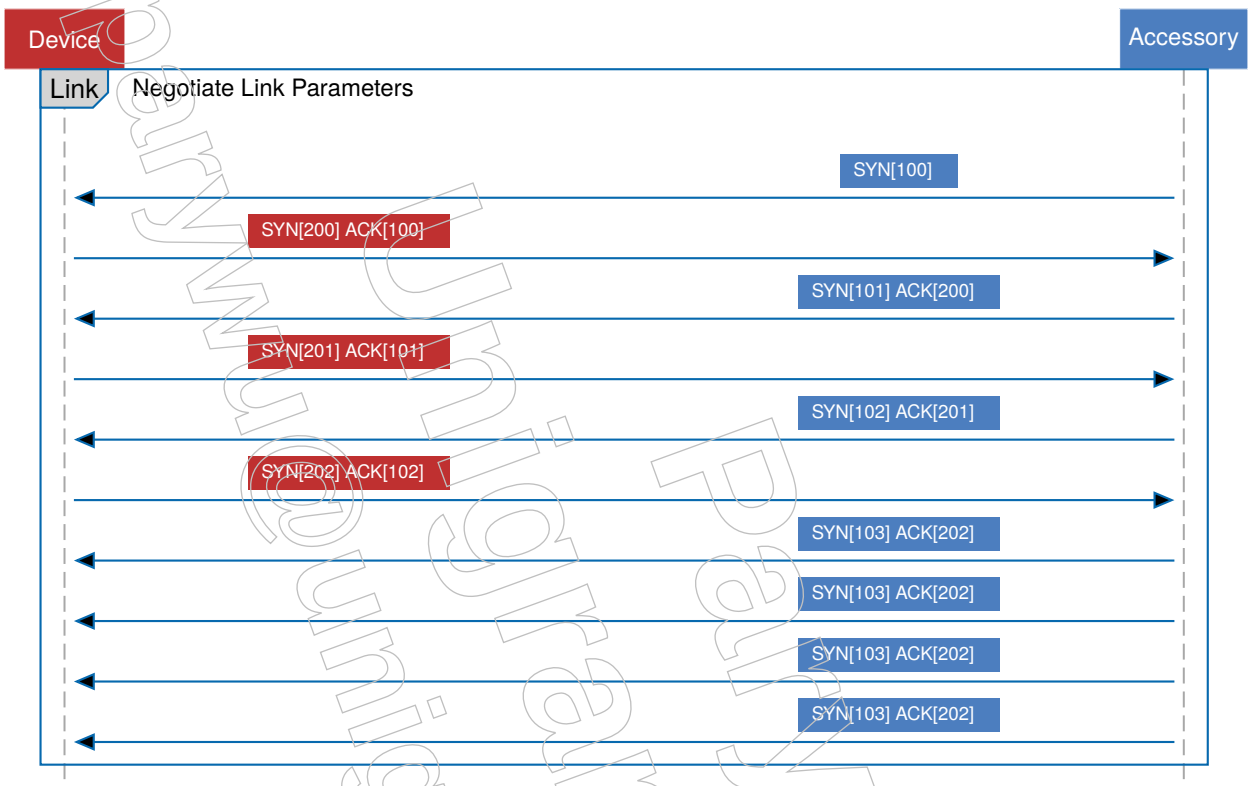
Multiple negotiation attempts may occur when accessory and device do not mutually agree on iAP2 Link parameters right away.

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

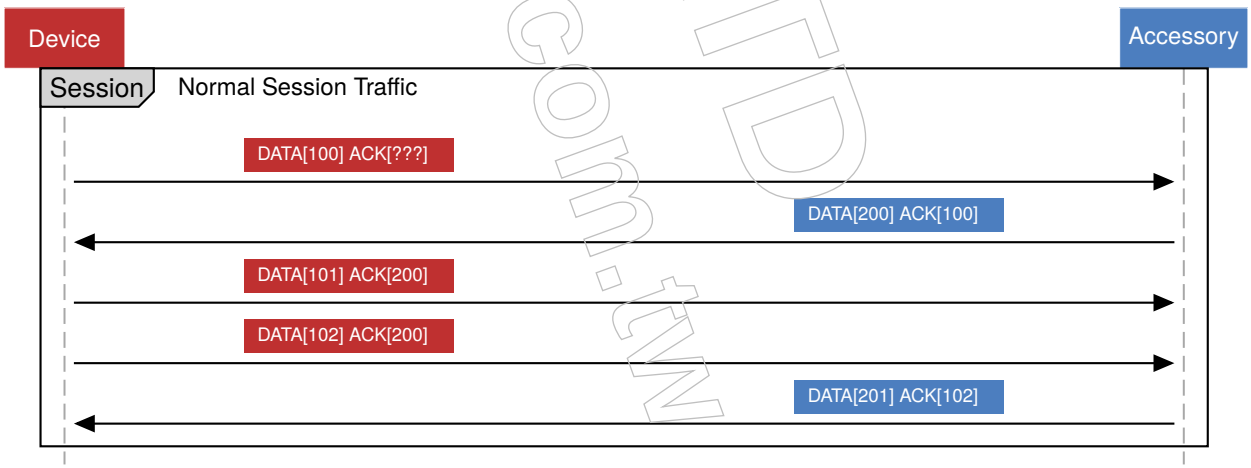
50.8.4 Connection With Failed Negotiation



50.8.5 Normal Connection Traffic

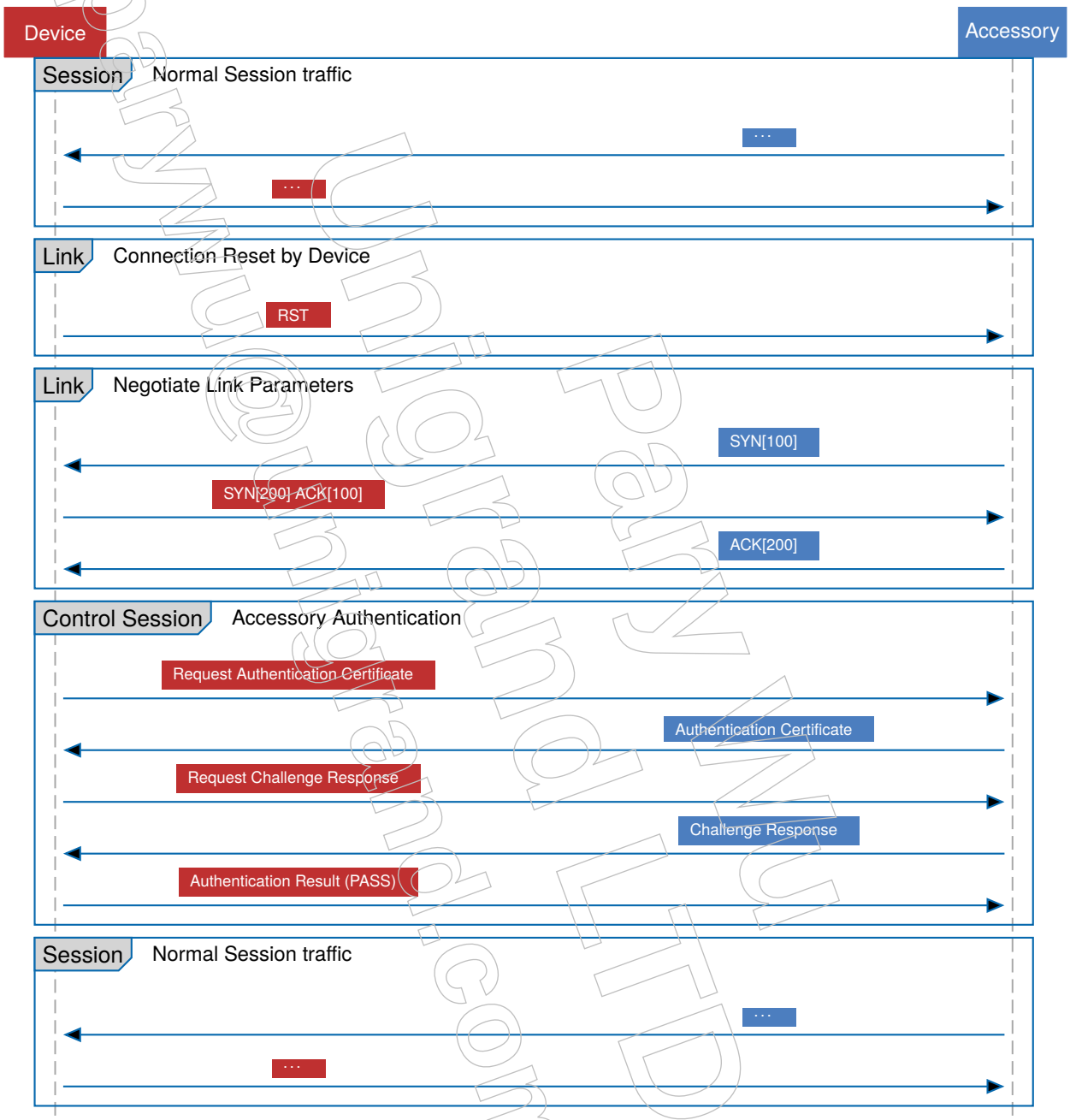
After initialization/setup of the connection with accessory, normal runtime traffic consists of regular ACK packets with data as needed.

At any time, a RST packet may be sent by the Device to the Accessory. Accessory must reset the connection and start the connection sequence again by sending the SYN packet.



Party Wai
Unregistered
Purchase at FTD
www.ftdiphone.com.tw

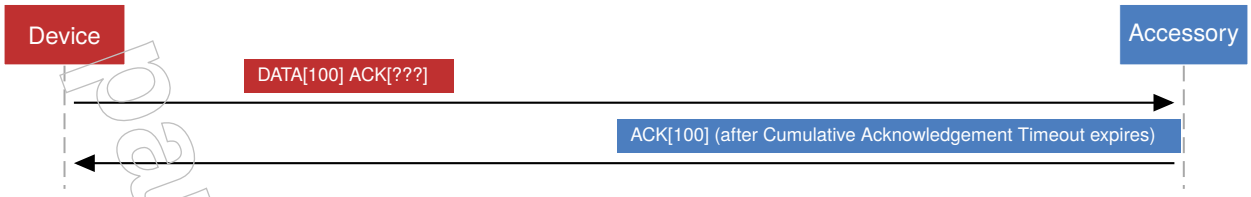
50.8.6 Device Reset of Transport Connection



50.8.7 Cumulative Ack Timeout Expired

In this example, the Maximum Cumulative Acknowledgements link parameter is set to two. The device has sent one packet but has no need to send another one at this time. After the Cumulative Acknowledgement Timeout has expired the accessory assumes that there is no need to wait for a second DATA packet and sends an ACK packet.

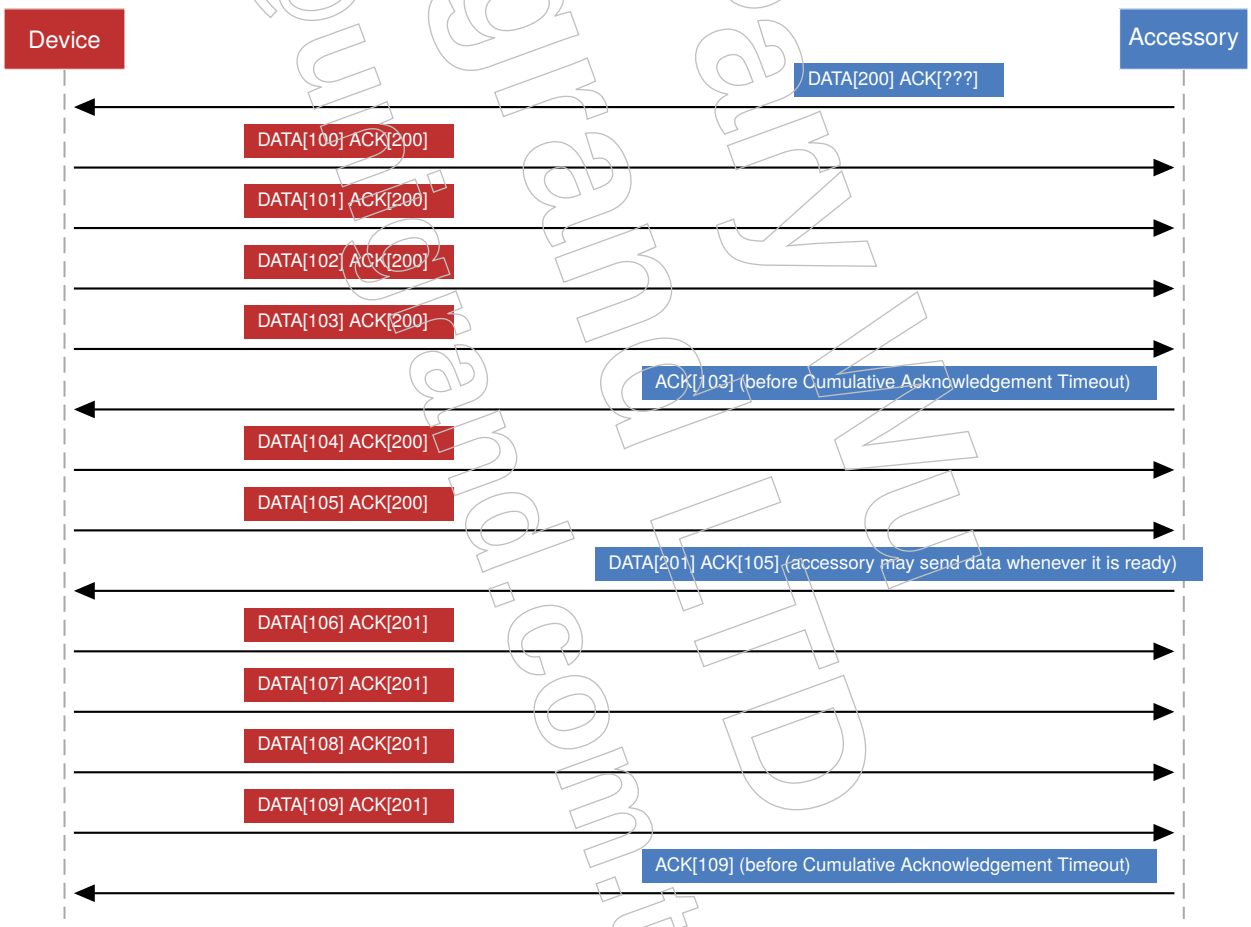
2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.



50.8.8 Continuous Data Transmission with ACKs

ACK packets and Data packets with ACK are used to acknowledge receipt of a previously sent packet and to send data. In this example, the Maximum Cumulative Acknowledgement parameter is set to 4, and the Device does not put more than 4 packets in flight at any given time.

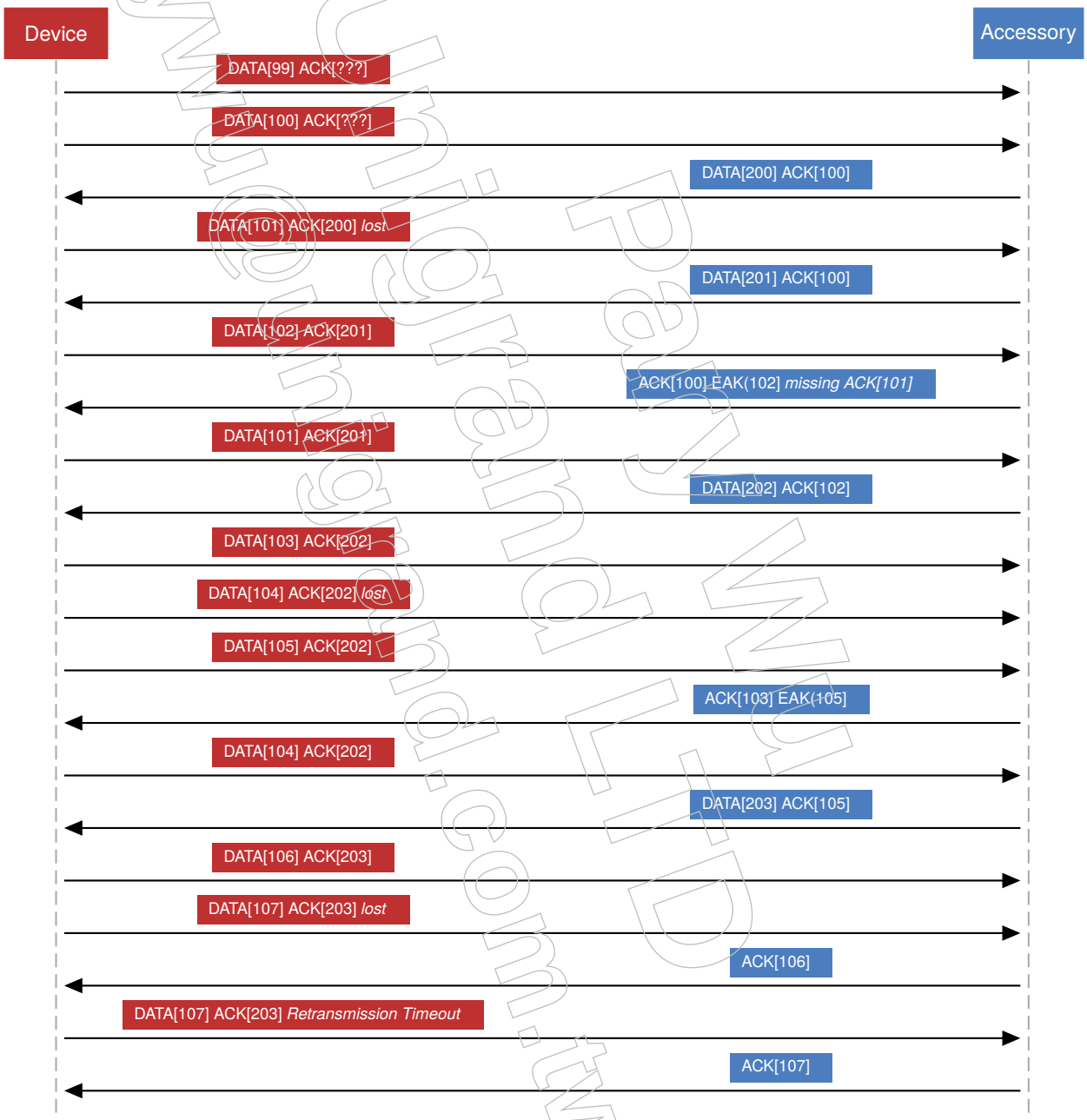
The accessory does not need to wait for Cumulative Acknowledgement Timeout to expire if it has additional packets to send.



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

50.8.9 Resend of missing packets using EAK

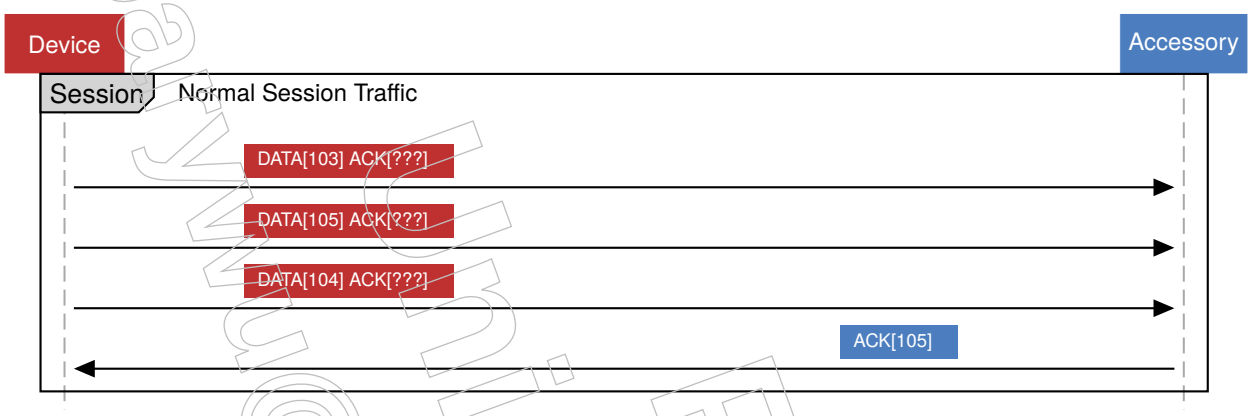
The EAK packet is used to indicate missing packets. Upon receiving an EAK packet, the recipient must resend the missing packets. The packet transfer recipient must wait for acknowledgement of all in-flight packets up to the Maximum Cumulative Acknowledgement parameter, or wait the Cumulative Acknowledgement Timeout to expire before sending an EAK.



Paul Wu [1] D
Unauthorized Distribution
paul@unauthorized.com.tw

50.8.10 Receiving Packets Out Of Order

The accessory will receive packets out of order and ACK the last part of the data it receives.



50.8.11 iAP2 Link Packet Structure Example

Packet Length MSB and *Packet Length LSB* refer to the size of the packet (0x001A). *Control Byte* identifies the type of payload being sent (0x80), SYN packet. *Packet Sequence Number* in this case in randomly picked (0x2B). *Packet Acknowledgement Number* and *Session Identifier* have no meaning in this context. *Header Checksum* is calculated to be 0xE2. *Payload Data* and *Payload Checksum* are described in detail in iAP2 Link Synchronization Payload Example (page 560).

Refer to Packet Structure (page 538) for more details.

Table 50-11 iAP2 Link Packet Structure Example - Accessory SYN Packet

Start of Packet MSB (FF)
Start of Packet LSB (5A)
Packet Length MSB (00)
Packet Length LSB (1A)
Control Byte (80 SYN Packet)
Packet Sequence Number (2B)
Packet Acknowledgement Number (00)
Session Identifier (00)
Header Checksum (E2)
Payload Data

Payton Wu
payton@unionpaytaiwan.com.tw
UnionPay International LTD

(01 05 10 00 04 0B 00 17 03 03 0A 00 01 0B 02 01)
Payload Checksum (A5)

50.8.12 iAP2 Link Synchronization Payload Example

Refer to Link Synchronization Payload (page 542) for more details.

Table 50-12 Link Synchronization Payload Example

Link Version (01)
Maximum Number of Outstanding Packets (05)
Maximum Packet Length MSB (10)
Maximum Packet Length LSB (00)
Retransmission Timeout MSB (04)
Retransmission Timeout LSB (0B)
Cumulative Acknowledgement Timeout MSB (00)
Cumulative Acknowledgement Timeout LSB (17)
Maximum Number of Retransmissions (03)
Maximum Cumulative Acknowledgements (03)
iAP2 Session 1: Session Identifier (0A)
iAP2 Session 1: Session Type (00)
iAP2 Session 1: Session Version (01)
iAP2 Session 2: Session Identifier (0B)
iAP2 Session 2: Session Type (02)
iAP2 Session 3: Session Version (01)

Party Wu Unofficial LTD
party@unofficial.com.tw

51. iAP2 Sessions

Once an iAP2 link has been established, higher level communication between an accessory and an Apple device is structured into one or more sessions. At the very minimum, all accessories must establish a control session for the purposes of authentication and identification. Once initial authentication and identification are complete, the accessory may use additional sessions.

Every session has a session identifier, session type, and session version. The session identifier is used to uniquely identify a session within an iAP2 connection. The session identifier of 0 is reserved for link use, and all sessions used by an accessory must have unique nonzero identifiers. The combination of both session type and session version dictates which session protocol variant is used for all data traffic in that session. The session identifier is part of every iAP2 link packet header.

51.1 Attributes

51.1.1 Type

Table 51-1 iAP2 Session Types

Session Type	Name
0	Control session
1	File Transfer session
2	External Accessory session

All link implementations must have one control session. All other sessions are optional depending on the features implemented. Multiple sessions of the same type are not permitted.

51.1.2 Version

The session version is used to distinguish between different variants of a particular session protocol. The exact nature of these differences depends on the session type. See the sections in the specification on each session type for more details.

Property of Gary Wu
Garywu@icloud.com
Unigrand LTD

51.2 Control Session

The control session has 2 primary goals:

- Notify both the device and accessory of changes in the other's state and/or configuration.
- Provide the accessory with means to request changes in device state and/or configuration.

The control session protocol is a stateless protocol. Each message is an independent entity that has no relationship to any prior message. The device does not retain any information or status about each attached accessory from message to message. An accessory must behave the same way. Specifically, an accessory request for a change in device state and/or configuration must be treated as a request and not a command. The device may or may not choose to honor the request, and will either send an update message to confirm a change in state/configuration or not send any message at all.

All accessories must use control session version 1 with one exception. Accessories supporting the CarPlay feature (see CarPlay (page 244)) must advertise support for control session version 2.

Accessories that support control session version 1 must complete authentication (see Accessory Authentication (page 201)) before sending any other messages.

Accessories that support control session version 2 may send identification messages (see Accessory Identification (page 205)) before authentication completes. They must fall back to control session 1 if the Apple device does not support control session 2 or not support those Apple devices.

Accessories must use control session messages for their specified purpose. Relying on unspecified behavior is likely to cause incompatibilities with future Apple devices and/or iOS releases.

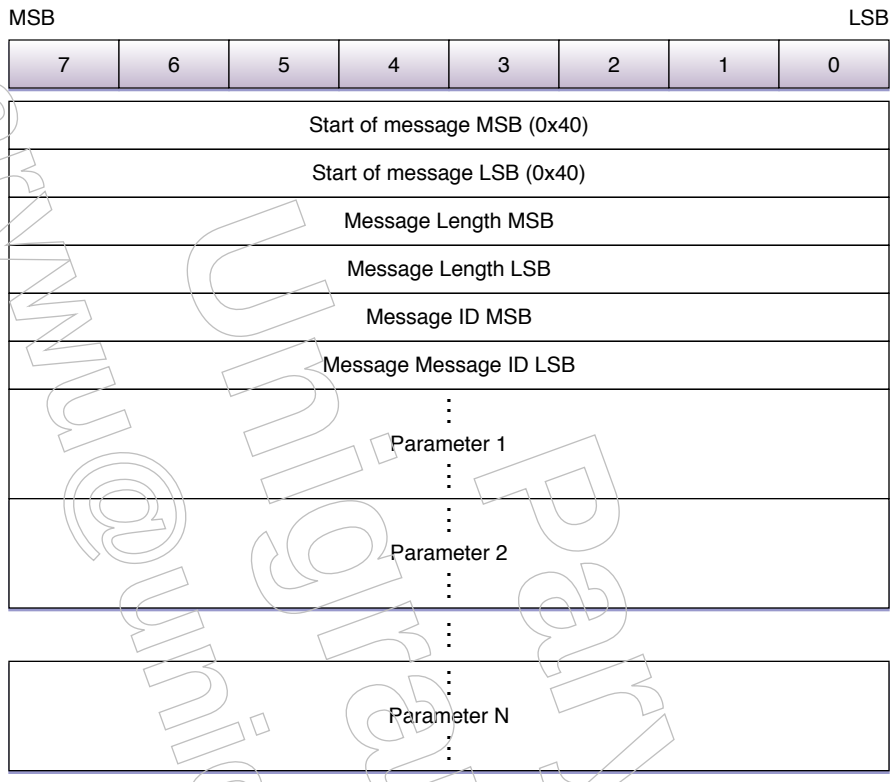
Accessories must rely on the iAP2 link layer to guarantee message delivery. Accessories should never resend messages at the control session layer.

51.2.1 Message Structure

Each message has a header followed by 0 or more parameters or parameter groups:

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Figure 51-1 Control Session Message Structure



Message Length:
Size of the whole message, in bytes, including the Start of Message field, Message Length field, Message ID field and all Parameters.

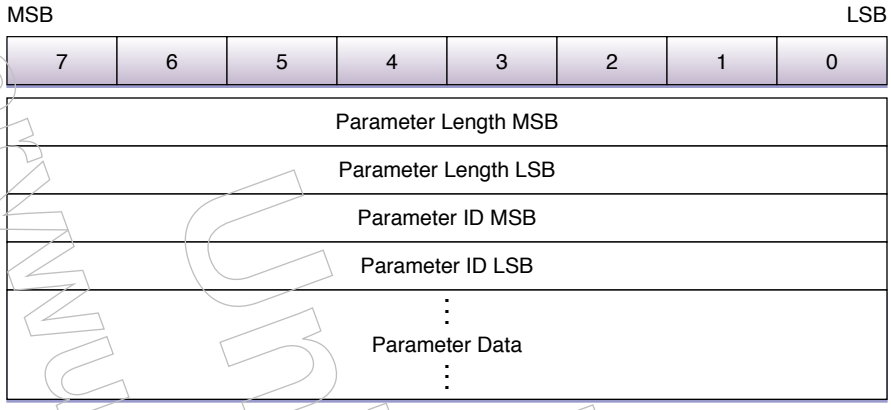
Message ID:
Message Identifier

Parameter:
Parameters for this message (see below)

Parameters are structured as follows:

Party Wu
Unigrand LTD
Party Wu@unigrand.com.tw

Figure 51-2 Control Session Message Parameter Structure



Parameter Length:
Length of parameter, in bytes, including the parameter length field, parameter ID and parameter data.

Parameter ID:
Interpretation is message specific.

Parameter Data:
Type of the data is determined by the parameter ID for this specific message ID.

51.2.2 Message Parsing

All iAP2 control session message parsers must comply with the following requirements:

- Parameters can be specified in any order.
- Received messages must be ignored if the message identifier is unrecognized.
- Received messages must be ignored if any required parameters are missing.
- Received messages must be ignored if any parameters have invalid values.
- Extra unknown parameters in a received message must be ignored and message parsing must proceed with the remaining parameters.

51.2.3 Parameter Types

All parameter bytes are transmitted and received in big-endian order.

51.2.3.1 Number

Only certain types of numbers are permitted in messages:

- **int8** - signed 8-bit byte
- **int16** - signed 16-bit word

Paywu@uniguandigital.com.tw Uniguandigital Ltd

- **int32** - signed 32-bit long
- **int64** - signed 64-bit quadword
- **uint8** - unsigned 8-bit byte
- **uint16** - unsigned 16-bit word
- **uint32** - unsigned 32-bit long
- **uint64** - unsigned 64-bit quadword

If a number parameter has a limited range of permitted values, the range will be specified in the parameter notes. Otherwise, the full range of possible values is assumed to be valid.

51.2.3.2 Enumeration

Written in parameter lists as 'enum'. Unsigned 8-bit byte which corresponds to an exact listing of all of its possible values.

51.2.3.3 Boolean

Written in parameter lists as 'bool'. Special type of Enumeration encoded as a one byte unsigned integer where 0 indicates NO and 1 means YES. All other enumeration values are invalid.

51.2.3.4 String

Written in parameter lists as 'utf8'. Null terminated UTF-8 string. Accessories that receive string parameters must be prepared to handle any valid UTF-8 string, as Apple devices allow users to input strings in all supported languages and character sets, such as Emoji.

51.2.3.5 Blob

Written in parameter lists as 'blob'. The contents of the binary blob will be defined in the parameter notes. Typically, most blobs are arrays of custom data structures.

Note: Accessories must be prepared to handle zero-length blobs.

51.2.3.6 None

Written in parameter lists as 'none'. No parameter data. The mere existence or absence of the parameter in the message has meaning.

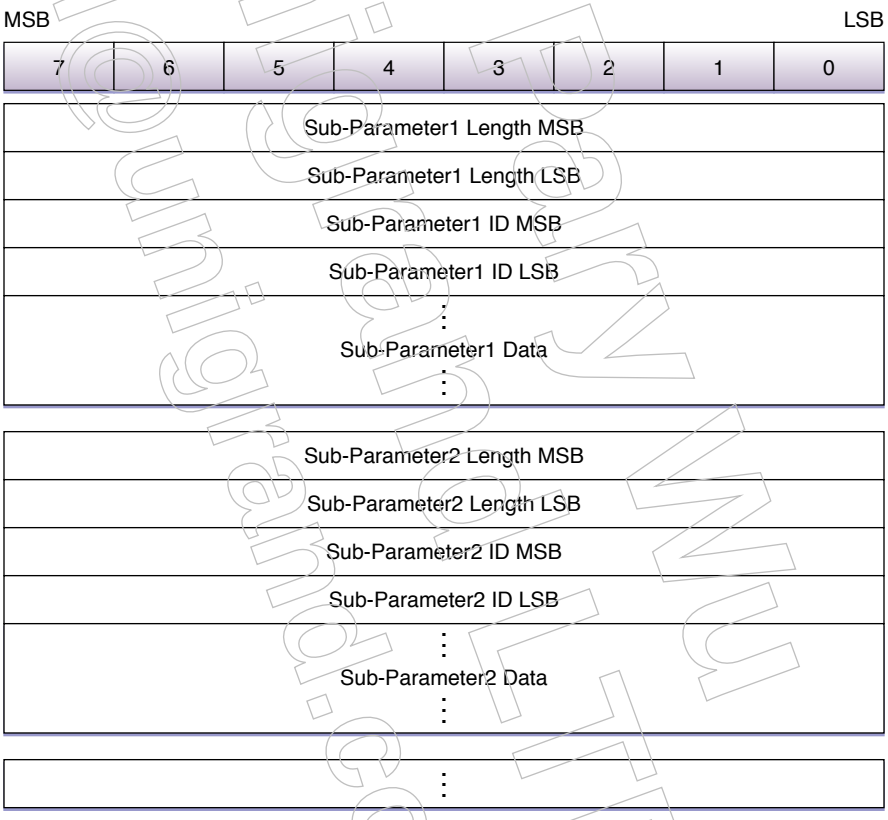
Paul Wu Universal LTD paul.wu@universal.com.tw

Note: Void parameters must have a length of 4 and no payload data.

51.2.3.7 Group

Written in parameter lists as 'group'. Parameters maybe grouped together as a group. Group parameter data consists of a number of sub-parameters each having a Format consisting of Sub-parameter Length, ID, and Data. Sub-parameters may be present in any order within a group.

Figure 51-3 Group Parameter Structure



51.2.4 Message Example

Example StartExternalAccessoryProtocolSession Control Session message

Confidential
Property of Foxlink
Foxlink@foxlink.com.tw

Figure 51-4 StartExternalAccessoryProtocolSession Control Session Message Example

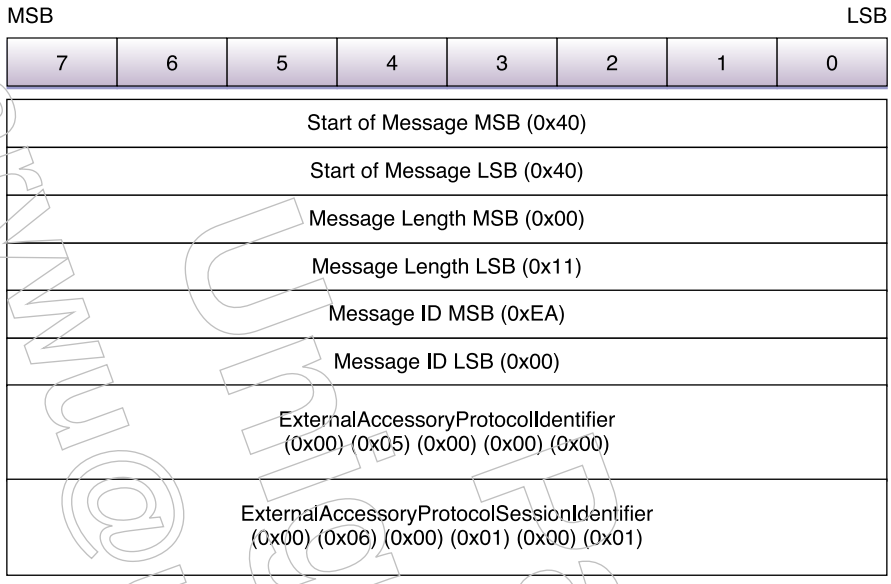


Figure 51-5 ExternalAccessoryProtocolIdentifier Parameter

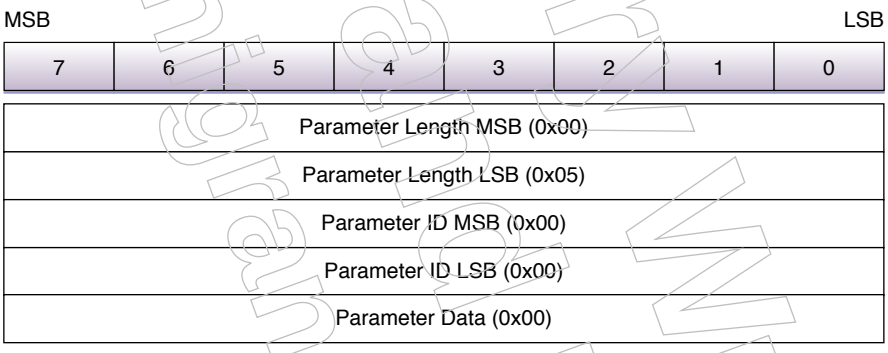
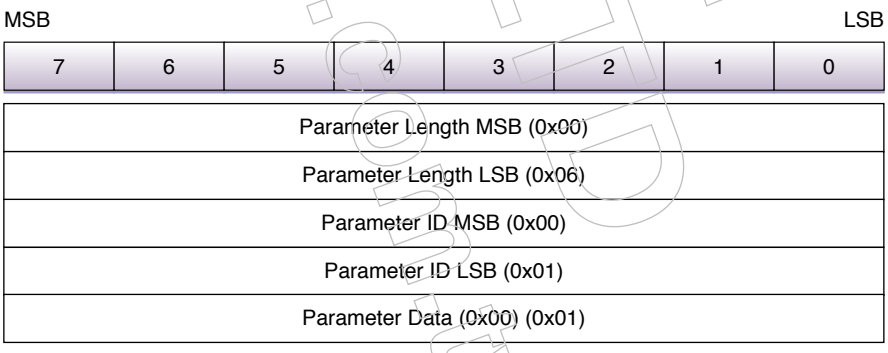


Figure 51-6 ExternalAccessoryProtocolSessionIdentifier Parameter



Example StartNowPlayingUpdates Control Session message with parameter group

patwu@uniguru.com

Figure 51-7 StartNowPlayingUpdates Control Session Message Example

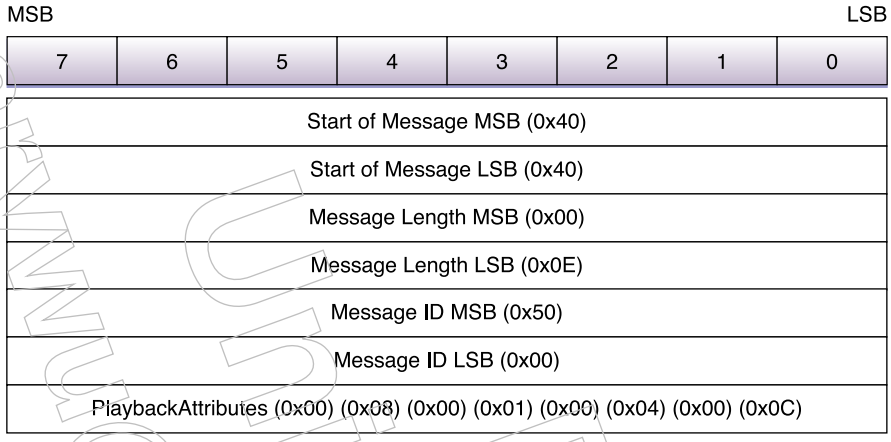
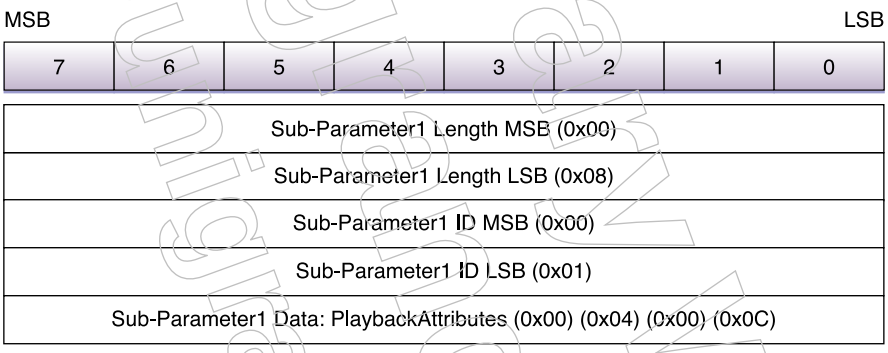


Figure 51-8 PlaybackAttributes Parameter Group



51.3 File Transfer Session

Any accessory that identifies itself as capable of sending or receiving any messages associated with file transfers must set up a file transfer session during link negotiation.

All file transfers are initiated by the sender and have a unique identifier that is also assigned by the sender. Up to 128 simultaneous transfers in each direction are possible, and transfers may be paused and/or canceled before they are complete by either the sender or the receiver.

All file transfer datagrams must fit within a single iAP2 link packet payload.

At this time, the only valid file transfer session version is version 1.

51.3.1 TransferIdentifier

Setup of a file transfer takes place in the control session. The following iAP2 messages can generate a file transfer identifier parameter:

Paid Upgrade Wu
Unregistered/Unlicensed
Patywu@gmail.com

- MediaLibraryUpdate (page 606)
- NowPlayingUpdate (page 614)
- TelephonyCallStateInformation (page 621)

To support file transfers from the Accessory to the Device, every iAP2 connection must maintain a global unsigned 8-bit FileTransferIdentifier that starts at 0 when the connection is established. The FileTransferIdentifier must be incremented to the next inactive FileTransferIdentifer value once it has been used. Once the global FileTransferIdentifier reaches 127 the accessory must immediately reset it to 0 or the lowest inactive FileTransferIdentifier value, whichever is lower, before it initiates the next transfer.

MediaLibraryUpdate and NowPlayingUpdate messages can generate a FileTransferIdentifer parameter for transfers from the Device to the Accessory, which will have parameters ranging from 128 to 255.

51.3.2 Setup Datagram

Once a message with a FileTransferIdentifier parameter has been sent, the sender must issue a Setup datagram that contains the file size in bytes over the file transfer session using the same FileTransferIdentifier.

Table 51-2 File Transfer Session Setup Datagram

Byte	Value
0	FileTransferIdentifier
1	0x04
2	File Size Byte 0
3	File Size Byte 1
4	File Size Byte 2
5	File Size Byte 3
6	File Size Byte 4
7	File Size Byte 5
8	File Size Byte 6
9	File Size Byte 7

Paul Chang Paul@Digiland.com.tw Digiland LTD

51.3.3 Start

Upon receiving the Setup datagram from the sender, the receiver may send a Start datagram back to the sender to signify that it is ready to receive file data.

Table 51-3 File Transfer Session Start Datagram

Byte	Value
0	FileTransferIdentifier
1	0x01

51.3.4 FirstData

The sender may start sending file data once it receives a Start datagram. The first datagram is unique and must take the following format:

Table 51-4 File Transfer Session FirstData Datagram

Byte	Value
0	FileTransferIdentifier
1	0x80
2	Data Byte 0
...	...
n+2	Data Byte n

Only one FirstData datagram may be sent for any given assigned FileTransferIdentifier. The sender must switch to Data datagrams for all subsequent file data until the end of the file.

51.3.5 FirstAndOnlyData

If the file is so short as to fit in one session datagram, the sender may send a FirstAndOnlyData datagram as follows:

Pary Wu
Unigrand LTD
paryw@unigrand.com.tw

Table 51-5 File Transfer Session FirstAndOnlyData Datagram

Byte	Value
0	FileTransferIdentifier
1	0xC0
2	Data Byte 0
...	...
n+2	Data Byte n

Only one FirstAndOnlyData datagram may be sent for any given assigned FileTransferIdentifier. The file transfer is considered complete upon receipt of the FirstAndOnlyData datagram.

51.3.6 Data

Regular file data datagrams must take the following format:

Table 51-6 File Transfer Session Data Datagram

Byte	Value
0	FileTransferIdentifier
1	0x00
2	Data Byte 0
...	...
n+2	Data Byte n

51.3.7 LastData

The last object data datagram is unique and must take the following format:

Table 51-7 File Transfer Session LastData Datagram

Byte	Value
0	FileTransferIdentifier
1	0x40

Confidentiary Wu
Courtyard and LTD
courtyard.com.tw

Byte	Value
2	Data Byte 0
...	...
n+2	Data Byte n

51.3.8 Cancel

The Cancel datagram may be sent at any time by either the sender or receiver after the Setup datagram is sent or received. The FileTransferIdentifier must be marked as inactive by both the sender and receiver after it is transmitted.

The Cancel datagram must be sent by any accessory or device if any other File Transfer Session datagram is received with an inactive FileTransferIdentifier.

Table 51-8 File Transfer Session Cancel Datagram

Byte	Value
0	FileTransferIdentifier
1	0x02

51.3.9 Pause

The Pause datagram may be sent at any time by the sender or receiver after the first Start datagram has been sent by the receiver. Unlike the Cancel datagram, the FileTransferIdentifier remains active, but no FirstData, Data, or LastData datagrams may be transmitted until the end that sent the Pause datagram sends a Start datagram with the same FileTransferIdentifier. Data datagrams will resume from the point where the transfer left off.

Table 51-9 File Transfer Session Pause Datagram

Byte	Value
0	FileTransferIdentifier
1	0x03

Party Wei Ubiquand LTD
partywei@ubiquand.com.tw

51.3.10 Success

The Success datagram must be sent by the receiver after it has received and processed all file data. The sender may choose to delete the file or otherwise dispose of it after receiving this datagram from the receiver.

The FileTransferIdentifier must be marked as inactive by both the sender and receiver after this datagram is transmitted.

Table 51-10 File Transfer Session Success Datagram

Byte	Value
0	FileTransferIdentifier
1	0x05

51.3.11 Failure

The Failure datagram must be sent by the receiver if it has received all file data but is unable to store it. This sender may choose to resend the file or not, depending on the circumstances.

The FileTransferIdentifier must be marked as inactive by both the sender and receiver after this datagram is transmitted.

Table 51-11 File Transfer Session Failure Datagram

Byte	Value
0	FileTransferIdentifier
1	0x06

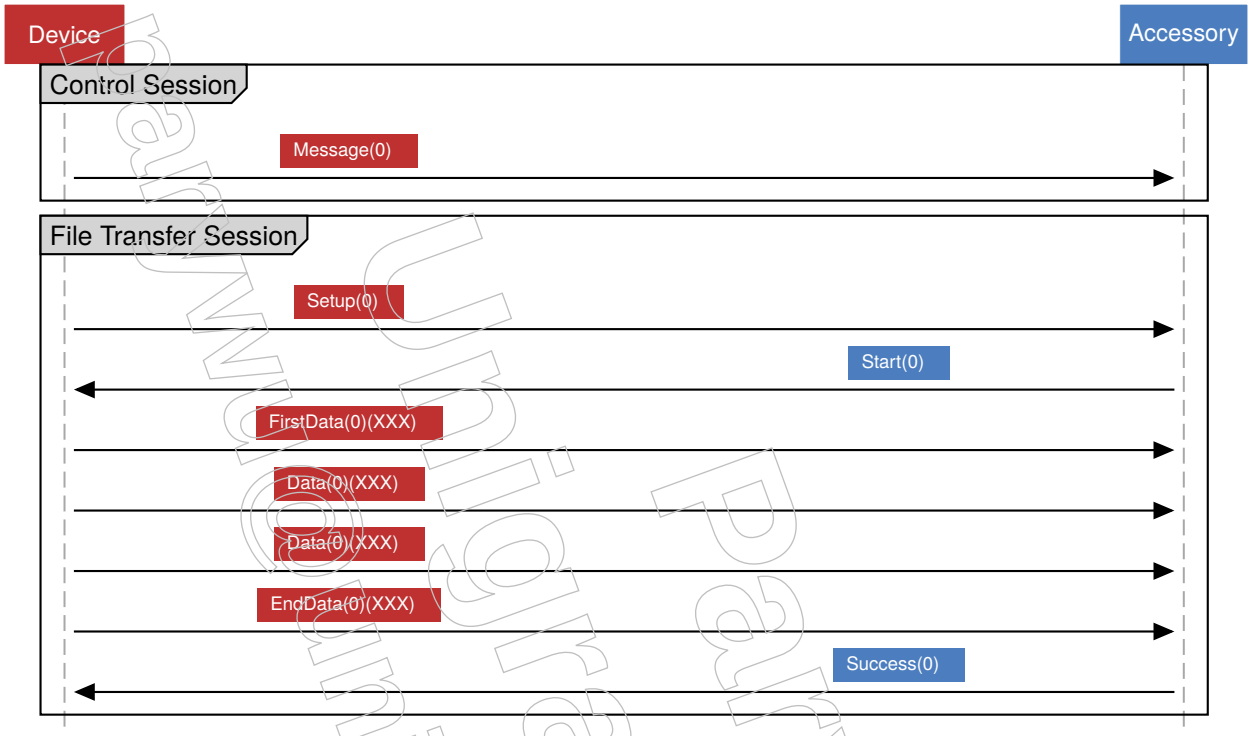
51.3.12 Examples

In all examples, the device is playing the role of the file sender, and the accessory is the file receiver. These roles can be reversed depending on the needs of a particular accessory interface feature

51.3.12.1 Typical Transfer

In this example, the device is sending one file to the accessory, and no other transfers are taking place.

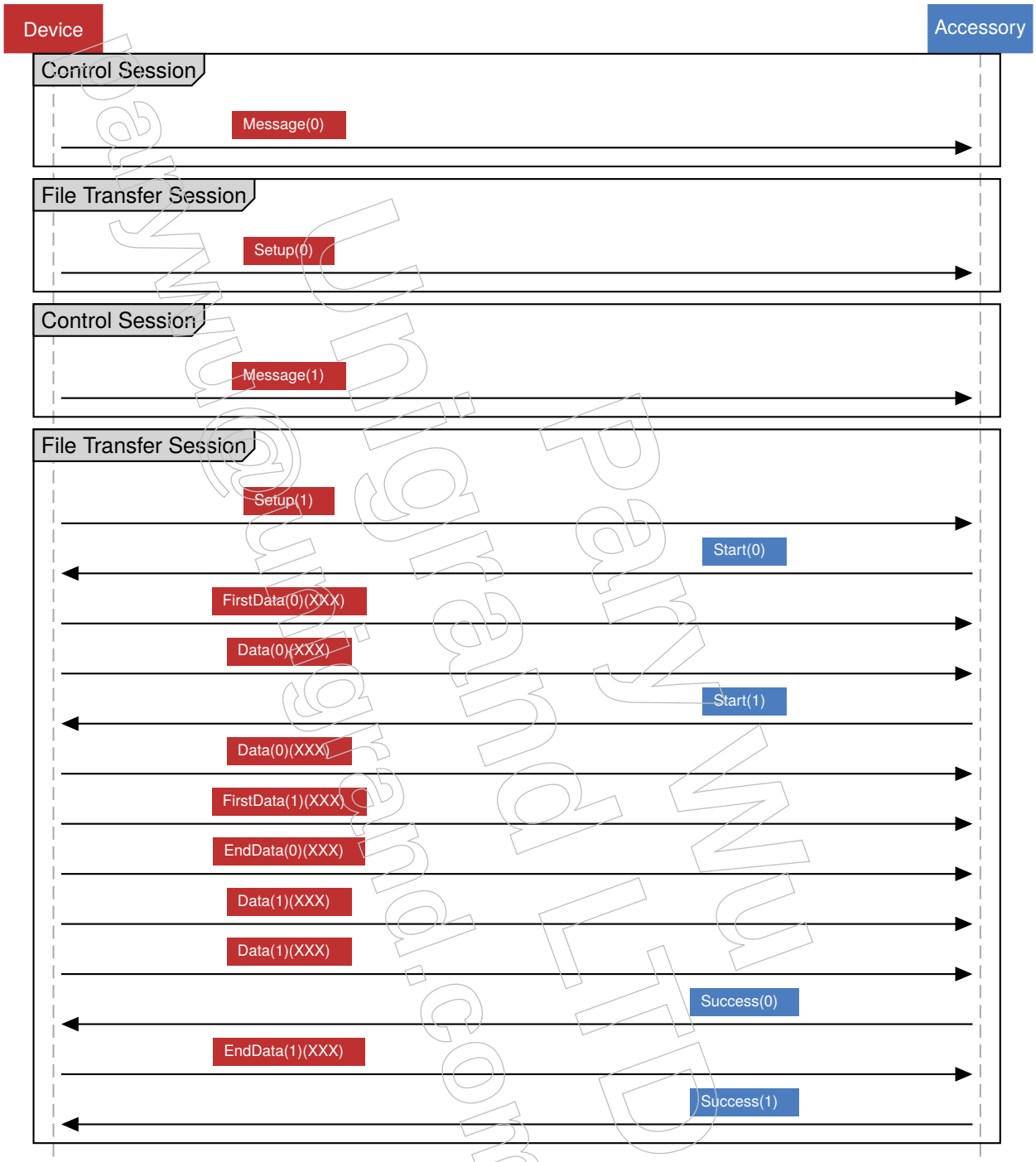
Paly Wu
paly_wu@unigrand.com.tw
Unigrand LTD



51.3.12.2 Two Simultaneous Transfers

In this example, the device is sending two files to the accessory simultaneously. Note that all of the traffic can be interleaved arbitrarily, so long as the message containing the FileTransferIdentifier is sent before any file transfer session traffic starts.

Paywu Digital LTD
paywu@digital.com.tw



51.4 External Accessory Session

The external accessory (EA) session is used by the External Accessory Protocol (page 372) to create one or more bidirectional serial data streams, or transfers, between an accessory and iOS apps.

Paywu International Ltd
paywu@paywu.com.tw

At this time, the only valid external accessory session version is version 1.

51.4.1 Setup

Setup of an EA session takes place in the control session. The accessory will receive a StartExternalAccessoryProtocolSession (page 598) message with assigned `ExternalAccessoryProtocolIdentifier` and `ExternalAccessoryProtocolSessionIdentifier` parameters.

The `ExternalAccessoryProtocolIdentifier` is defined by the accessory during identification; an accessory may define and support multiple supported protocols.

The `ExternalAccessorySessionIdentifier` is unique to each EA connection between an app and the accessory. Accessories must use this identifier to distinguish between datagrams sent over the EA session.

Accessories that declare an EA session during link synchronization must declare support for one or more `ExternalAccessoryProtocol` parameters during accessory identification as specified in External Accessory Protocol (page 372).

Accessories must not send any EA session datagrams with an unassigned transfer identifier. Conversely, accessories must ignore any EA session datagrams with unassigned session identifiers.

51.4.2 ExternalAccessorySession Datagram

All datagrams for an EA session must follow this format:

Table 51-12 ExternalAccessorySession Datagram

Byte	Value
0	ExternalAccessorySessionIdentifier Byte 0
1	ExternalAccessorySessionIdentifier Byte 1
2+	External Accessory Session Data

All EA session datagrams must fit within a single iAP2 link packet payload.

Downloaded by Yu Wu (wuyu@eastunited.com) LTD

52. iAP2 Control Session Messages

52.1 Accessory Authentication

For more information, see Accessory Authentication (page 201).

52.1.1 RequestAuthenticationCertificate

Source	ID
Device	0xAA00

Table 52-1 `RequestAuthenticationCertificate` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.1.2 AuthenticationCertificate

Source	ID
Accessory	0xAA01

Table 52-2 `AuthenticationCertificate` message parameters

Name	ID	Type	#	Notes
AuthenticationCertificate	0	blob	1	Accessory's X.509 certificate

52.1.3 RequestAuthenticationChallengeResponse

Source	ID
Device	0xAA02

Party Wu Chigrand LTD partywu@chigrand.com.tw

Table 52-3 `RequestAuthenticationChallengeResponse` message parameters

Name	ID	Type	#	Notes
AuthenticationChallenge	0	blob	1	Random #

52.1.4 AuthenticationResponse

Source	ID
Accessory	0xAA03

Table 52-4 `AuthenticationResponse` message parameters

Name	ID	Type	#	Notes
AuthenticationResponse	0	blob	1	Computed challenge response

52.1.5 AuthenticationFailed

Source	ID
Device	0xAA04

Table 52-5 `AuthenticationFailed` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.1.6 AuthenticationSucceeded

Source	ID
Device	0xAA05

Table 52-6 `AuthenticationResponseSucceeded` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

Party Wu and LTD
tonywu@unigrand.com.tw

52.2 Accessory Identification

For more information, see Accessory Identification (page 205).

52.2.1 StartIdentification

Source	ID
Device	0x1D00

Table 52-7 `StartIdentification` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.2.2 IdentificationInformation

Source	ID
Accessory	0x1D01

Table 52-8 `IdentificationInformation` message parameters

Name	ID	Type	#	Notes
Name	0	utf8	1	Must match the accessory's markings and packaging. A blank string is not allowed.
ModelIdentifier	1	utf8	1	Must match the accessory's markings and packaging. A blank string is not allowed.
Manufacturer	2	utf8	1	Must match the accessory's markings and packaging. A blank string is not allowed.
SerialNumber	3	utf8	1	Must match the accessory's markings and packaging. A blank string is not allowed.

Paryl Wei LTD
Unigrand.com.tw
Unigrand@unigrand.com

Name	ID	Type	#	Notes
FirmwareVersion	4	utf8	1	Must uniquely reflect the current revision of the accessory's firmware. A blank string is not allowed.
HardwareVersion	5	utf8	1	Must uniquely reflect the current revision of the accessory's hardware. A blank string is not allowed.
MessagesSentByAccessory	6	blob	1	The exhaustive set of unique messages that this accessory will send. This set is expressed as an array of uint16 message identifiers (uint16[]).
MessagesReceivedFromDevice	7	blob	1	The exhaustive set of unique messages that this accessory expects to receive. This set is expressed as an array of uint16 message identifiers (uint16[]).
PowerProvidingCapability	8	enum	1	See Table 52-9 (page 581). This must be 'None' if the accessory does not provide power to the Apple device.
MaximumCurrentDrawnFromDevice	9	uint16	1	Maximum current drawn by accessory from Accessory Power pin in mA. This must be 0 if accessory does not draw power from the Apple device.
SupportedExternalAccessoryProtocol	10	group	0+	See Table 52-10 (page 581)
AppMatchTeamID	11	utf8	0/1	See App Match (page 212)
CurrentLanguage	12	utf8	1	The accessory's current active language setting. Must be one of the supported languages.

parkwu@uniqland.com.tw Uniqland LTD Parkwu

Name	ID	Type	#	Notes
SupportedLanguage	13	utf8	1+	A language supported by the accessory. Use the ISO 639-1 designation unless it is not available, in which case use the ISO 639-2 designation. For a complete list of ISO 639-1 and ISO 639-2 codes, see http://www.loc.gov/standards/iso639-2/php/English_list.php.
SerialTransportComponent	14	group	0/1	See Table 52-15 (page 583)
USBDeviceTransportComponent	15	group	0/1	See Table 52-12 (page 582)
USBHostTransportComponent	16	group	0/1	See Table 52-14 (page 583)
BluetoothTransportComponent	17	group	0+	See Table 52-16 (page 584)
iAP2HIDComponent	18	group	0+	See Table 52-17 (page 584)
VehicleInformationComponent	20	group	0/1	See Table 52-18 (page 584)
VehicleStatusComponent	21	group	0/1	See Table 52-20 (page 585)
LocationInformationComponent	22	group	0/1	See Table 52-21 (page 585)
USBHostHIDComponent	23	group	0+	See Table 52-22 (page 586)

Table 52-9 `PowerProvidingCapability` enum

Value	Meaning
0	None
1	Reserved
2	Advanced

Table 52-10 `ExternalAccessoryProtocol` parameter group

Name	ID	Type	#	Notes
ExternalAccessoryProtocolIdentifier	0	uint8	1	All ExternalAccessoryProtocol identifiers must be unique

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
ExternalAccessoryProtocolName	1	utf8	1	
ExternalAccessoryProtocolMatchAction	2	enum	1	See Table 52-11 (page 582) and App Match (page 212)
NativeTransportComponentIdentifier	3	uint16	0/1	Must refer to the `TransportComponent-Identifier` of a declared Table 52-14 (page 583)

Table 52-11 `MatchAction` enum

Value	Meaning
0	The device will not prompt the user to find a matching app, and there will not be a *Find App For This Accessory* button in *Settings* > *General* > *About* > *'Accessory Name'*
1	The device may prompt the user to find a matching app, and there will be a *Find App For This Accessory* button in *Settings* > *General* > *About* > *'Accessory Name'*. See App Match (page 212) for more information.
2	The device will not prompt the user to find a matching app, but there will be a *Find App For This Accessory* button in *Settings* > *General* > *About* > *'Accessory Name'*. See App Match (page 212) for more information.

Table 52-12 `USBDeviceTransportComponent` parameter group

Name	ID	Type	#	Notes
TransportComponentIdentifier	0	uint16	1	All TransportComponent identifiers must be unique
TransportComponentName	1	utf8	1	
TransportSupportsiAP2Connection	2	none	0/1	
USBDeviceSupportedAudioSampleRate	3	enum	0+	See Table 52-13 (page 582)

Table 52-13 `USBDeviceModeAudioSampleRate` enum

Value	Meaning
0	8000 Hz

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Value	Meaning
1	11025 Hz
2	12000 Hz
3	16000 Hz
4	22050 Hz
5	24000 Hz
6	32000 Hz
7	44100 Hz
8	48000 Hz

Table 52-14 `USBHostTransportComponent` parameter group

Name	ID	Type	#	Notes
TransportComponentIdentifier	0	uint16	1	All TransportComponent identifiers must be unique
TransportComponentName	1	utf8	1	
TransportSupportsiAP2Connection	2	none	0/1	
USBHostTransportCarPlayInterfaceNumber	3	uint8	0/1	The accessory's NCM Control Interface number. See CarPlay (page 244), CarPlay over USB (page 250) for more details.

Table 52-15 `SerialTransportComponent` parameter group

Name	ID	Type	#	Notes
TransportComponentIdentifier	0	uint16	1	All TransportComponent identifiers must be unique
TransportComponentName	1	utf8	1	
TransportSupportsiAP2Connection	2	none	0/1	

Paywand LTD
Unigrand.com.tw
paywu@unigrand.com

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Table 52-16 `BluetoothTransportComponent` parameter group

Name	ID	Type	#	Notes
TransportComponentIdentifier	0	uint16	1	All TransportComponent identifiers must be unique
TransportComponentName	1	utf8	1	
TransportSupportsiAP2Connection	2	none	0/1	
BluetoothTransportMediaAccessControlAddress	3	blob	1	A valid 6-byte IEEE EUI-48 identifier (uint8[6])

Table 52-17 `iAP2HIDComponent` parameter group

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	All HIDComponentIdentifiers must be unique
HIDComponentName	1	utf8	1	
HIDComponentFunction	2	enum	1	See Table 52-23 (page 586)

Table 52-18 `VehicleInformationComponent` parameter group

Name	ID	Type	#	Notes
Identifier	0	uint16	1	All Identifiers must be unique.
Name	1	utf8	1	
EngineType	2	enum	0+	See Table 52-19 (page 584).

Table 52-19 `EngineTypes` enum

Value	Meaning
0	Gasoline
1	Diesel
2	Electric

Paul Wu (paulwu@ulgrandtech.com.tw)

Value	Meaning
3	CNG

Table 52-20 `VehicleStatusComponent` parameter group

Name	ID	Type	#	Notes
Identifier	0	uint16	1	All Identifiers must be unique.
Name	1	utf8	1	
Range	3	none	0/1	
OutsideTemperature	4	none	0/1	
RangeWarning	6	none	0/1	

Table 52-21 `LocationInformationComponent` parameter group

Name	ID	Type	#	Notes
Identifier	0	uint16	1	All Identifiers must be unique.
Name	1	utf8	1	
GlobalPositioningSystemFixData	17	none	0/1	If present, the accessory is capable of generating NMEA GPGGA sentences.
RecommendedMinimumSpecificGPSTransitData	18	none	0/1	If present, the accessory is capable of generating NMEA GPRMC sentences.
GPSSatellitesInView	19	none	0/1	If present, the accessory is capable of generating NMEA GPGSV sentences.
VehicleSpeedData	20	none	0/1	If present, the accessory is capable of generating NMEA PASCD sentences.

Taiwan Unigrand Ltd. Taiwan Unigrand.com.tw

Name	ID	Type	#	Notes
VehicleGyroData	21	none	0/1	If present, the accessory is capable of generating NMEA PAGCD sentences.
VehicleAccelerometerData	22	none	0/1	If present, the accessory is capable of generating NMEA PAACD sentences.
VehicleHeadingData	23	none	0/1	If present, the accessory is capable of generating NMEA GPHDT sentences.

Table 52-22 `USBHostHIDComponent` parameter group

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	All HIDComponentIdentifiers must be unique
HIDComponentName	1	utf8	1	
HIDComponentFunction	2	enum	1	See Table 52-23 (page 586)
USBHostTransportComponentIdentifier	3	uint16	1	Must refer to a Table 52-14 (page 583)
USBHostTransportInterfaceNumber	4	uint16	1	Must match the accessory's corresponding USB device interface descriptor. If more than one USBHostHIDComponent is present, the accessory must present multiple USB HID interfaces with unique interface numbers.

Table 52-23 `HIDComponentFunction` enum

Value	Meaning
0	Keyboard

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Value	Meaning
1	Media Playback Remote
2	AssistiveTouch Pointer
3	Standard Gamepad (Form-Fitting)
4	Extended Gamepad (Form-Fitting)
6	Extended Gamepad (Non Form-Fitting)
7	Assistive Switch Control
8	Headphone

52.2.3 IdentificationAccepted

Source	ID
Device	0x1D02

Table 52-24 `IdentificationAccepted` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.2.4 IdentificationRejected

Source	ID
Device	0x1D03

Table 52-25 `IdentificationRejected` message parameters

Name	ID	Type	#	Notes
Name	0	none	0/1	
ModelIdentifier	1	none	0/1	
Manufacturer	2	none	0/1	

Pary Wu LTD raywu@uniband.com.tw

Name	ID	Type	#	Notes
SerialNumber	3	none	0/1	
FirmwareVersion	4	none	0/1	
HardwareVersion	5	none	0/1	
MessagesSentByAccessory	6	blob	0/1	The set of unsupported messages sent by the accessory. This set is expressed as an array of uint16 message identifiers (uint16[])
MessagesReceivedFromDevice	7	blob	0/1	The set of unsupported messages received from the device. This set is expressed as an array of uint16 message identifiers (uint16[])
PowerProvidingCapability	8	none	0/1	
MaximumCurrentDrawnFromDevice	9	none	0/1	
SupportedExternalAccessoryProtocol	10	none	0/1	One or more of the External Accessory Protocols is not supported by the device
AppMatchTeamID	11	none	0/1	
CurrentLanguage	12	none	0/1	
SupportedLanguage	13	none	0/1	One or more of the identified languages is not supported by the device
SerialTransportComponent	14	none	0/1	
USBDeviceTransportComponent	15	none	0/1	
USBHostTransportComponent	16	none	0/1	
BluetoothTransportComponent	17	none	0/1	One or more of the identified Bluetooth Transport components is not supported by the device
iAP2HIDComponent	18	none	0+	One or more of the identified HID components is not supported by the device

Confidential Property of Unigrand Ltd.
Unigrand@unigrand.com.tw

Name	ID	Type	#	Notes
VehicleInformationComponent	20	none	0/1	One of more of the identified Vehicle Information components is not supported by the device
VehicleStatusComponent	21	none	0/1	One of more of the identified Vehicle Status components is not supported by the device
LocationInformationComponent	22	none	0/1	One of more of the identified Location Information components is not supported by the device
USBHostHIDComponent	23	none	0+	One or more of the identified HID components is not supported by the device

52.2.5 CancelIdentification

Source	ID
Accessory	0x1D05

Table 52-26 `CancelIdentification` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.2.6 IdentificationInformationUpdate

Source	ID
Accessory	0x1D06

Table 52-27 `IdentificationInformationUpdate` message parameters

Name	ID	Type	#	Notes
Name	0	utf8	1	
ModelIdentifier	1	utf8	1	

Pary Wu Unigrand LTD paywu@unigrand.com.tw

Name	ID	Type	#	Notes
Manufacturer	2	utf8	1	
SerialNumber	3	utf8	1	
FirmwareVersion	4	utf8	1	
HardwareVersion	5	utf8	1	
CurrentLanguage	6	utf8	1	

52.3 App Launch

For more information, see App Launch (page 210).

52.3.1 RequestAppLaunch

Source	ID
Accessory	0xEA02

Table 52-28 `RequestAppLaunch` message parameters

Name	ID	Type	#	Notes
AppBundleID	0	utf8	1	uniform type identifer (UTI) in reverse-DNS format, e.g. `com.Ajax.hello`
LaunchAlert	1	enum	0/1	see Table 52-29 (page 590)

Table 52-29 `AppLaunchMethod` enum

Value	Meaning
0	Launch with user alert (default)
1	Launch without any alert, but device must be unlocked and on Home Screen or Music/Video app

Party Wi
parwu@uniqrand.com.tw

52.4 AssistiveTouch

For more information, see AssistiveTouch (page 214).

52.4.1 StartAssistiveTouch

Source	ID
Accessory	0x5400

Table 52-30 `StartAssistiveTouch` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.4.2 StopAssistiveTouch

Source	ID
Accessory	0x5401

Table 52-31 `StopAssistiveTouch` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.4.3 StartAssistiveTouchInformation

Source	ID
Accessory	0x5402

Table 52-32 `StartAssistiveTouchInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

Party Wu Unigrand LTD
Party Wu@unigrand.com.tw

52.4.4 AssistiveTouchInformation

Source	ID
Device	0x5403

Table 52-33 `AssistiveTouchInformation` message parameters

Name	ID	Type	#	Notes
IsEnabled	0	bool	1	

52.4.5 StopAssistiveTouchInformation

Source	ID
Accessory	0x5404

Table 52-34 `StopAssistiveTouchInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.5 Bluetooth Connection

For more information, see Bluetooth Connection (page 235).

52.5.1 BluetoothComponentInformation

Source	ID
Accessory	0x4E01

Table 52-35 `BluetoothComponentInformation` message parameters

Name	ID	Type	#	Notes
BluetoothComponentStatus	0	group	0+	See Table 52-36 (page 593)

Party Wu
LTD
Shanghai and Inland
wu@uniland.com.tw

Table 52-36 `BluetoothComponentStatus` parameter group

Name	ID	Type	#	Notes
ComponentIdentifier	0	uint16	1	See Table 52-16 (page 584)
ComponentEnabled	1	bool	1	`true` if the Bluetooth component is ready for connections to the Apple device

52.5.2 StartBluetoothConnectionUpdates

Source	ID
Accessory	0x4E03

Table 52-37 `StartBluetoothConnectionUpdates` message parameters

Name	ID	Type	#	Notes
BluetoothTransportComponentIdentifier	0	uint16	1+	See Table 52-16 (page 584)

52.5.3 BluetoothConnectionUpdate

Source	ID
Device	0x4E04

Table 52-38 `BluetoothConnectionUpdate` message parameters

Name	ID	Type	#	Notes
BluetoothTransportComponentIdentifier	0	uint16	1	See Table 52-16 (page 584)
ConnectedBluetoothProfiles	1	group	0/1	See Table 52-39 (page 593)

Table 52-39 `BluetoothComponentProfiles` parameter group

Name	ID	Type	#	Notes
BluetoothHandsFree	0	none	0/1	
BluetoothPhoneBookAccess	1	none	0/1	

Pay Wu (paywu@uniguru.com) Uniguru and Ltd

Name	ID	Type	#	Notes
BluetoothAudioVideoRemoteControl	3	none	0/1	
BluetoothAdvancedAudioDistribution	4	none	0/1	
BluetoothHumanInterfaceDevice	5	none	0/1	
BluetoothiAP2Link	7	none	0/1	
BluetoothPersonalAreaNetworkAccessPoint	8	none	0/1	Apple device is in the Access Point role
BluetoothMessageAccess	9	none	0/1	
BluetoothPersonalAreaNetworkClient	12	none	0/1	Apple device is in the Client role

52.5.4 StopBluetoothConnectionUpdates

Source	ID
Accessory	0x4E05

Table 52-40 `StopBluetoothConnectionUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.6 Device Authentication

For more information, see Device Authentication (page 362).

52.6.1 RequestDeviceAuthenticationCertificate

Source	ID
Accessory	0xAA10

Party Wu LTD
party@unigrand.com.tw

Table 52-41 `RequestDeviceAuthenticationCertificate` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.6.2 DeviceAuthenticationCertificate

Source	ID
Device	0xAA11

Table 52-42 `DeviceAuthenticationCertificate` message parameters

Name	ID	Type	#	Notes
DeviceAuthenticationCertificate	0	blob	1	Device's X.509 certificate

52.6.3 RequestDeviceAuthenticationChallengeResponse

Source	ID
Accessory	0xAA12

Table 52-43 `RequestDeviceAuthenticationChallengeResponse` message parameters

Name	ID	Type	#	Notes
DeviceAuthenticationChallenge	0	blob	1	Random #

52.6.4 DeviceAuthenticationResponse

Source	ID
Device	0xAA13

Table 52-44 `DeviceAuthenticationResponse` message parameters

Name	ID	Type	#	Notes
DeviceAuthenticationResponse	0	blob	1	Computed challenge response

Property of Unigrand Ltd. terrywu@unigrand.com.tw

52.6.5 DeviceAuthenticationFailed

Source	ID
Accessory	0xAA14

Table 52-45 `DeviceAuthenticationFailed` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.6.6 DeviceAuthenticationSucceeded

Source	ID
Accessory	0xAA15

Table 52-46 `DeviceAuthenticationResponseSucceeded` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.7 Device Notifications

For more information, see Device Notifications (page 365).

52.7.1 DeviceInformationUpdate

Source	ID
Device	0x4E09

Table 52-47 `DeviceInformationUpdate` message parameters

Name	ID	Type	#	Notes
DeviceName	0	utf8	0/1	Device name as shown in Settings, i.e. "John Doe's iPhone"

Party Wu Digrand LTD
partywu@unigrand.com.tw

52.7.2 DeviceLanguageUpdate

Source	ID
Device	0x4E0A

Table 52-48 `DeviceLanguageUpdate` message parameters

Name	ID	Type	#	Notes
DeviceLanguage	0	utf8	0/1	ISO 639-1 or ISO 639-2 designation for the current device language. For a complete list of ISO 639-1 and ISO 639-2 codes, see http://www.loc.gov/standards/iso639-2/php/English_list.php

52.7.3 DeviceTimeUpdate

Source	ID
Device	0x4E0B

Table 52-49 `DeviceTimeUpdate` message parameters

Name	ID	Type	#	Notes
SecondsSinceReferenceDate	0	uint64	1	Time interval in seconds since reference date (Jan 1, 1970, GMT)
TimeZoneOffsetMinutes	1	int16	1	Difference in minutes between the device time zone and GMT
DaylightSavingsOffsetMinutes	2	int8	1	Daylight savings time offset in minutes

52.8 External Accessory Protocol

For more information, see External Accessory Protocol (page 372).

Confidential Property of Apple Inc.
Registered to tianyu@tianyuniganan.com

52.8.1 StartExternalAccessoryProtocolSession

Source	ID
Device	0xEA00

Table 52-50 `StartExternalAccessoryProtocolSession` message parameters

Name	ID	Type	#	Notes
ExternalAccessoryProtocolIdentifier	0	uint8	1	See Table 52-10 (page 581)
ExternalAccessoryProtocolSessionIdentifier	1	uint16	1	

52.8.2 StopExternalAccessoryProtocolSession

Source	ID
Device	0xEA01

Table 52-51 `StopExternalAccessoryProtocolSession` message parameters

Name	ID	Type	#	Notes
ExternalAccessoryProtocolSessionIdentifier	0	uint16	1	

52.8.3 StatusExternalAccessoryProtocolSession

Source	ID
Accessory	0xEA03

Table 52-52 `StatusExternalAccessoryProtocolSession` message parameters

Name	ID	Type	#	Notes
ExternalAccessoryProtocolSessionIdentifier	0	uint16	1	
ExternalAccessoryProtocolSessionStatus	1	enum	1	See Table 52-53 (page 599).

Apple Confidential Property of Unigrand LTD
Unigrand@unigrand.com.tw

Table 52-53 `ExternalAccessoryProtocolSessionStatus` enum

Value	Meaning
0	SessionStatusOK
1	SessionClose

52.9 Human Interface Device

For more information, see HID (Human Interface Device) (page 407).

52.9.1 StartHID

Source	ID
Accessory	0x6800

Table 52-54 `StartHID` message parameters

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	Must refer to an identified Table 52-17 (page 584).
VendorIdentifier	1	uint16	1	Must be assigned and registered by the USB-IF for the accessory manufacturer.
ProductIdentifier	2	uint16	1	Must be unique for each accessory made by the accessory manufacturer.
LocalizedKeyboardCountryCode	3	uint8	0/1	Only required if the HID component is a localized (non-ANSI) keyboard. Must be drawn from the list of assigned country codes defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor*.
HIDReportDescriptor	4	blob	1	HID Report Descriptor.

Property of Unigrand Ltd.
taywei@unigrand.com.tw

52.9.2 DeviceHIDReport

Source	ID
Device	0x6801

Table 52-55 `DeviceHIDReport` message parameters

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	Must refer to an identified Table 52-17 (page 584)
HIDReport	1	blob	1	HID Report

52.9.3 AccessoryHIDReport

Source	ID
Accessory	0x6802

Table 52-56 `AccessoryHIDReport` message parameters

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	Must refer to an identified Table 52-17 (page 584)
HIDReport	1	blob	1	HID Report

52.9.4 StopHID

Source	ID
Accessory	0x6803

Table 52-57 `StopHID` message parameters

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	Must refer to an identified Table 52-17 (page 584)

Confidentiality Wu
tonywu@uniguardian.com.tw

52.10 Location

For more information, see Location Information (page 460).

52.10.1 StartLocationInformation

Source	ID
Device	0xFFFA

Table 52-58 `StartLocationInformation` message parameters

Name	ID	Type	#	Notes
GlobalPositioningSystemFixData	1	none	0/1	If present, the accessory may provide NMEA GPGGA sentences.
RecommendedMinimumSpecificGPSTransitData	2	none	0/1	If present, the accessory may provide NMEA GPRMC sentences.
GPSSatellitesInView	3	none	0/1	If present, the accessory may provide NMEA GPGSV sentences.
VehicleSpeedData	4	none	0/1	If present, the accessory may provide NMEA PASCD sentences.
VehicleGyroData	5	none	0/1	If present, the accessory may provide NMEA PAGCD sentences.
VehicleAccelerometerData	6	none	0/1	If present, the accessory may provide NMEA PAACD sentences.

Paul@Unigrand.com.tw Unigrand LTD

Name	ID	Type	#	Notes
VehicleHeadingData	7	none	0/1	If present, the accessory may provide NMEA GPHDT sentences.

52.10.2 LocationInformation

Source	ID
Accessory	0xFFFB

Table 52-59 `LocationInformation` message parameters

Name	ID	Type	#	Notes
NMEASentence	0	utf8	1+	One or more NMEA sentences of the type(s) specified by StartLocationInformation (page 601).

52.10.3 StopLocationInformation

Source	ID
Device	0xFFFC

Table 52-60 `StopLocationInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.11 Media Library Access

For more information, see Media Library Access (page 472).

Paywu@uniquind.com.tw Paywu Unique Ind LTD

52.11.1 StartMediaLibraryInformation

Source	ID
Accessory	0x4C00

Table 52-61 `StartMediaLibraryInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.11.2 MediaLibraryInformation

Source	ID
Device	0x4C01

Table 52-62 `MediaLibraryInformation` message parameters

Name	ID	Type	#	Notes
MediaLibraryInformation	0	group	0+	See Table 52-63 (page 603)

Table 52-63 `MediaLibraryInformation` parameter group

Name	ID	Type	#	Notes
MediaLibraryName	0	utf8	1	
MediaLibraryUniqueIdentifier	1	utf8	1	
MediaLibraryType	2	enum	1	See Table 52-64 (page 603)

Table 52-64 `MediaLibraryType` enum

Value	Meaning
0	Local device library
2	iTunes Radio library

Paraguard Engineering Inc. ID Paraguard.com.tw

52.11.3 StopMediaLibraryInformation

Source	ID
Accessory	0x4C02

Table 52-65 `StopMediaLibraryInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.11.4 StartMediaLibraryUpdates

Source	ID
Accessory	0x4C03

Table 52-66 `StartMediaLibraryUpdates` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1	
LastKnownMediaLibraryRevision	1	utf8	0/1	If no LastKnownMediaLibraryRevision is included, a full database update will be sent.
MediaItemProperties	2	group	0/1	See Table 52-67 (page 604)
MediaPlaylistProperties	3	group	0/1	See Table 52-68 (page 605)
MediaLibraryUpdateProgress	4	none	0/1	
MediaLibraryIsHidingRemoteItems	5	none	0/1	
PlayAllSongsCapable	6	none	0/1	

Table 52-67 `MediaItemProperties` parameter group

Name	ID	Type	#	Notes
MediaItemPropertyPersistentIdentifier	0	none	1	

Paryl Digital LLC
paryl@uniqland.com

Name	ID	Type	#	Notes
MediaItemPropertyTitle	1	none	0/1	
MediaItemPropertyMediaType	2	none	0/1	
MediaItemPropertyRating	3	none	0/1	
MediaItemPropertyPlaybackDurationInMilliseconds	4	none	0/1	
MediaItemPropertyAlbumPersistentIdentifer	5	none	0/1	
MediaItemPropertyAlbumTitle	6	none	0/1	
MediaItemPropertyAlbumTrackNumber	7	none	0/1	
MediaItemPropertyAlbumTrackCount	8	none	0/1	
MediaItemPropertyAlbumDiscNumber	9	none	0/1	
MediaItemPropertyAlbumDiscCount	10	none	0/1	
MediaItemPropertyArtistPersistentIdentifier	11	none	0/1	
MediaItemPropertyArtist	12	none	0/1	
MediaItemPropertyAlbumArtistPersistentIdentifier	13	none	0/1	
MediaItemPropertyAlbumArtist	14	none	0/1	
MediaItemPropertyGenrePersistentIdentifier	15	none	0/1	
MediaItemPropertyGenre	16	none	0/1	
MediaItemPropertyComposerPersistentIdentifier	17	none	0/1	
MediaItemPropertyComposer	18	none	0/1	
MediaItemPropertyIsPartOfCompilation	19	none	0/1	
MediaItemPropertyIsResidentOnDevice	25	none	0/1	
MediaItemPropertyChapterCount	27	none	0/1	

Table 52-68 `MediaPlaylistProperties` parameter group

Name	ID	Type	#	Notes
MediaPlaylistPropertyPersistentIdentifer	0	none	1	

Confidential Property of Apple Inc.
Do not distribute
Developer.apple.com.tw

Name	ID	Type	#	Notes
MediaPlaylistPropertyName	1	none	0/1	
MediaPlaylistPropertyParentPersistentIdentifer	2	none	0/1	
MediaPlaylistPropertyIsGeniusMix	3	none	0/1	
MediaPlaylistPropertyIsFolder	4	none	0/1	
MediaPlaylistPropertyContainedMediaItemsFileTransferIdentifier	5	none	0/1	
MediaPlaylistPropertyIsiTunesRadioStation	6	none	0/1	

52.11.5 MediaLibraryUpdate

Source	ID
Device	0x4C04

Table 52-69 `MediaLibraryUpdate` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1	
MediaLibraryRevision	1	utf8	0/1	
MediaItem	2	group	0+	See Table 52-70 (page 607). Every MediaItem in a MediaLibraryUpdate message will contain a MediaItemPersistentIdentifier parameter. Note that not all MediaItem parameters are supported by the Media Library feature. See Table 52-67 (page 604).
MediaPlaylist	3	group	0+	See Table 52-71 (page 609). Every MediaPlaylist in a MediaLibraryUpdate message will contain a MediaPlaylistPersistentIdentifer parameter.

Party WE LTD
Unigrand.com.tw

Name	ID	Type	#	Notes
MediaItemDeletePersistentIdentifier	4	uint64	0+	PersistentIdentifier of the media item that has been deleted.
MediaPlaylistDeletePersistentIdentifier	5	uint64	0+	PersistentIdentifier of the playlist that has been deleted.
MediaLibraryReset	6	none	0/1	If present, the accessory must delete all data pertaining to the specified media library before applying new updates.
MediaLibraryUpdateProgress	7	uint8	0/1	If present, indicates percentage completion for the current set of media library updates (0-100).
MediaLibraryIsHidingRemoteItems	8	bool	0/1	If present, the Apple device's UI is only presenting media items that are resident on the device to the user.
PlayAllSongsCapable	9	bool	0/1	If present, the Apple device is capable of playing all songs. See PlayMediaLibrarySpecial (page 612).

Table 52-70 `MediaItem` parameter group

Name	ID	Type	#	Notes
MediaItemPersistentIdentifier	0	uint64	0/1	
MediaItemTitle	1	utf8	0/1	
MediaItemMediaType	2	enum	0+	See Table 52-72 (page 609). Note that it is possible for some media items to be associated with multiple media types.

Parwuq@unigrand.com.tw Unigrand Parwuq ID

Name	ID	Type	#	Notes
MediaItemRating	3	uint8	0/1	Shall be 0...5
MediaItemPlaybackDurationInMilliseconds	4	uint32	0/1	
MediaItemAlbumPersistentIdentifer	5	uint64	0/1	
MediaItemAlbumTitle	6	utf8	0/1	
MediaItemAlbumTrackNumber	7	uint16	0/1	
MediaItemAlbumTrackCount	8	uint16	0/1	
MediaItemAlbumDiscNumber	9	uint16	0/1	
MediaItemAlbumDiscCount	10	uint16	0/1	
MediaItemArtistPersistentIdentifier	11	uint64	0/1	
MediaItemArtist	12	utf8	0/1	
MediaItemAlbumArtistPersistentIdentifier	13	uint64	0/1	
MediaItemAlbumArtist	14	utf8	0/1	
MediaItemGenrePersistentIdentifier	15	uint64	0/1	
MediaItemGenre	16	utf8	0/1	
MediaItemComposerPersistentIdentifier	17	uint64	0/1	
MediaItemComposer	18	utf8	0/1	
MediaItemIsPartOfCompilation	19	bool	0/1	
MediaItemIsLikeSupported	21	bool	0/1	
MediaItemIsBanSupported	22	bool	0/1	
MediaItemIsLiked	23	bool	0/1	
MediaItemIsBanned	24	bool	0/1	
MediaItemIsResidentOnDevice	25	bool	0/1	
MediaItemArtworkFileTransferIdentifier	26	uint8	0/1	
MediaItemChapterCount	27	uint16	0/1	

Apple Confidential
Do Not Distribute

Table 52-71 `MediaPlaylist` parameter group

Name	ID	Type	#	Notes
MediaPlaylistPersistentIdentifer	0	uint64	0/1	
MediaPlaylistName	1	utf8	0/1	
MediaPlaylistParentPersistentIdentifer	2	uint64	0/1	
MediaPlaylistIsGeniusMix	3	bool	0/1	
MediaPlaylistIsFolder	4	bool	0/1	
MediaPlaylistContainedMediaItemsFileTransferIdentifier	5	uint8	0/1	Note that it is entirely possible for the playlist content file transfer to be 0 length indicating a playlist with 0 contained items
MediaPlaylistIsiTunesRadioStation	6	bool	0/1	

Note: If the MediaPlaylist parameter group does not contain a MediaPlaylistContainedMediaItemsFileTransferIdentifier parameter, the playlist represents an endless playlist type such as Genius Mix or iTunes Radio station.

Table 52-72 `MediaType` enum

Value	Meaning
0	MediaTypeMusic
1	MediaTypePodcast
2	MediaTypeAudioBook
3	MediaTypeiTunesU

Paly Wu
paly.wu@uniqrand.com.tw
Uniqrand LTD

52.11.6 StopMediaLibraryUpdates

Source	ID
Accessory	0x4C05

Table 52-73 `StopMediaLibraryUpdate` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1+	

52.11.7 PlayMediaLibraryCurrentSelection

Source	ID
Accessory	0x4C06

Table 52-74 `PlayMediaLibraryCurrentSelection` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1	See MediaLibraryInformation (page 603)

52.11.8 PlayMediaLibraryItems

Source	ID
Accessory	0x4C07

Table 52-75 `PlayMediaLibraryItems` message parameters

Name	ID	Type	#	Notes
ItemsPersistentIdentifiers	0	blob	1	Array of ordered uint64 MediaItemPersistentIdentifiers (uint64[])
ItemsStartingIndex	1	uint32	0/1	The index of the first item in PersistentIdentifiers to start playback with. If this parameter is invalid or not present the starting index defaults to 0.

Paul Wu iPlaybox LTD
paul@iplaybox.com.tw

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	2	utf8	1	See MediaLibraryInformation (page 603)

52.11.9 PlayMediaLibraryCollection

Source	ID
Accessory	0x4C08

Table 52-76 `PlayMediaLibraryCollection` message parameters

Name	ID	Type	#	Notes
CollectionPersistentIdentifier	0	uint64	1	
CollectionType	1	enum	1	See Table 52-77 (page 611)
CollectionStartingIndex	2	uint32	0/1	The index of the first item in the collection to start playback with. If this parameter is invalid or not present the starting index defaults to 0.
MediaLibraryUniqueIdentifier	3	utf8	1	See MediaLibraryInformation (page 603)

Note: CollectionStartingIndex parameter must only be used for Playlist collection types. Behavior when used with other collection types is undefined.

Table 52-77 `MediaLibraryCollectionType` enum

Value	Meaning
0	Playlist
1	Artist
2	Album
3	Album Artist
4	Genre
5	Composer

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

52.11.10 PlayMediaLibrarySpecial

Source	ID
Accessory	0x4C09

Table 52-78 `PlayMediaLibrarySpecial` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1	
AllSongs	1	none	0/1	

52.12 Now Playing

For more information, see Now Playing Updates (page 478) feature.

52.12.1 StartNowPlayingUpdates

Source	ID
Accessory	0x5000

Table 52-79 `StartNowPlayingUpdates` message parameters

Name	ID	Type	#	Notes
MediaItemAttributes	0	group	0/1	See Table 52-80 (page 612)
PlaybackAttributes	1	group	0/1	See Table 52-81 (page 613)

Table 52-80 `StartNowPlayingMediaItemAttributes` parameter group

Name	ID	Type	#	Notes
MediaItemPersistentIdentifier	0	none	0/1	
MediaItemTitle	1	none	0/1	
MediaItemPlaybackDurationInMilliseconds	4	none	0/1	

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
MediaItemAlbumTitle	6	none	0/1	
MediaItemAlbumTrackNumber	7	none	0/1	
MediaItemAlbumTrackCount	8	none	0/1	
MediaItemAlbumDiscNumber	9	none	0/1	
MediaItemAlbumDiscCount	10	none	0/1	
MediaItemArtist	12	none	0/1	
MediaItemGenre	16	none	0/1	
MediaItemComposer	18	none	0/1	
MediaItemIsLikeSupported	21	none	0/1	
MediaItemIsBanSupported	22	none	0/1	
MediaItemIsLiked	23	none	0/1	
MediaItemIsBanned	24	none	0/1	
MediaItemArtworkFileTransferIdentifier	26	none	0/1	The device may not actually initiate the artwork file transfer if the now playing information changes, including the artwork, before the transfer has started.
MediaItemChapterCount	27	none	0/1	

Table 52-81 `StartNowPlayingPlaybackAttributes` parameter group

Name	ID	Type	#	Notes
PlaybackStatus	0	none	0/1	
PlaybackElapsedTimeInMilliseconds	1	none	0/1	
PlaybackQueueIndex	2	none	0/1	
PlaybackQueueCount	3	none	0/1	
PlaybackQueueChapterIndex	4	none	0/1	

Apple Confidential
Licensed to Unigrand.com
raw@unigrand.com

Name	ID	Type	#	Notes
PlaybackShuffleMode	5	none	0/1	
PlaybackRepeatMode	6	none	0/1	
PlaybackAppName	7	none	1	
PlaybackMediaLibraryUniqueIdentifier	8	none	0/1	
PBiTunesRadioAd	9	none	0/1	
PBiTunesRadioStationName	10	none	0/1	
PBiTunesRadioStationMediaPlaylistPersistentID	11	none	0/1	
PlaybackSpeed	12	none	0/1	
SetElapsedTimeAvailable	13	none	0/1	
PlaybackQueueListAvail	14	none	0/1	
PlaybackQueueListTransferID	15	none	0/1	
PlaybackAppBundleID	16	none	0/1	

52.12.2 NowPlayingUpdate

Source	ID
Device	0x5001

Table 52-82 `NowPlayingUpdate` message parameters

Name	ID	Type	#	Notes
MediaItemAttributes	0	group	0/1	See Table 52-70 (page 607). Note that not all MediaItem parameters are supported by the Now Playing feature. See Table 52-80 (page 612)
PlaybackAttributes	1	group	0/1	See Table 52-83 (page 615)

Patwu@unionpay.com Patwu@unionpay.com Patwu@unionpay.com

Table 52-83 `PlaybackAttributes` parameter group

Name	ID	Type	#	Notes
PlaybackStatus	0	enum	0/1	See Table 52-84 (page 616)
PlaybackElapsedTimeInMilliseconds	1	uint32	0/1	
PlaybackQueueIndex	2	uint32	0/1	
PlaybackQueueCount	3	uint32	0/1	
PlaybackQueueChapterIndex	4	uint32	0/1	
PlaybackShuffleMode	5	enum	0/1	See Table 52-85 (page 616)
PlaybackRepeatMode	6	enum	0/1	See Table 52-86 (page 616)
PlaybackAppName	7	utf8	0/1	The name of the Now Playing app
PBMediaLibraryUniqueIdentifier	8	utf8	0/1	
PBiTunesRadioAd	9	bool	0/1	Now playing an iTunes Radio ad
PBiTunesRadioStationName	10	utf8	0/1	Name of the now playing iTunes Radio station
PBiTunesRadioStationMediaPlaylistPersistentID	11	uint64	0/1	MediaPlaylistPersistentIdentifer for the now playing iTunes Radio station
PlaybackSpeed	12	uint16	0/1	Fixed decimal number with 2 digits after the decimal (i.e. 100 = 1.00x, 150 = 1.50x, 25 = 0.25x, etc.). A value of 0 when PlaybackStatus is not Stopped indicates Now Playing app is not providing a value.
SetElapsedTimeAvailable	13	bool	0/1	
PlaybackQueueListAvail	14	bool	0/1	Playback queue contents are available. If false, then playback queue must be cleared
PlaybackQueueListTransferID	15	uint8	0/1	If `PlaybackQueueListAvail`, the transfer ID for the contents of the playback queue

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
PlaybackAppBundleID	16	utf8	0/1	Uniform type identifer (UTI) in reverse-DNS format, e.g. `com.Ajax.hello`, of the Now Playing app

Table 52-84 `PlaybackStatus` enum

Value	Meaning
0	Stopped
1	Playing
2	Paused
3	SeekForward
4	SeekBackward

Table 52-85 `PlaybackShuffle` enum

Value	Meaning
0	Off
1	Songs
2	Albums

Table 52-86 `PlaybackRepeat` enum

Value	Meaning
0	Off
1	One
2	All

Pary Wu Unigrand LTD paryw@unigrand.com.tw

52.12.3 StopNowPlayingUpdates

Source	ID
Accessory	0x5002

Table 52-87 `StopNowPlayingUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.12.4 SetNowPlayingInformation

Source	ID
Accessory	0x5003

Table 52-88 `SetNowPlayingInformation` message parameters

Name	ID	Type	#	Notes
ElapsedTime	0	uint32	0/1	Only effective if `NowPlayingUpdate` message `SetElapsedTimeAvail` parameter is true.
PlaybackQueueIndex	1	uint32	0/1	

52.13 Power

For more information, see Power (page 480).

52.13.1 StartPowerUpdates

Source	ID
Accessory	0xAE00

Parkvui LTD
parkvui@planetland.com.tw

Table 52-89 `StartPowerUpdates` message parameters

Name	ID	Type	#	Notes
MaximumCurrentDrawnFromAccessory	0	none	0/1	
DeviceBatteryWillChargeIfPowerIsPresent	1	none	0/1	
AccessoryPowerMode	2	none	0/1	Must be present if and only if the accessory draws power from the Apple device
IsExternalChargerConnected	4	none	0/1	
BatteryChargingState	5	none	0/1	
BatteryChargeLevel	6	none	0/1	

52.13.2 PowerUpdate

Source	ID
Device	0xAE01

Table 52-90 `PowerUpdate` message parameters

Name	ID	Type	#	Notes
MaximumCurrentDrawnFromAccessory	0	uint16	0/1	The Apple device will draw up to this amount of current (in mA) from the accessory
DeviceBatteryWillChargeIfPowerIsPresent	1	bool	0/1	
AccessoryPowerMode	2	enum	0/1	See Table 52-91 (page 619)
IsExternalChargerConnected	4	bool	0/1	This parameter is sent when an external charger is connected or disconnected
BatteryChargingState	5	enum	0/1	This parameter provides the battery charging state. See Table 52-92 (page 619)
BatteryChargeLevel	6	uint16	0/1	This specifies the battery charge level (0-100)

Party Wei LID
tonywu@unigrand.com.tw

Table 52-91 `AccessoryPowerModes` enum

Value	Meaning
0	Reserved
1	Low Power Mode
2	Intermittent High Power Mode

Table 52-92 `BatteryChargingState` enum

Value	Meaning
0	Disabled
1	Charging
2	Charged

52.13.3 StopPowerUpdates

Source	ID
Accessory	0xAE02

Table 52-93 `StopPowerUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.13.4 PowerSourceUpdate

Source	ID
Accessory	0xAE03

Tay Wu @ Unigrand LTD
tay@unigrand.com.tw

Table 52-94 `PowerSourceUpdate` message parameters

Name	ID	Type	#	Notes
AvailableCurrentForDevice	0	uint16	0/1	Must be one of the following values - 0, 1000, 2100, or 2400
DeviceBatteryShouldChargeIfPowerIsPresent	1	bool	0/1	

52.14 USB Device Mode Audio

For more information, see Digital Audio (page 366).

52.14.1 StartUSBDeviceModeAudio

Source	ID
Accessory	0xDA00

Table 52-95 `StartUSBDeviceModeAudio` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.14.2 USBDeviceModeAudioInformation

Source	ID
Device	0xDA01

Table 52-96 `USBDeviceModeAudioInformation` message parameters

Name	ID	Type	#	Notes
SampleRate	0	enum	1	See Table 52-13 (page 582)

Pary Wu LTD
pary.wu@qinglanda.com.tw

52.14.3 StopUSBDeviceModeAudio

Source	ID
Accessory	0xDA02

Table 52-97 `StopUSBDeviceModeAudio` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.15 Telephony

For more information, see Telephony (page 512).

52.15.1 StartTelephonyCallStateInformation

Source	ID
Accessory	0x4154

Table 52-98 `StartTelephonyCallStateInformation` message parameters

Name	ID	Type	#	Notes
CallStatevCardFileTransferIdentifier	5	none	0/1	

52.15.2 TelephonyCallStateInformation

Source	ID
Device	0x4155

Table 52-99 `TelephonyCallStateInformation` message parameters

Name	ID	Type	#	Notes
CallStatePhoneNumber	0	utf8	1	Calling party's number

Party Wu
Copyright © 2014 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
CallStateNetworkSentCallerName	1	utf8	1	Calling party's name sent by network
CallStateStatus	2	enum	1	see Table 52-100 (page 622)
CallStateDirection	3	enum	1	see Table 52-101 (page 622)
UniqueCallID	4	utf8	1	Differentiates between cellular and data calls from same contact as well as conference calls
CallStatevCardFileTransferIdentifier	5	uint8	0/1	

Table 52-100 `CallStateStatusValue` enum

Value	Meaning
0	Disconnected
1	Active
2	Holding (waiting)
3	Connecting

Table 52-101 `CallStateDirectionValue` enum

Value	Meaning
0	Incoming
1	Outgoing
2	Unknown

52.15.3 StopTelephonyCallStateInformation

Source	ID
Accessory	0x4156

Pary Wu
Unigrand LTD
pary.wu@unigrand.com.tw

Table 52-102 `StopTelephonyCallStateInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.15.4 StartTelephonyUpdates

Source	ID
Accessory	0x4157

Table 52-103 `StartTelephonyUpdate` message parameters

Name	ID	Type	#	Notes
TelephonySignalStrength	0	none	0/1	
TelephonyRegistrationStatus	1	none	0/1	
TelephonyAirplaneModeStatus	2	none	0/1	
TelephonyTTYStatus	3	none	0/1	
MobileOperator	4	none	0/1	

52.15.5 TelephonyUpdate

Source	ID
Device	0x4158

Table 52-104 `TelephonyUpdate` message parameters

Name	ID	Type	#	Notes
TelephonySignalStrength	0	enum	0/1	see Table 52-105 (page 624)
TelephonyRegistrationStatus	1	enum	0/1	see Table 52-106 (page 624)
TelephonyAirplaneModeStatus	2	bool	0/1	

Party Wu Unicorn LTD
Party@unicornland.com.tw

Name	ID	Type	#	Notes
TelephonyTTYStatus	3	bool	0/1	
MobileOperator	4	utf8	0/1	

Table 52-105 `TelephonySignalStrengthValue` enum

Value	Meaning
0	0 bars (no service)
1	1 bar
2	2 bars
3	3 bars
4	4 bars
5	5 bars

Table 52-106 `TelephonyRegistrationStatusValue` enum

Value	Meaning
0	Unknown
1	Not Registered
2	Searching
3	Denied
4	Registered Home
5	Roaming
6	Emergency Calls Only

Pary Wu
Uniqrand LTD
pary.wu@uniqrand.com.tw

52.15.6 StopTelephonyUpdates

Source	ID
Accessory	0x4159

Table 52-107 `StopTelephonyUpdate` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.16 Vehicle Status

For more information, see Vehicle Status (page 532).

52.16.1 StartVehicleStatusUpdates

Source	ID
Device	0xA100

Table 52-108 `StartVehicleStatusUpdates` message parameters

Name	ID	Type	#	Notes
Range	3	none	0/1	
OutsideTemperature	4	none	0/1	
RangeWarning	6	none	0/1	

52.16.2 VehicleStatusUpdate

Source	ID
Accessory	0xA101

Party Wu
Unigrand LTD
Unigrand.com.tw

Table 52-109 `VehicleStatusUpdate` message parameters

Name	ID	Type	#	Notes
Range	3	uint16	0/1	Remaining vehicle range in kilometers
OutsideTemperature	4	int16	0/1	measured outside temperature in degrees Celsius
RangeWarning	6	bool	0/1	if True, the vehicle's low range warning indicator is set

52.16.3 StopVehicleStatusUpdates

Source	ID
Device	0xA102

Table 52-110 `StopVehicleStatusUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.17 VoiceOver

For more information, see VoiceOver (page 534).

52.17.1 StartVoiceOver

Source	ID
Accessory	0x5612

Table 52-111 `StartVoiceOver` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

Party Wu Unigrand LTD
partywu@unigrand.com.tw

52.17.2 StopVoiceOver

Source	ID
Accessory	0x5613

Table 52-112 `StopVoiceOver` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.17.3 RequestVoiceOverMoveCursor

Source	ID
Accessory	0x5601

Table 52-113 `RequestVoiceOverMoveCursor` message parameters

Name	ID	Type	#	Notes
CursorDirection	0	enum	1	See Table 52-114 (page 627)

Table 52-114 `VoiceOverCursorDirection` enum

Value	Meaning
0	Next
1	Previous
2	Escape

52.17.4 RequestVoiceOverActivateCursor

Source	ID
Accessory	0x5602

Parvinder Uniguard LTD parvinder@uniguard.com.tw

Table 52-115 `RequestVoiceOverActivateCursor` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.17.5 RequestVoiceOverScrollPage

Source	ID
Accessory	0x5603

Table 52-116 `RequestVoiceOverScrollPage` message parameters

Name	ID	Type	#	Notes
ScrollDirection	0	enum	1	See Table 52-117 (page 628)

Table 52-117 `VoiceOverScrollDirection` enum

Value	Meaning
0	Left
1	Right
2	Up
3	Down

52.17.6 RequestVoiceOverSpeakText

Source	ID
Accessory	0x5606

Payment@Unigrand.com.tw Unigrand LTD Payment Wu

Table 52-118 `RequestVoiceOverSpeakText` message parameters

Name	ID	Type	#	Notes
TextToSpeak	0	utf8	0/1	If not present, VoiceOver will read the currently visible onscreen text starting from the top. Otherwise, VoiceOver will read the text present in this parameter

52.17.7 RequestVoiceOverPauseText

Source	ID
Accessory	0x5608

Table 52-119 `RequestVoiceOverPauseText` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.17.8 RequestVoiceOverResumeText

Source	ID
Accessory	0x5609

Table 52-120 `RequestVoiceOverResumeText` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.17.9 StartVoiceOverUpdates

Source	ID
Accessory	0x560B

Pary Wu
pary.wu@uniquand.com.tw
Uniquand LTD

Table 52-121 `StartVoiceOverUpdates` message parameters

Name	ID	Type	#	Notes
SpeakingVolume	0	none	0/1	
SpeakingRate	1	none	0/1	
Enabled	2	none	0/1	

52.17.10 VoiceOverUpdate

Source	ID
Device	0x560C

Table 52-122 `VoiceOverUpdate` message parameters

Name	ID	Type	#	Notes
SpeakingVolume	0	uint8	0/1	0=muted, 255=loudest possible volume
SpeakingRate	1	uint8	0/1	0=off, 255=fastest possible speed
Enabled	2	bool	0/1	

52.17.11 StopVoiceOverUpdates

Source	ID
Accessory	0x560D

Table 52-123 `StopVoiceOverUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

Party Wu Unigrand LTD
paulwu@unigrand.com.tw

52.17.12 RequestVoiceOverConfiguration

Source	ID
Accessory	0x560E

Table 52-124 `RequestVoiceOverConfiguration` message parameters

Name	ID	Type	#	Notes
SpeakingVolume	0	uint8	0/1	0=muted, 255=loudest possible volume
SpeakingRate	1	uint8	0/1	0=off, 255=fastest possible speed

52.17.13 StartVoiceOverCursorUpdates

Source	ID
Accessory	0x560F

Table 52-125 `StartVoiceOverCursorUpdates` message parameters

Name	ID	Type	#	Notes
Label	0	none	0/1	
Value	1	none	0/1	
Hint	2	none	0/1	
Traits	3	none	0/1	

52.17.14 VoiceOverCursorUpdate

Source	ID
Device	0x5610

Third Party Wu
Uniqrand LTD
jay.wu@uniqrand.com.tw

Table 52-126 `VoiceOverCursorUpdate` message parameters

Name	ID	Type	#	Notes
Label	0	utf8	0/1	
Value	1	utf8	0/1	
Hint	2	utf8	0/1	
Traits	3	blob	0/1	An array of uint16s. See Table 52-127 (page 632) (uint16[])

The following VoiceOver item traits are defined:

Table 52-127 VoiceOverCursorUpdate Traits

Value	Meaning
0	Button
1	Link
2	Search Field
3	Image
4	Selected
5	Sound
6	Keyboard Key
7	Static Text
8	Summary Element
9	Not Enabled
10	Updates Frequently
11	Starts Media Session
12	Adjustable
13	Back Button
14	Map

Pary Wu Unigrand LTD paywu@unigrand.com.tw

Value	Meaning
15	Delete Key

52.17.15 StopVoiceOverCursorUpdates

Source	ID
Accessory	0x5611

Table 52-128 `StopVoiceOverCursorUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.18 Wi-Fi Information Sharing

For more information, see Wi-Fi Information Sharing (page 536).

52.18.1 RequestWiFiInformation

Source	ID
Accessory	0x5700

Table 52-129 `RequestWiFiInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

52.18.2 WiFiInformation

Source	ID
Device	0x5701

Party Wu Unigrand LTD
tawu@unigrand.com.tw

Table 52-130 `WiFiInformation` message parameters

Name	ID	Type	#	Notes
RequestStatus	0	enum	1	see Table 52-131 (page 634)
WiFiSSID	2	utf8	0/1	
WiFiPassphrase	3	utf8	0/1	

Table 52-131 `WiFiRequestStatus` enum

Value	Meaning
0	Success
1	User Declined
2	Network Information Unavailable

Pary Wu
Pary Wu@unigrand LTD
Pary Wu@unigrand.com.tw

53. Accessory Test System

Some of the test procedures in this specification require use of the Accessory Test System (ATS). ATS is used during accessory development and self-certification to verify iAP communication, and check signals on a Lightning or 30-pin connector, depending on the system.

The ATS is available in whole or in part via several configurations:

ATS Configuration	Primary Usage
ATS Lightning Box	The ATS (Accessory Test System) Lightning box is used during accessory development & self-certification to verify iAP communication, and check signals on the Lightning connector. ATS Bundle # 1 is required, but not included, in order to capture iAP over USB traffic.
ATS 30-pin	The ATS (Accessory Test System) 30-pin is used during accessory development & self-certification to verify iAP communication, and check signals on the 30-pin connector. ATS Bundle # 1 is required, but not included, in order to capture iAP over USB traffic.
ATS Bundle # 1	This bundle includes the TotalPhase Beagle USB 480 Protocol Analyzer (TP320510), a USB Breakout Board, and the Belkin USB Plus hub. ATS Bundle # 1 is required to capture iAP over USB traffic using an ATS 30-pin or ATS Lightning Box The ATS 30-pin or ATS Lightning Box is required, but not included.
ATS Bundle # 2	This bundle includes the USB Breakout Board, and the Belkin USB Plus hub. The ATS 30-pin or ATS Lightning Box is also required, but not included.

Partwell@ettlabtech.com.tw

54. Device Dimensional Drawings

This chapter contains the following dimensional drawings:

- iPhone 6 (page 638)
- iPhone 6 Plus (page 639)
- iPhone 5s (page 640)
- iPhone 5c (page 641)
- iPhone 5 (page 642)
- iPhone 4s (page 643)
- iPhone 4 (CDMA model) (page 644)
- iPhone 4 (GSM model) (page 645)
- iPhone 3G and iPhone 3GS (page 646)
- iPhone (page 647)
- iPad Air 2 Wi-Fi (page 648)
- iPad Air 2 Wi-Fi + Cellular (page 649)
- iPad mini 2 & 3 Wi-Fi (page 650)
- iPad mini 2 & 3 Wi-Fi + Cellular (page 651)
- iPad Air Wi-Fi (page 652)
- iPad Air Wi-Fi + Cellular (page 653)
- iPad mini with Wi-Fi (page 654)
- iPad mini with Wi-Fi + Cellular (page 655)
- iPad with Wi-Fi (4th generation) (page 656)
- iPad with Wi-Fi + Cellular (4th generation) (page 657)
- iPad with Wi-Fi (3rd generation) (page 658)
- iPad with Wi-Fi + 4G (3rd generation) (page 659)
- iPad 2 with Wi-Fi (page 660)
- iPad 2 with Wi-Fi + 3G (page 661)
- iPad with Wi-Fi (page 662)

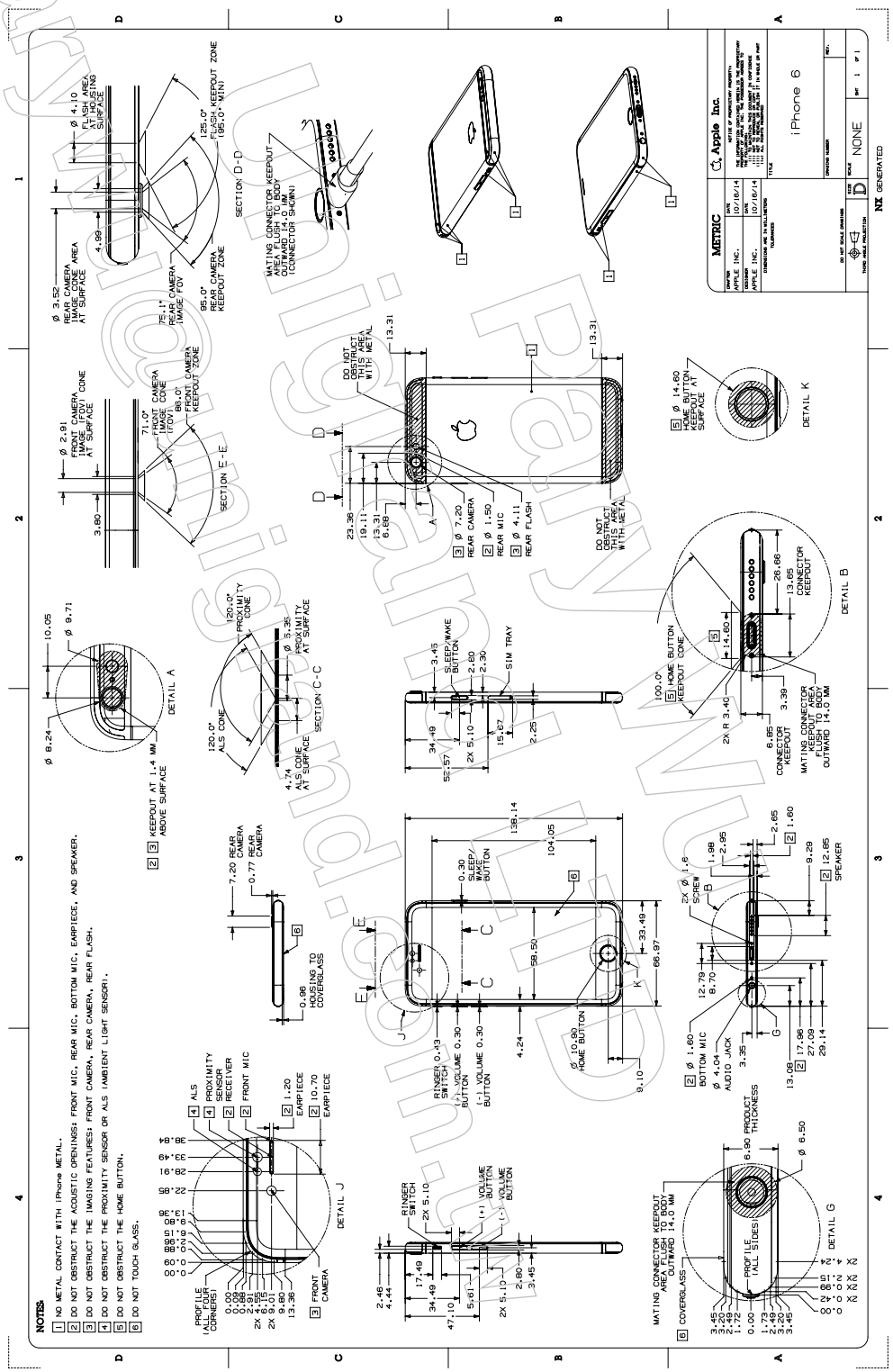
Pary Wu Unigrand LTD
pary@unigrand.com.tw

- iPad with Wi-Fi + 3G (page 663)
- iPod touch (5th generation) (page 664)
- iPod touch (4th generation) (page 665)
- iPod touch (3rd generation) (page 666)
- iPod touch (2nd generation) (page 667)
- iPod touch (page 668)
- iPod nano (7th generation) (page 669)
- iPod nano (6th generation) (page 670)
- iPod nano (5th generation) (page 671)
- iPod nano (4th generation) (page 672)
- iPod nano (3rd generation) (page 673)
- iPod nano (2nd generation) (page 674)
- iPod nano (page 675)
- iPod classic 160GB (page 676)
- iPod classic 80GB (page 677)
- iPod (5th generation) 60GB/80GB (page 678)
- iPod (5th generation) 30GB (page 679)
- iPod (4th generation) (page 680)
- iPod (3rd generation) (page 681)
- iPod photo 30GB/60GB (page 682)
- iPod photo (page 683)
- iPod shuffle (4th generation) (page 684)
- iPod shuffle (3rd generation) (page 685)
- iPod shuffle (2nd generation) (page 686)
- iPod shuffle (page 687)
- iPod mini (page 689)

Party Wu Unionland LTD
partywu@unionland.com.tw

54.1 iPhone 6

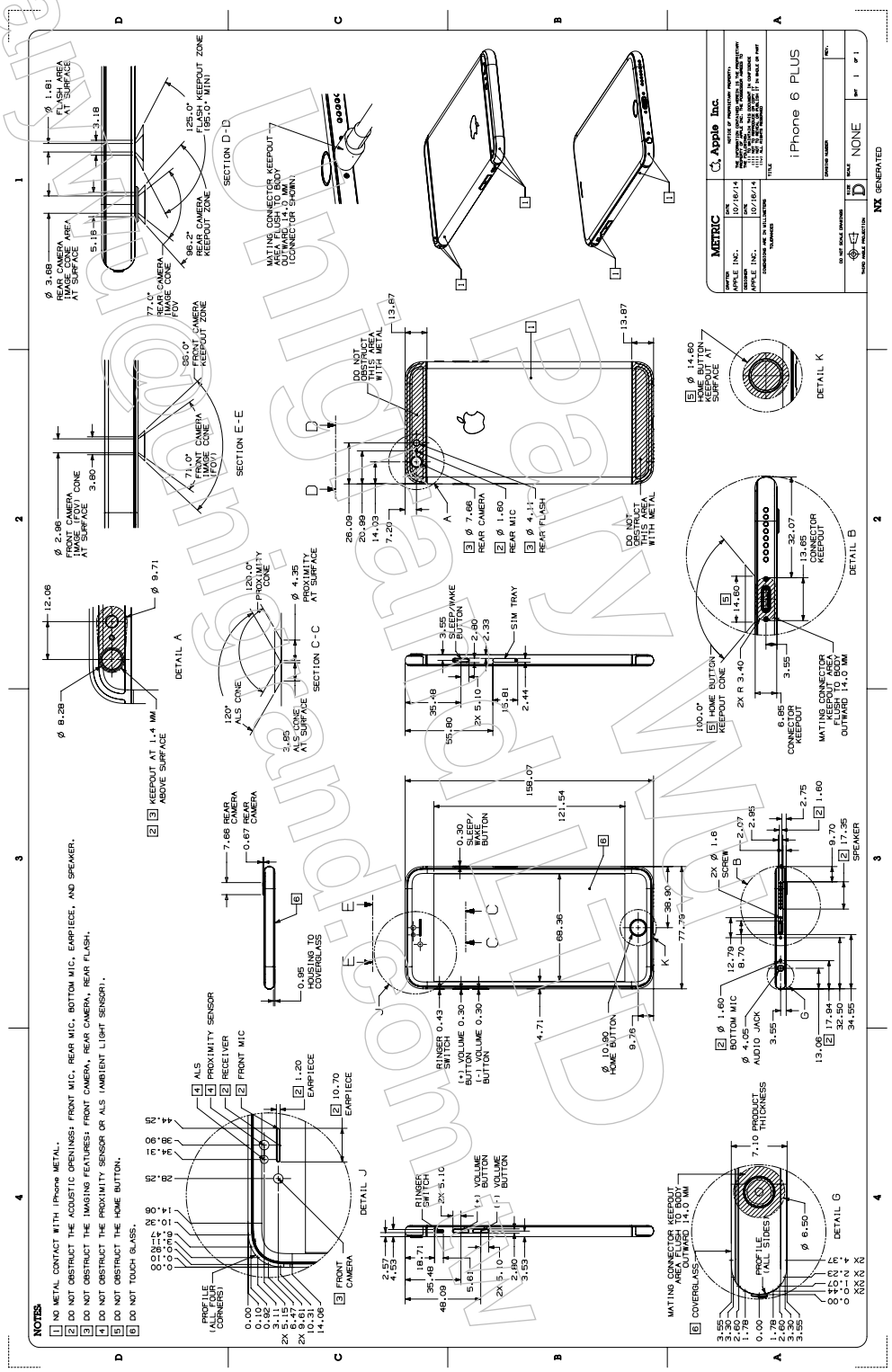
Figure 54-1 iPhone 6 Dimensional Drawing



Apple Confidential Partner Version

54.2 iPhone 6 Plus

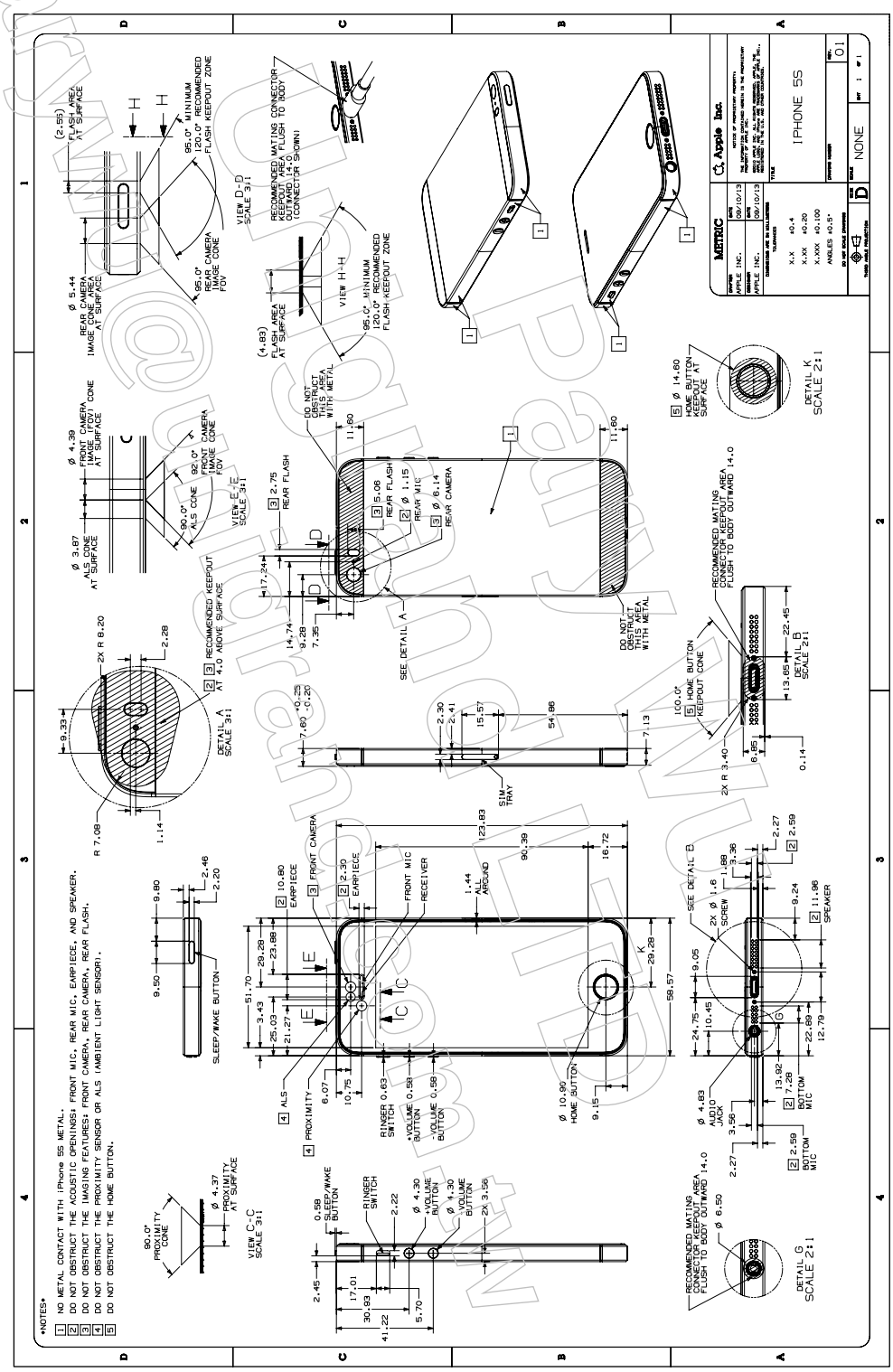
Figure 54-2 iPhone 6 Plus Dimensional Drawing



Apple Confidential
Internal Only

54.3 iPhone 5s

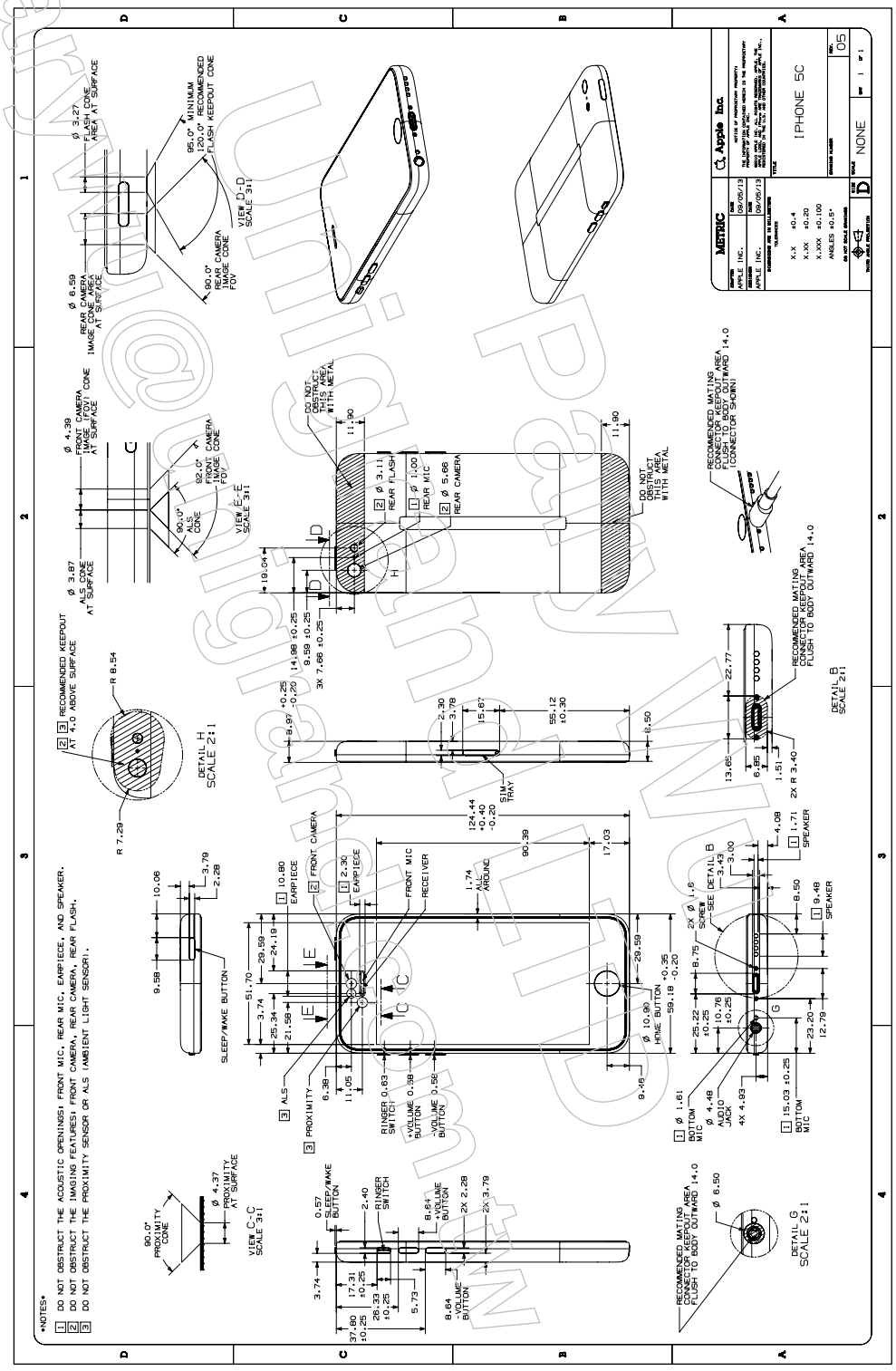
Figure 54-3 iPhone 5s Dimensional Drawing



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

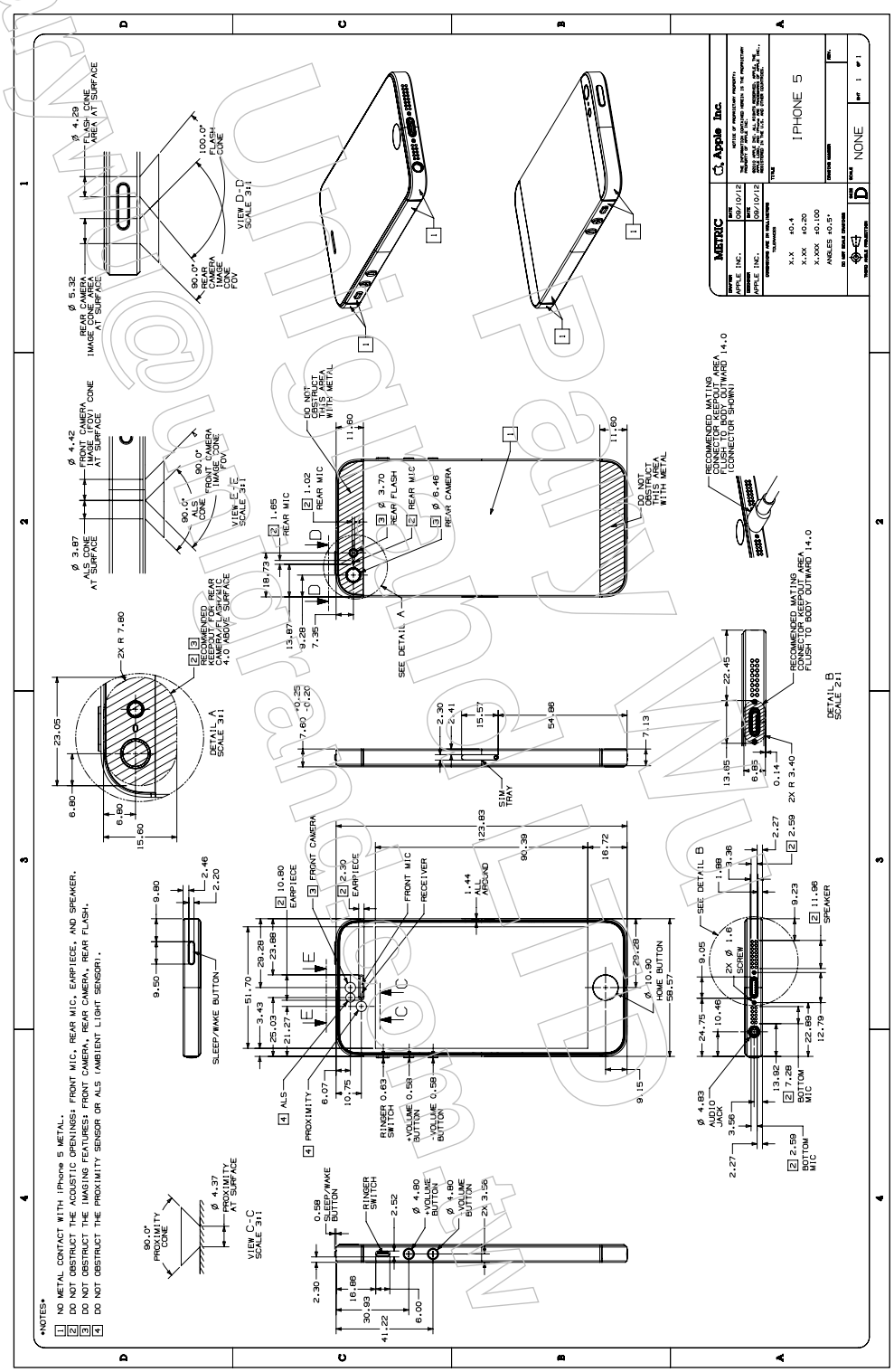
54.4 iPhone 5c

Figure 54-4 iPhone 5c Dimensional Drawing



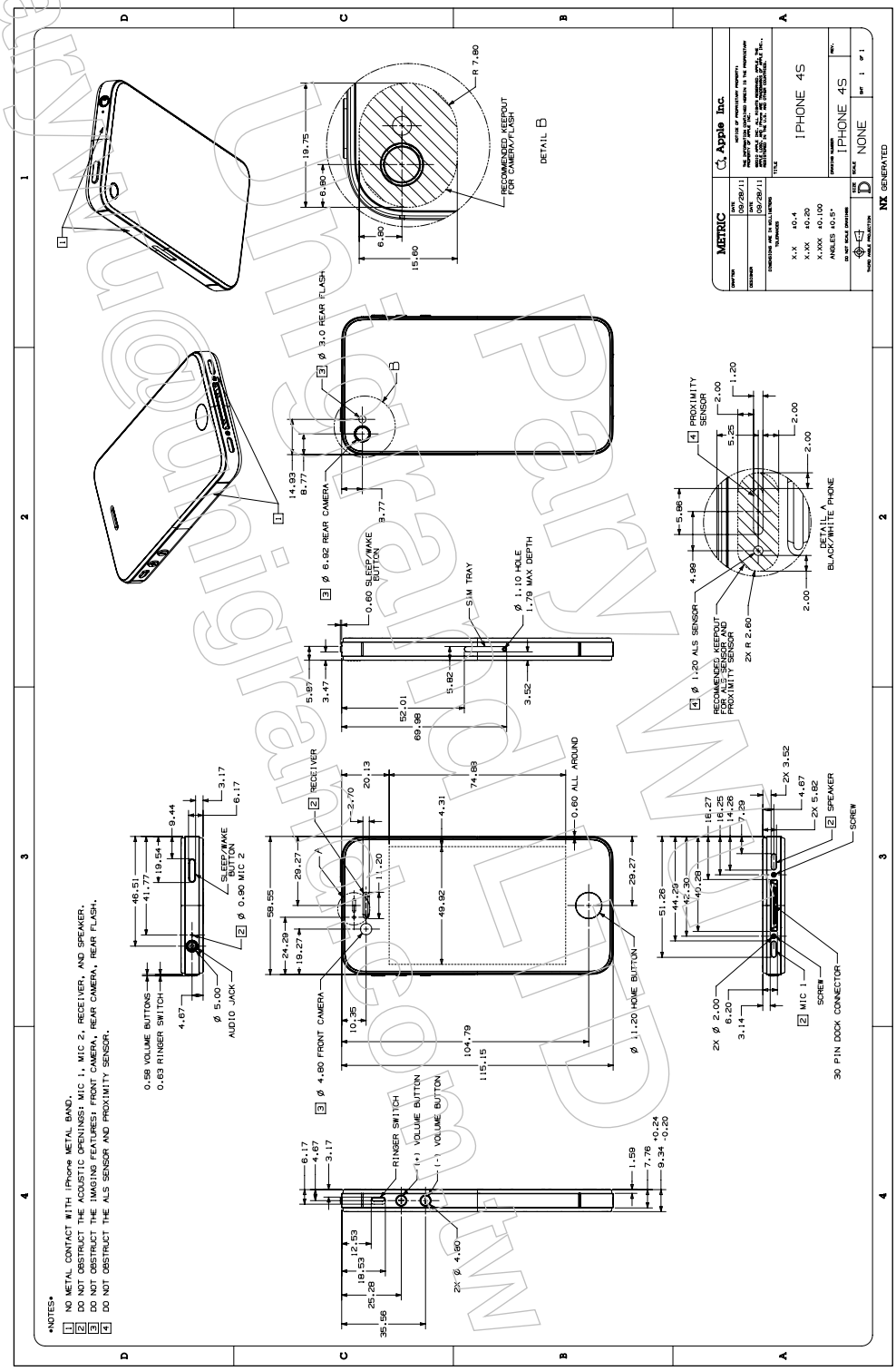
54.5 iPhone 5

Figure 54-5 iPhone 5 Dimensional Drawing



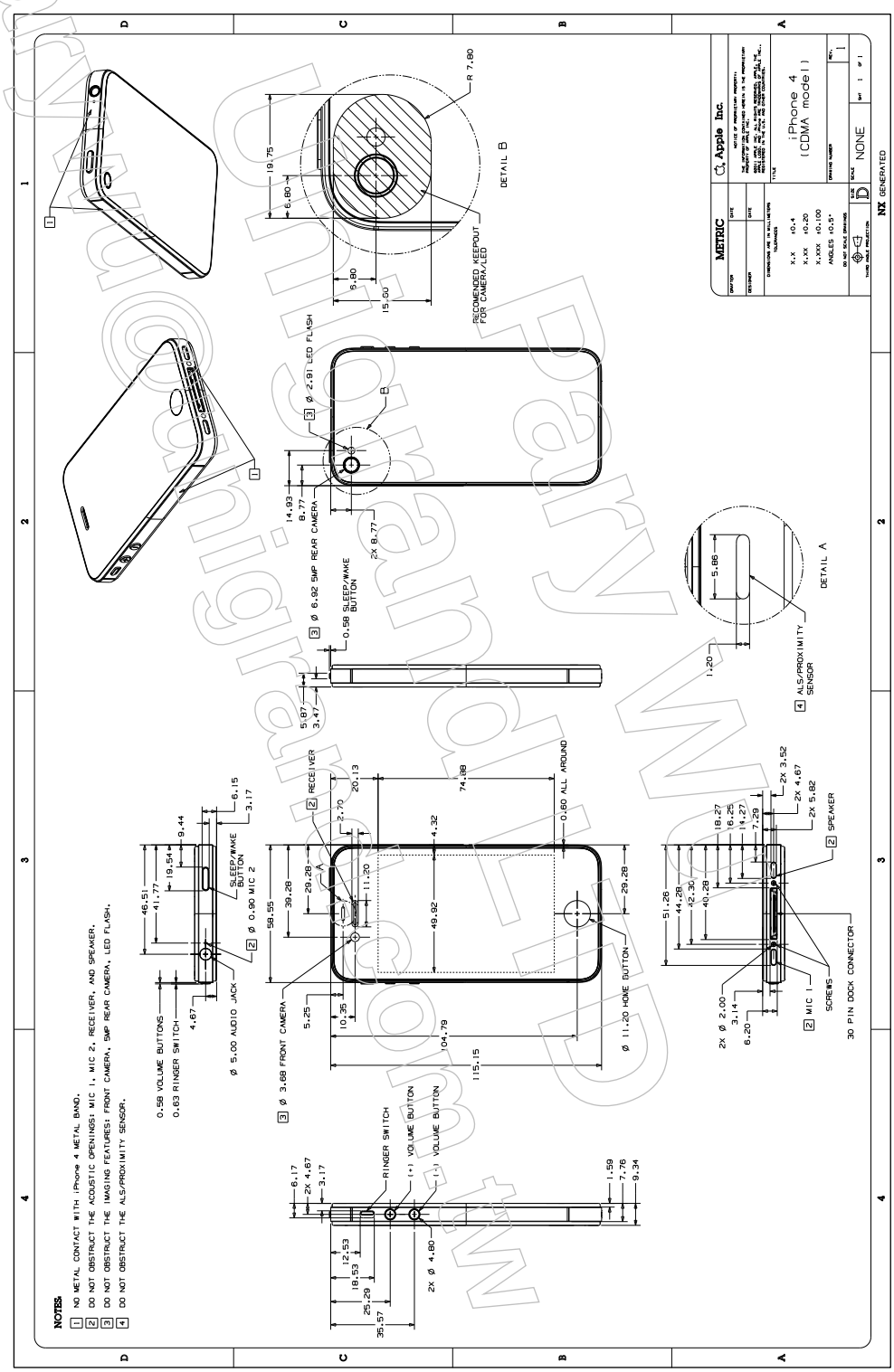
54.6 iPhone 4s

Figure 54-6 iPhone 4s Dimensional Drawing



54.7 iPhone 4 (CDMA model)

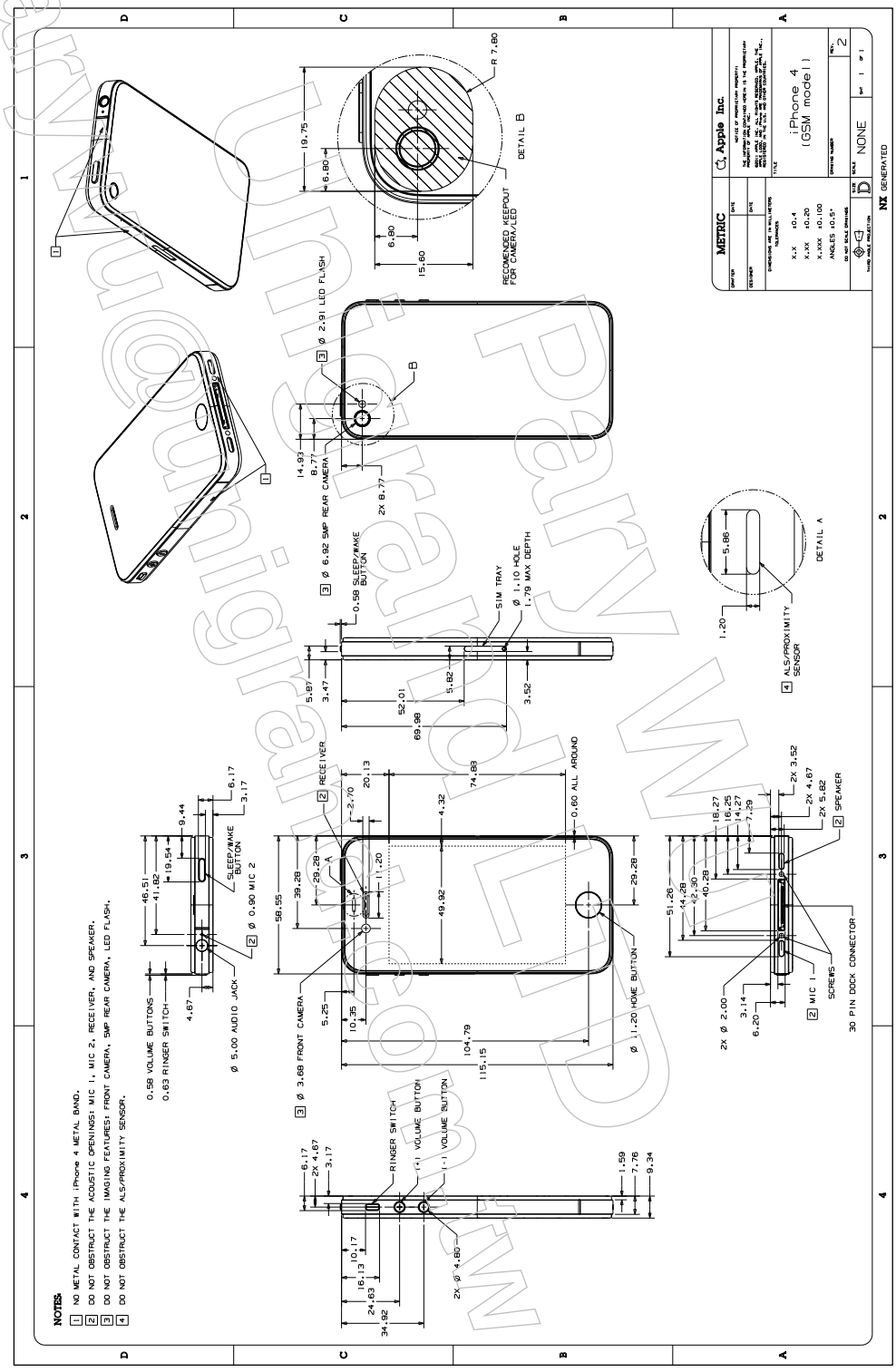
Figure 54-7 iPhone 4 CDMA Dimensional Drawing



Apple Confidential Property
Do not distribute

54.8 iPhone 4 (GSM model)

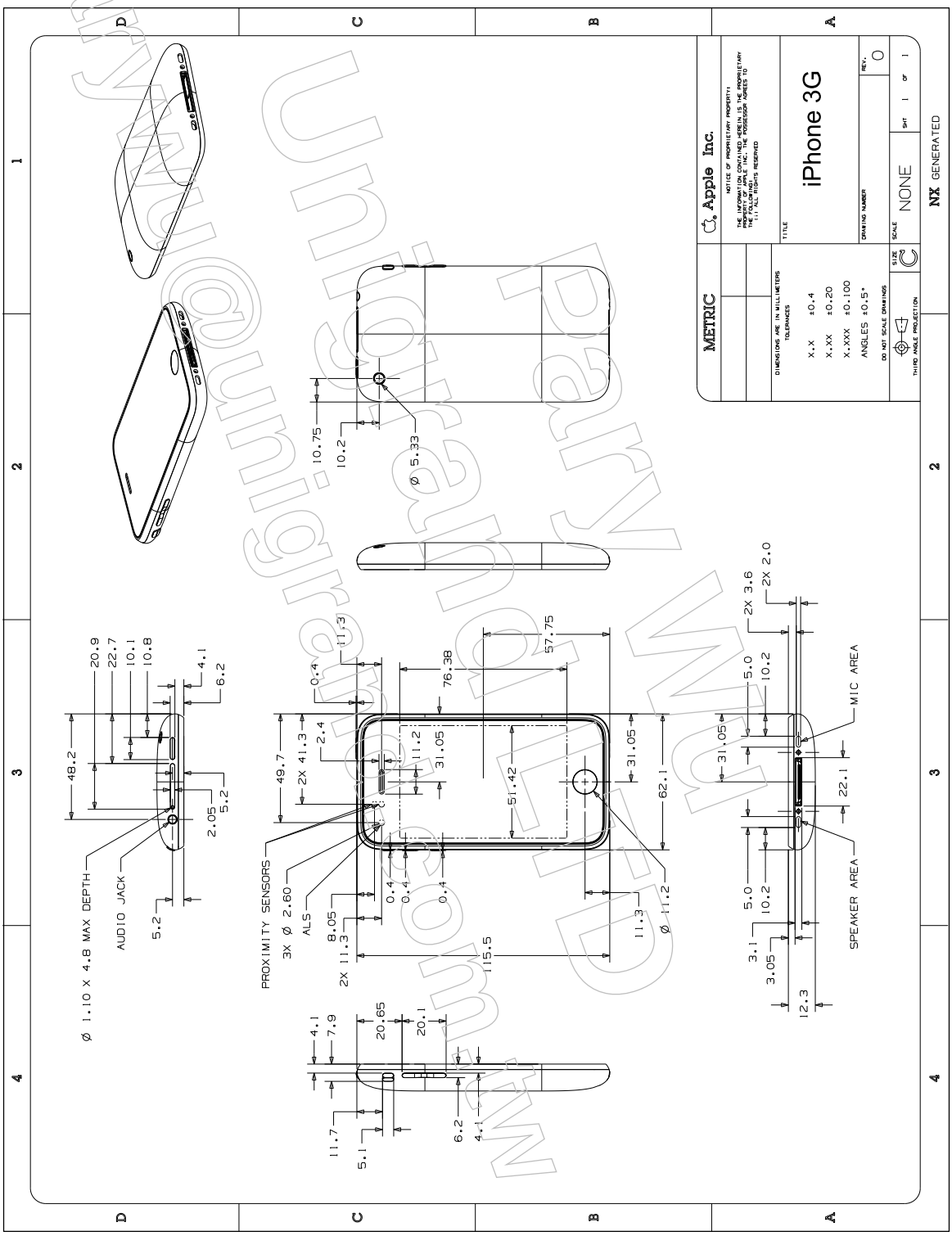
Figure 54-8 iPhone 4 GSM Dimensional Drawing



Part Number Pending

54.9 iPhone 3G and iPhone 3GS

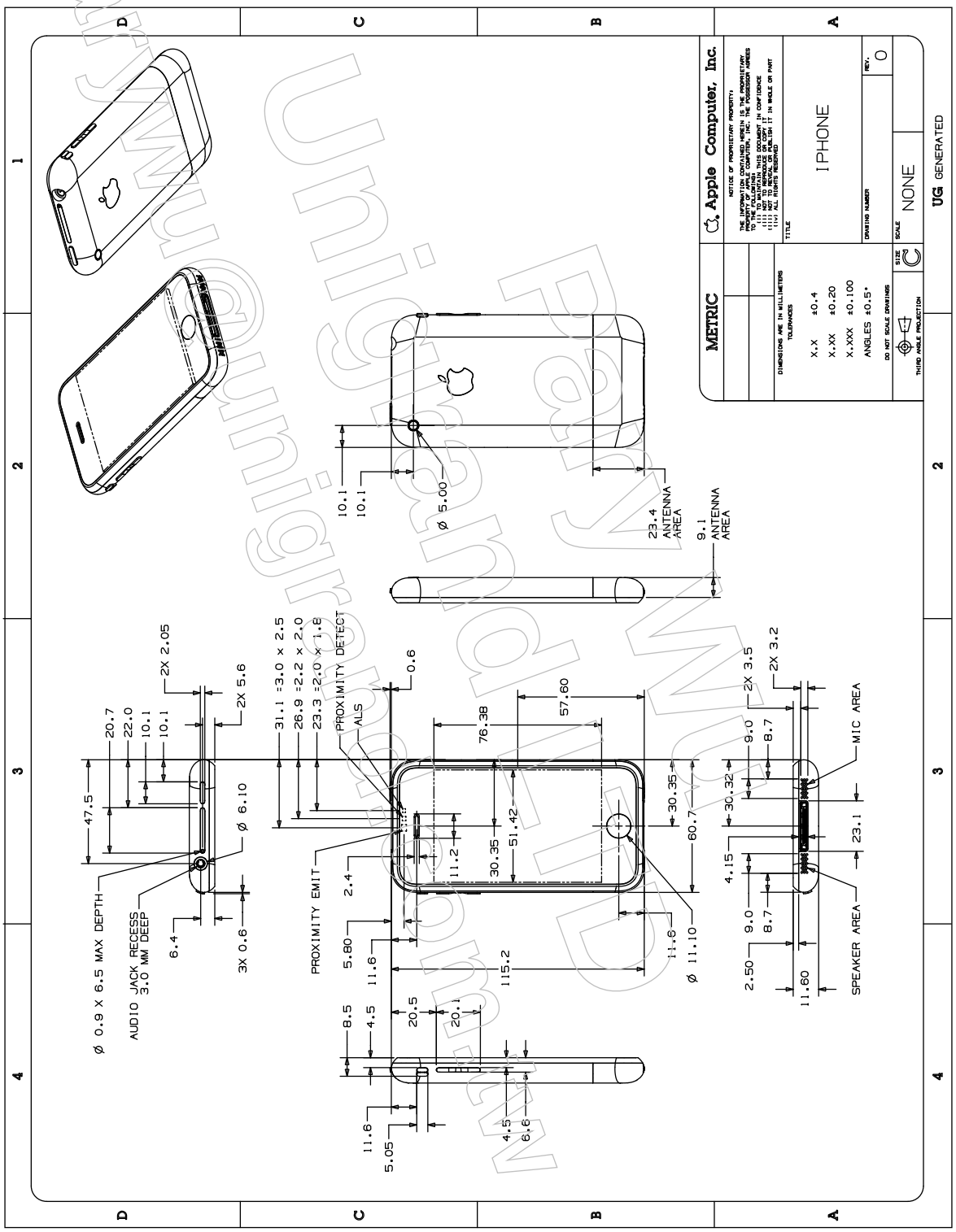
Figure 54-9 iPhone 3G and iPhone 3GS Dimensional Drawing



Unofficial Document

54.10 iPhone

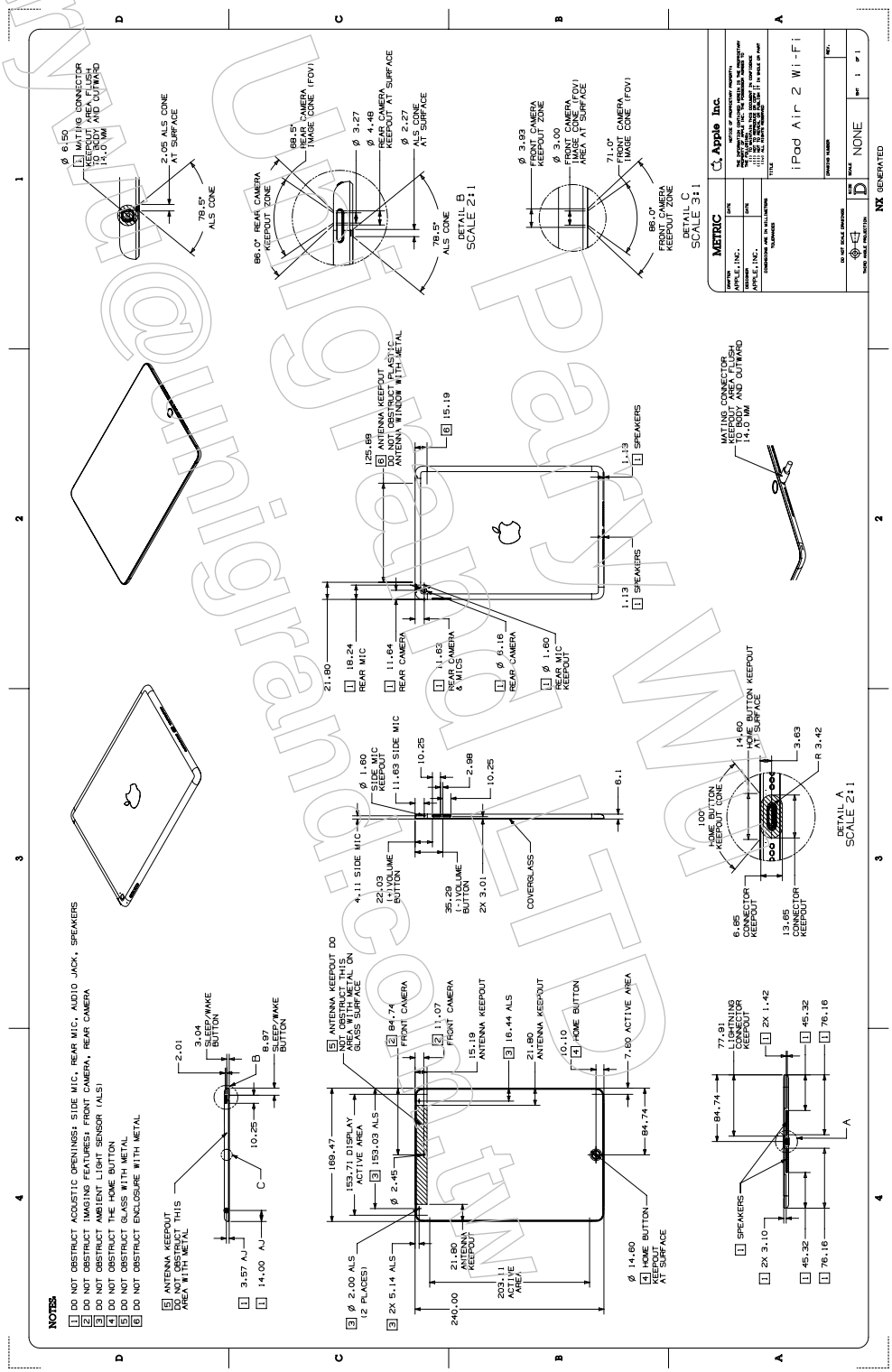
Figure 54-10 iPhone Dimensional Drawing



Paywordunigroup.com

54.11 iPad Air 2 Wi-Fi

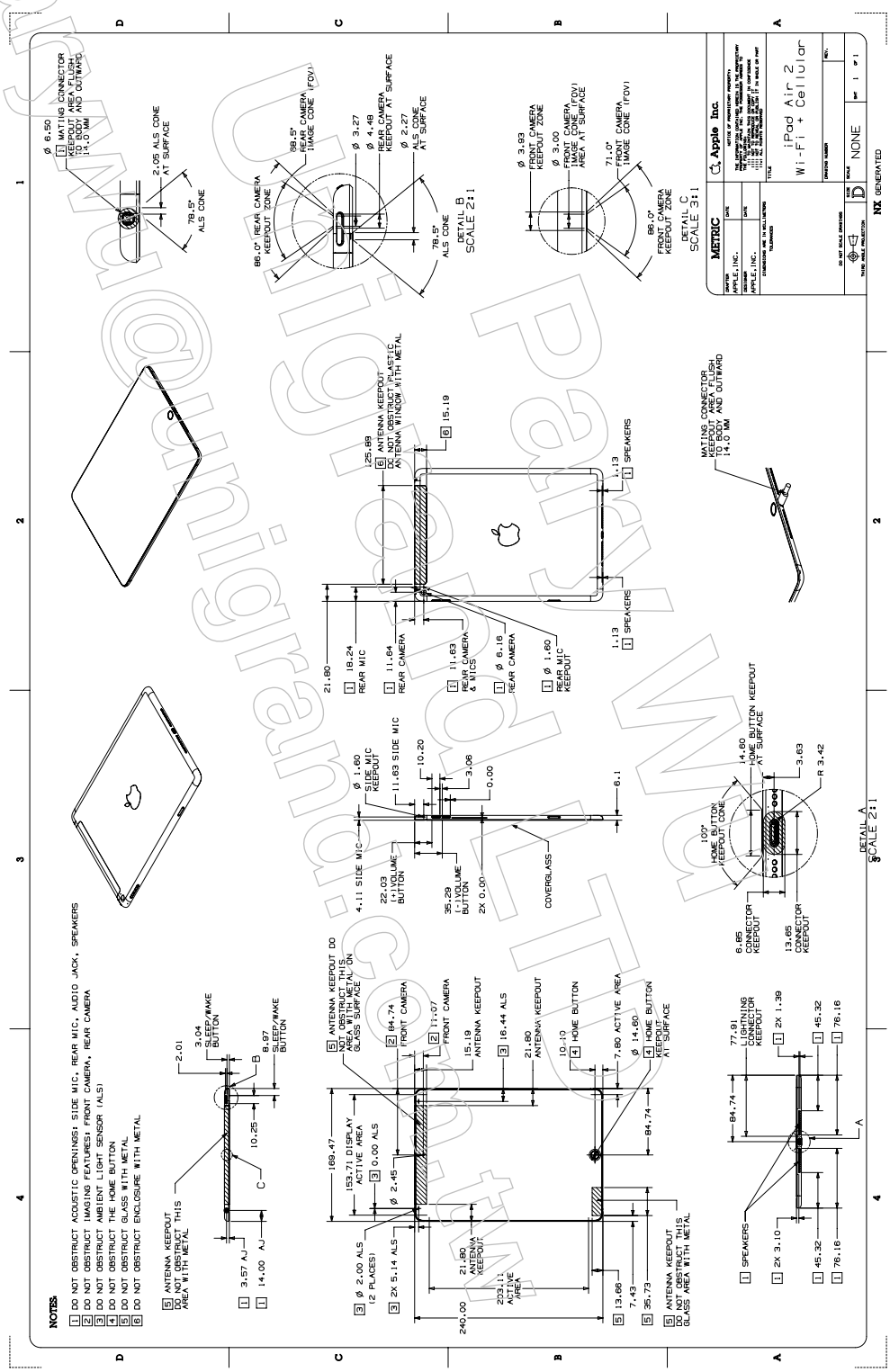
Figure 54-11 iPad Air 2 Wi-Fi Dimensional Drawing



Uncontrolled Copy
Apple Confidential

54.12 iPad Air 2 Wi-Fi + Cellular

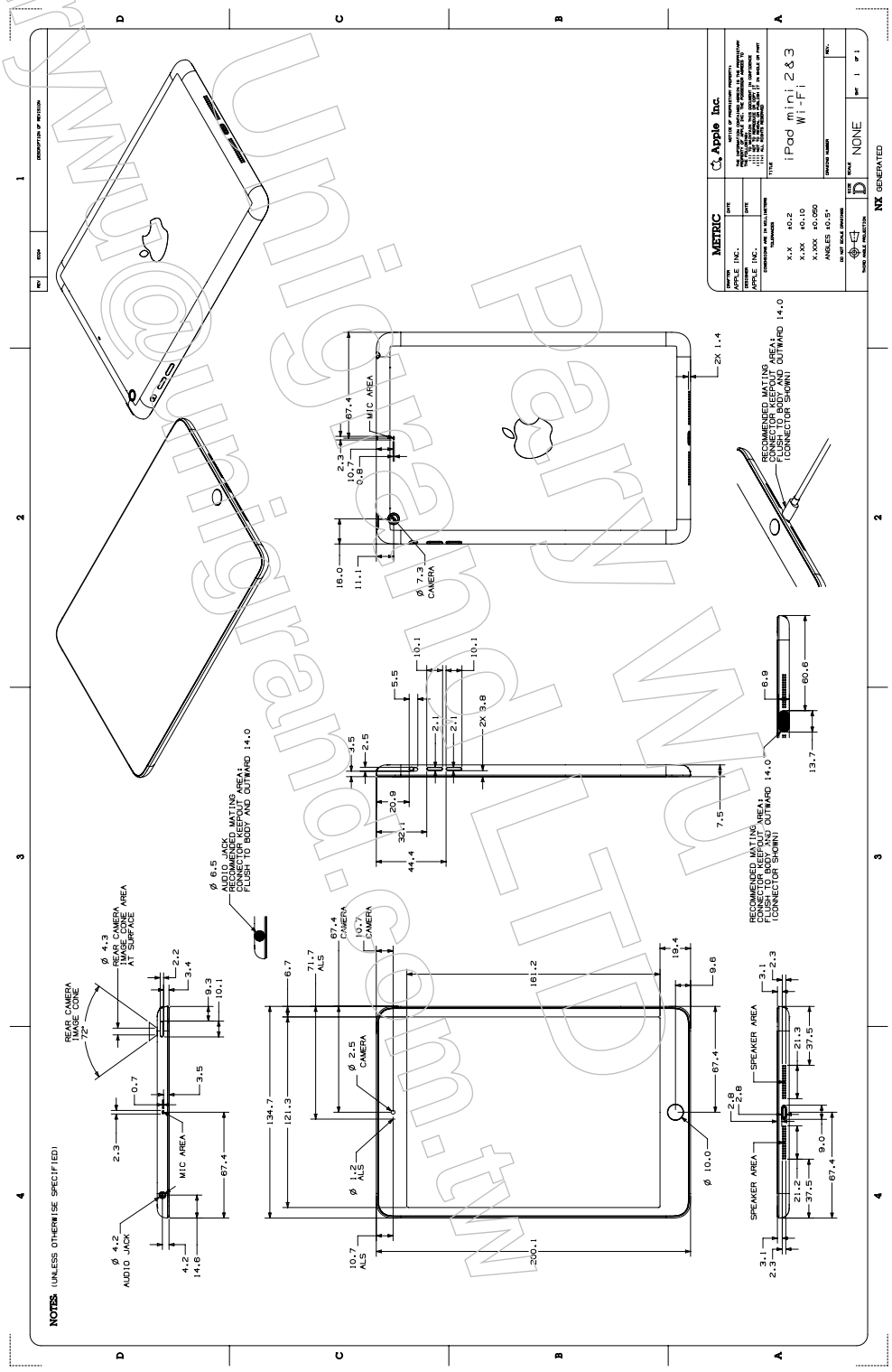
Figure 54-12 iPad Air 2 Wi-Fi + Cellular Dimensional Drawing



Uncontrolled Copy
Apple Confidential
Paul Meyer

54.13 iPad mini 2 & 3 Wi-Fi

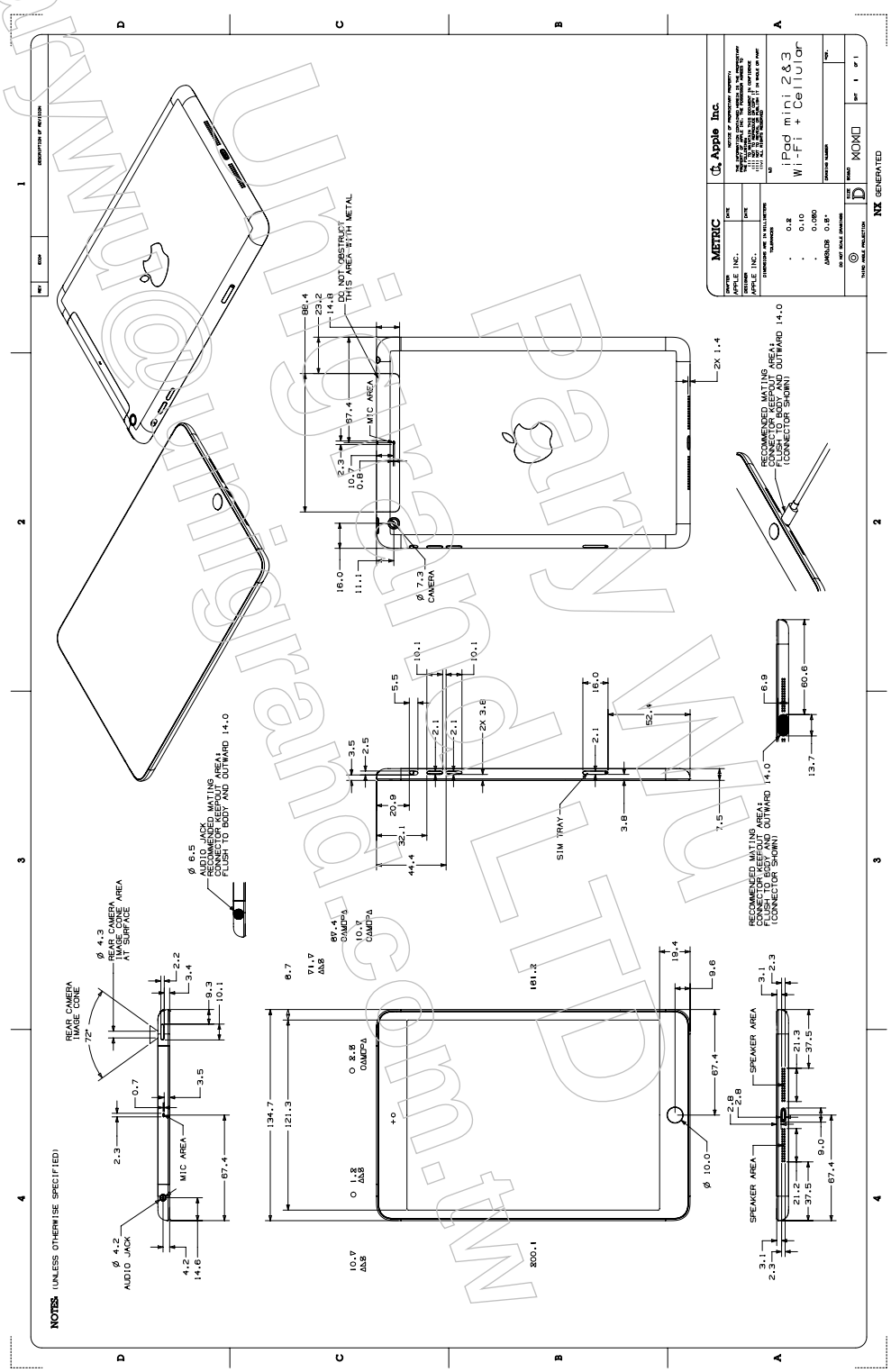
Figure 54-13 iPad mini 2 & iPad mini 3 Wi-Fi Dimensional Drawing



Party Wu Unigroup LTD
partywuunigroup.com.tw

54.14 iPad mini 2 & 3 Wi-Fi + Cellular

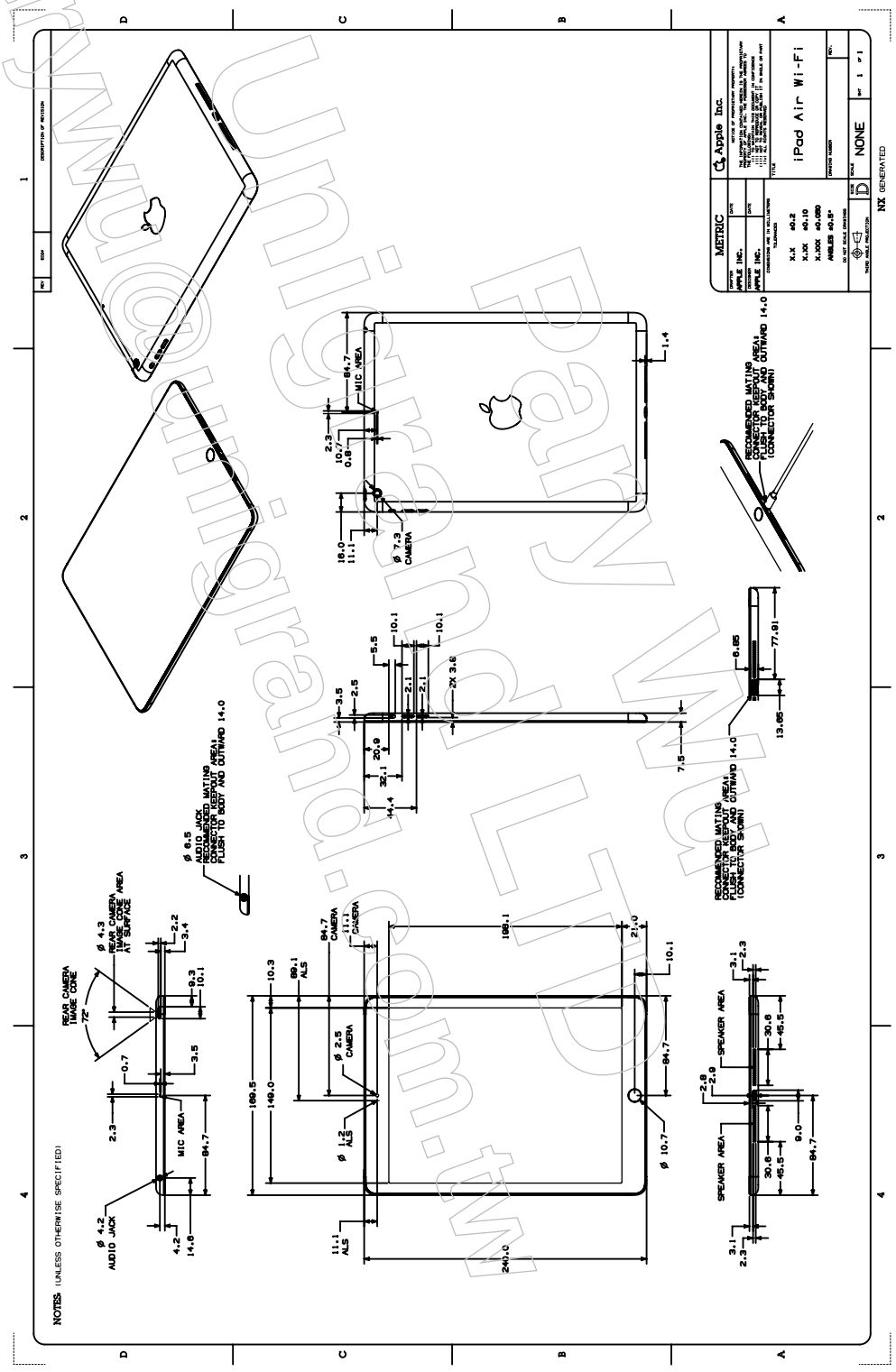
Figure 54-14 iPad mini 2 & iPad mini 3 Wi-Fi + Cellular Dimensional Drawing



Paraworld Untranslated.com.tw

54.15 iPad Air Wi-Fi

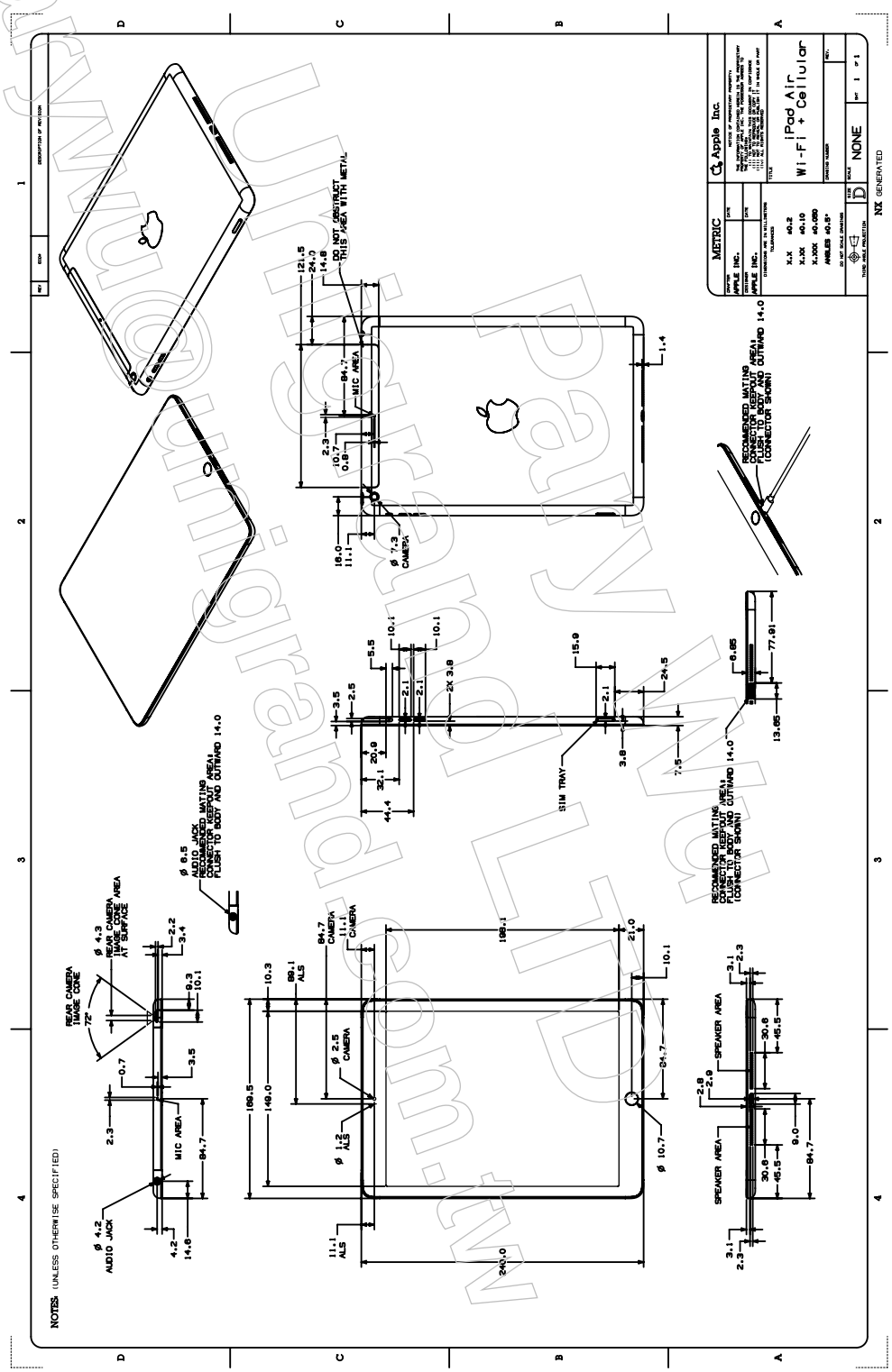
Figure 54-15 iPad Air Wi-Fi Dimensional Drawing



Partly Wistron Limited
Unauthorized.com.tw

54.16 iPad Air Wi-Fi + Cellular

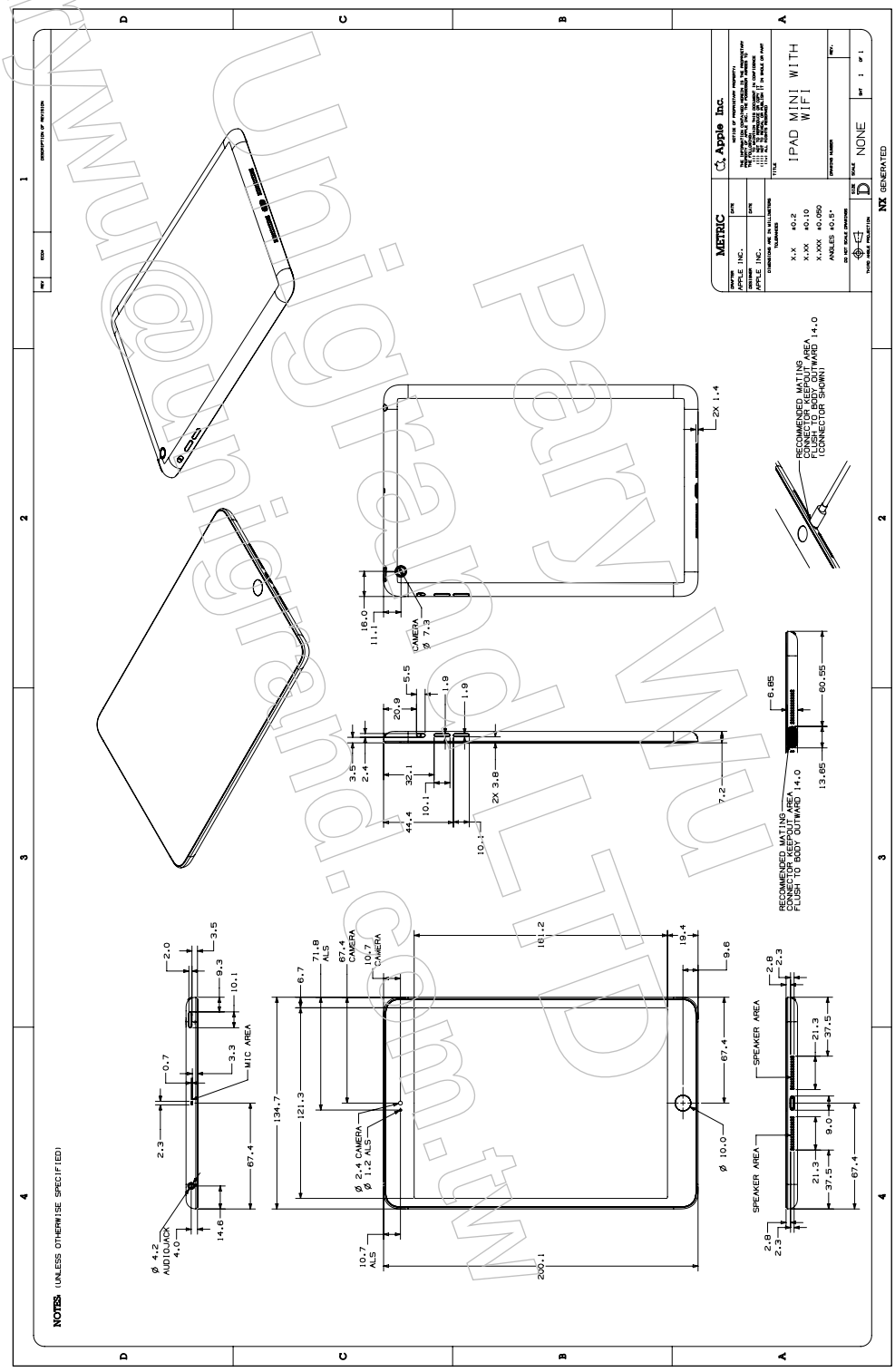
Figure 54-16 iPad Air Wi-Fi + Cellular Dimensional Drawing



Paywall Unlimited
Paywall Unlimited.com

54.17 iPad mini with Wi-Fi

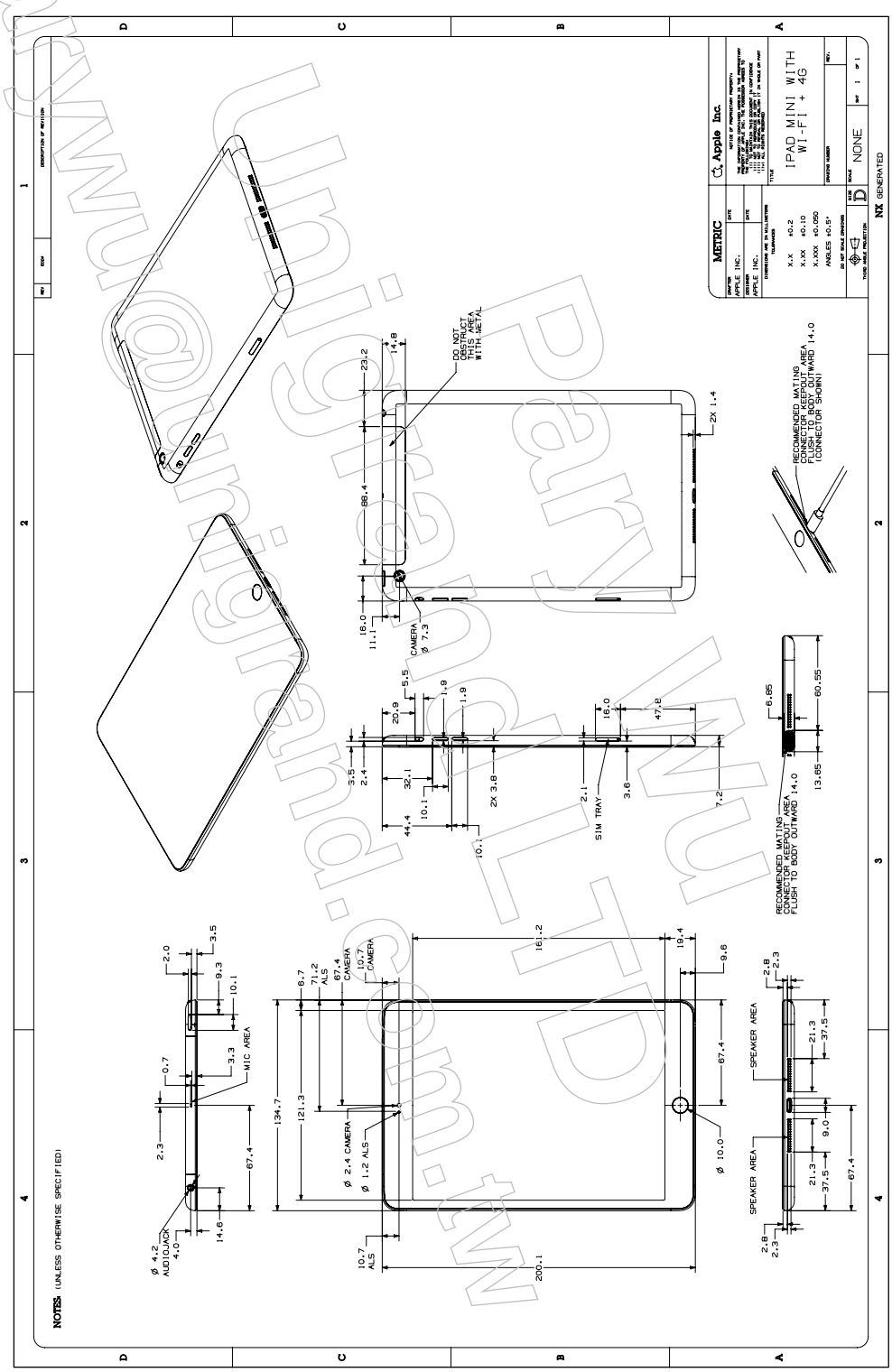
Figure 54-17 iPad mini with Wi-Fi Dimensional Drawing



Party Wu
Universal LTD
partywuuniversal.com.tw

54.18 iPad mini with Wi-Fi + Cellular

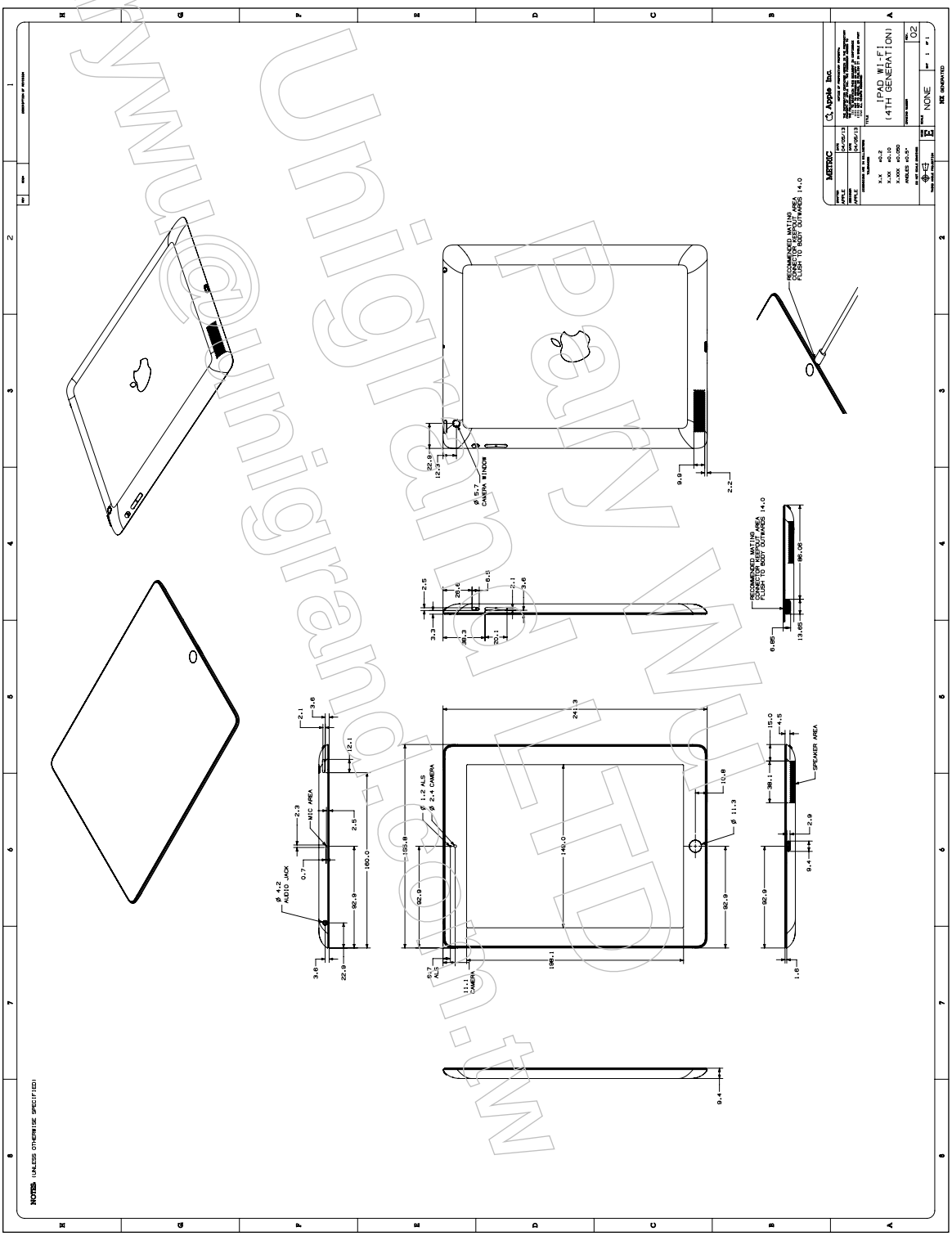
Figure 54-18 iPad mini Wi-Fi + Cellular Dimensional Drawing



Part Unreleased Unlicensed iTD.com.tw

54.19 iPad with Wi-Fi (4th generation)

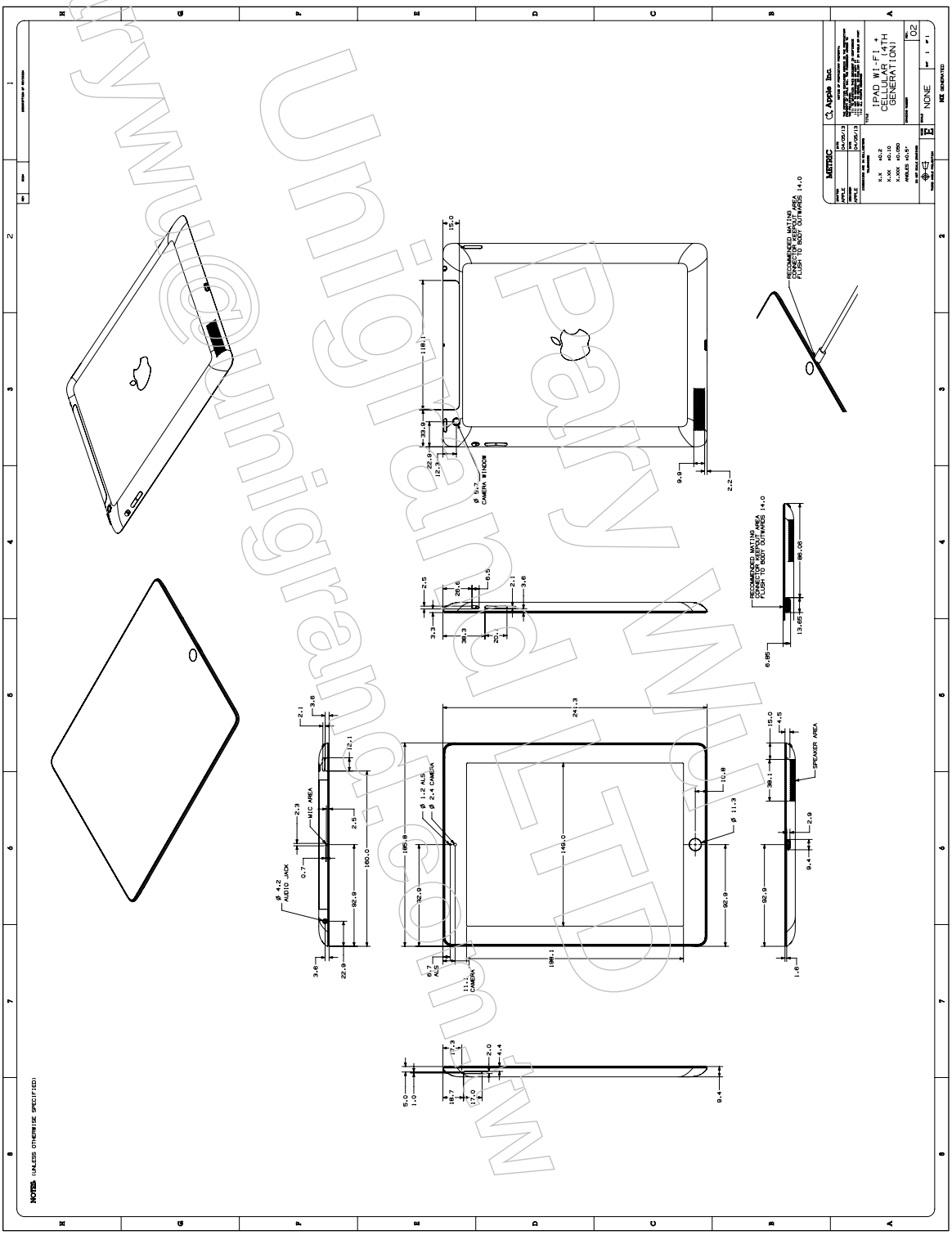
Figure 54-19 iPad Wi-Fi (4th generation) Dimensional Drawing



Party Wu@unigrandtech.com.tw

54.20 iPad with Wi-Fi + Cellular (4th generation)

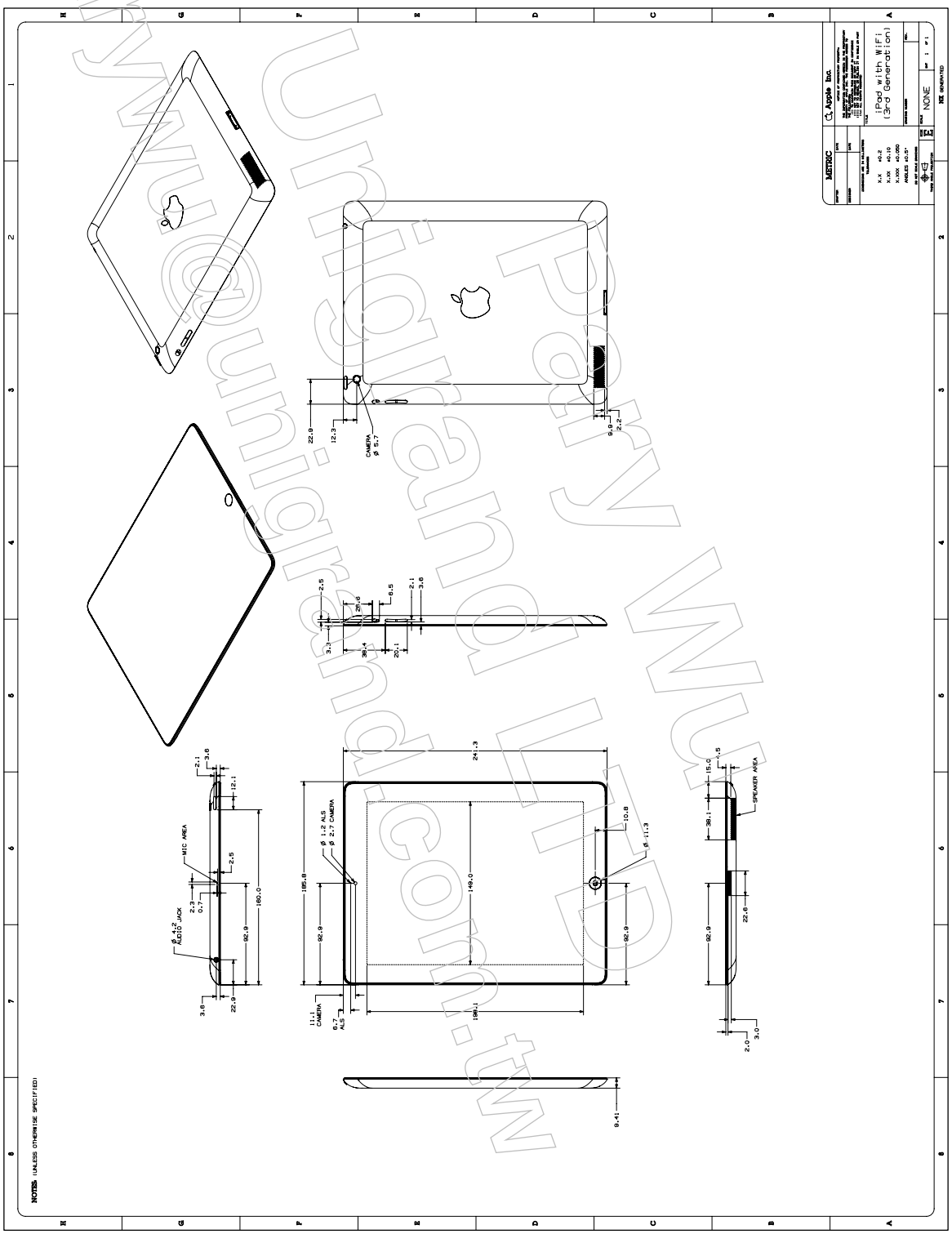
Figure 54-20 iPad Wi-Fi + Cellular (4th generation) Dimensional Drawing



Party Wired
party@unigrandtech.com

54.21 iPad with Wi-Fi (3rd generation)

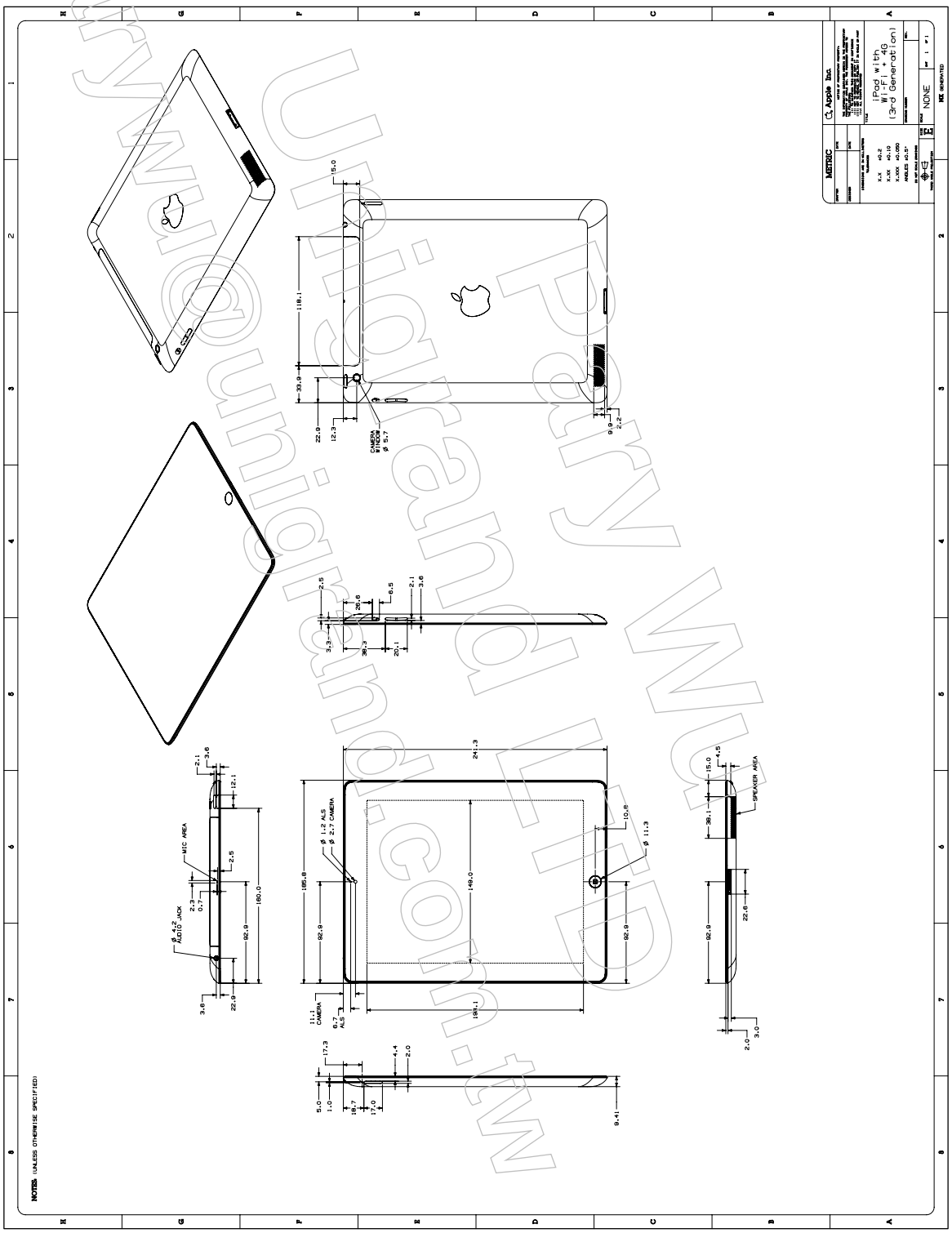
Figure 54-21 iPad Wi-Fi (3rd Generation) Dimensional Drawing



Party Wu@unigrand.com.tw Unigrand Ltd.

54.22 iPad with Wi-Fi + 4G (3rd generation)

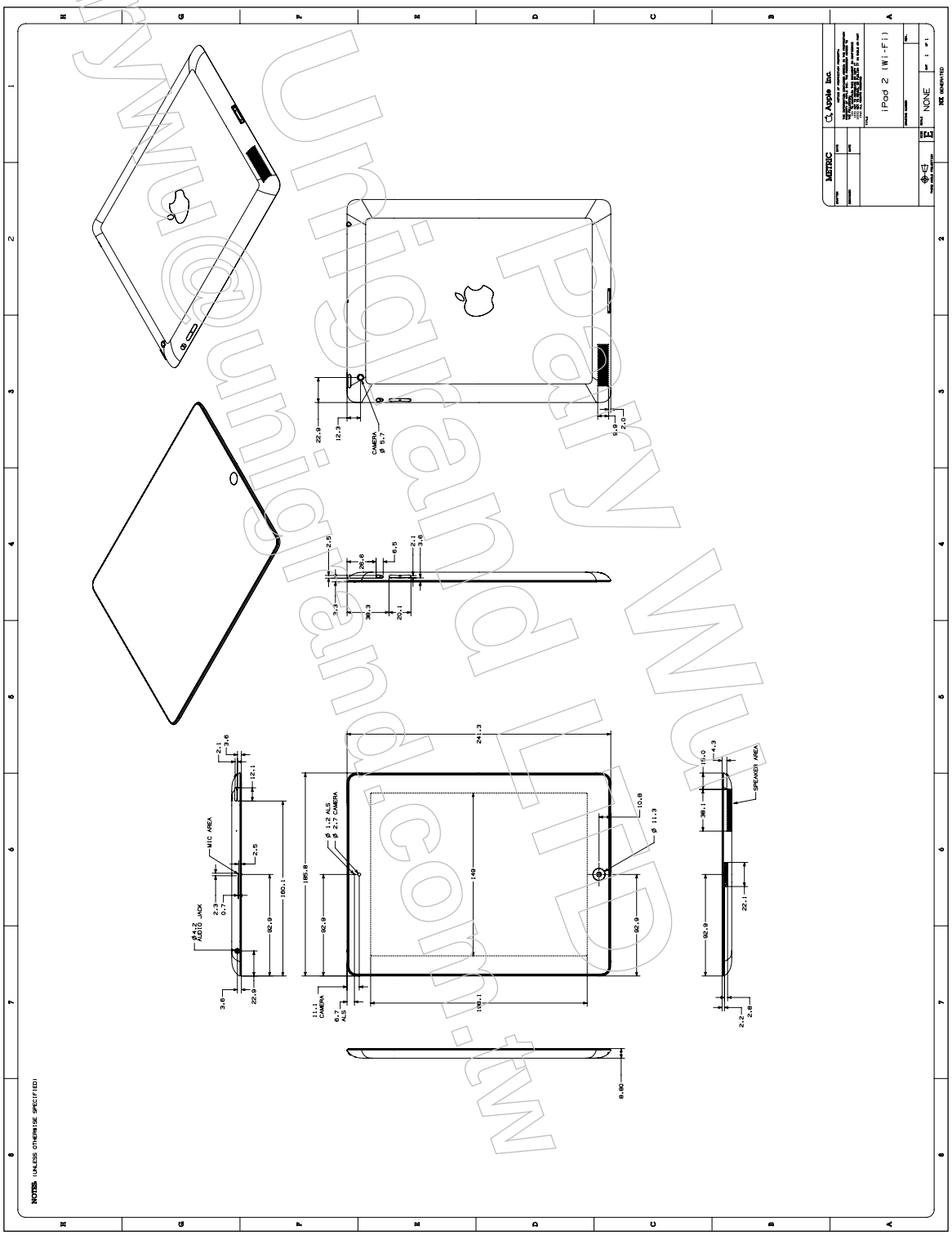
Figure 54-22 iPad Wi-Fi + 4G (3rd Generation) Dimensional Drawing



Party Wu
Unigrand Ltd
Party@unigrand.com.tw

54.23 iPad 2 with Wi-Fi

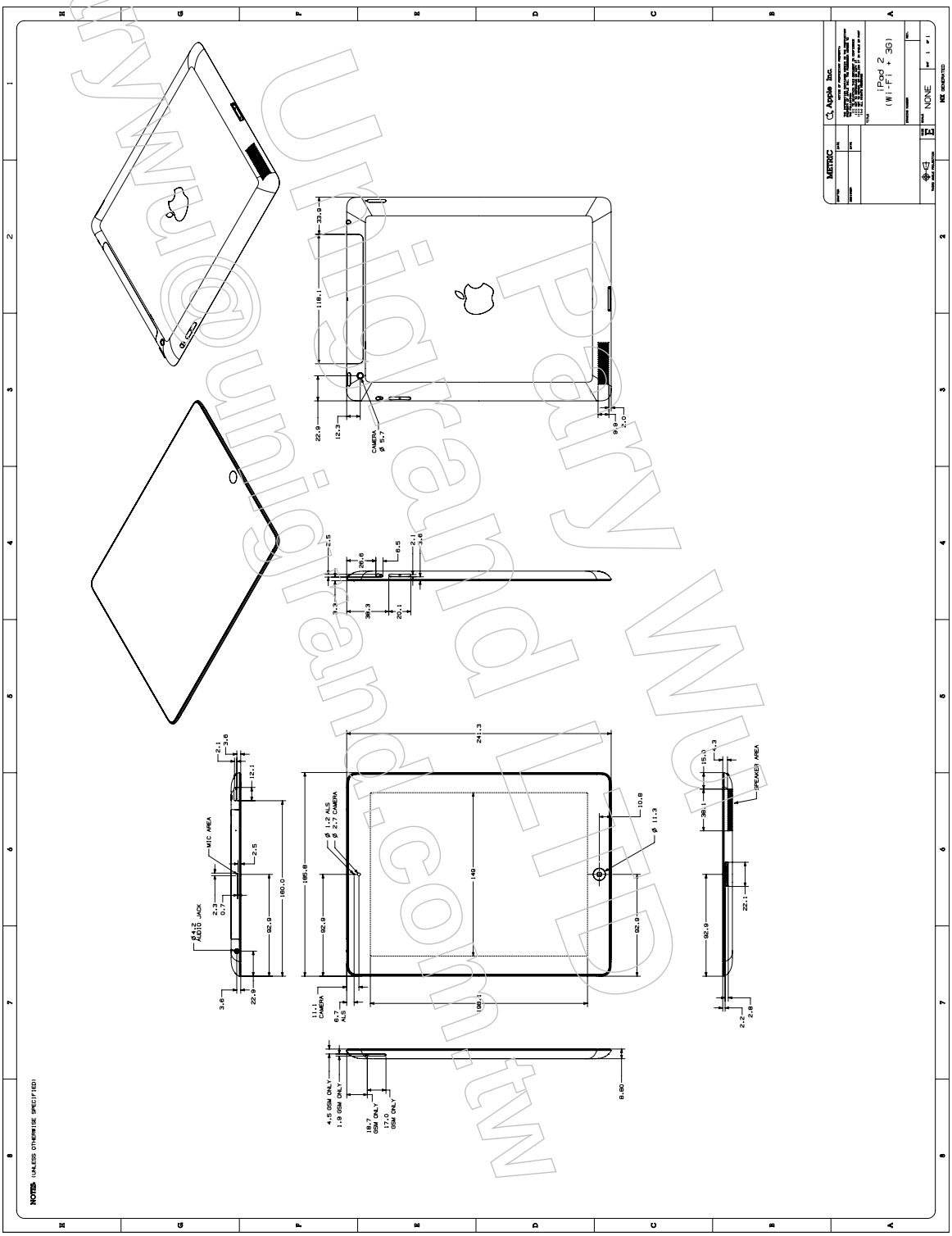
Figure 54-23 iPad 2 Wi-Fi Dimensional Drawing



Party Wu
Unauthorized
party@unauthorized.com.tw

54.24 iPad 2 with Wi-Fi + 3G

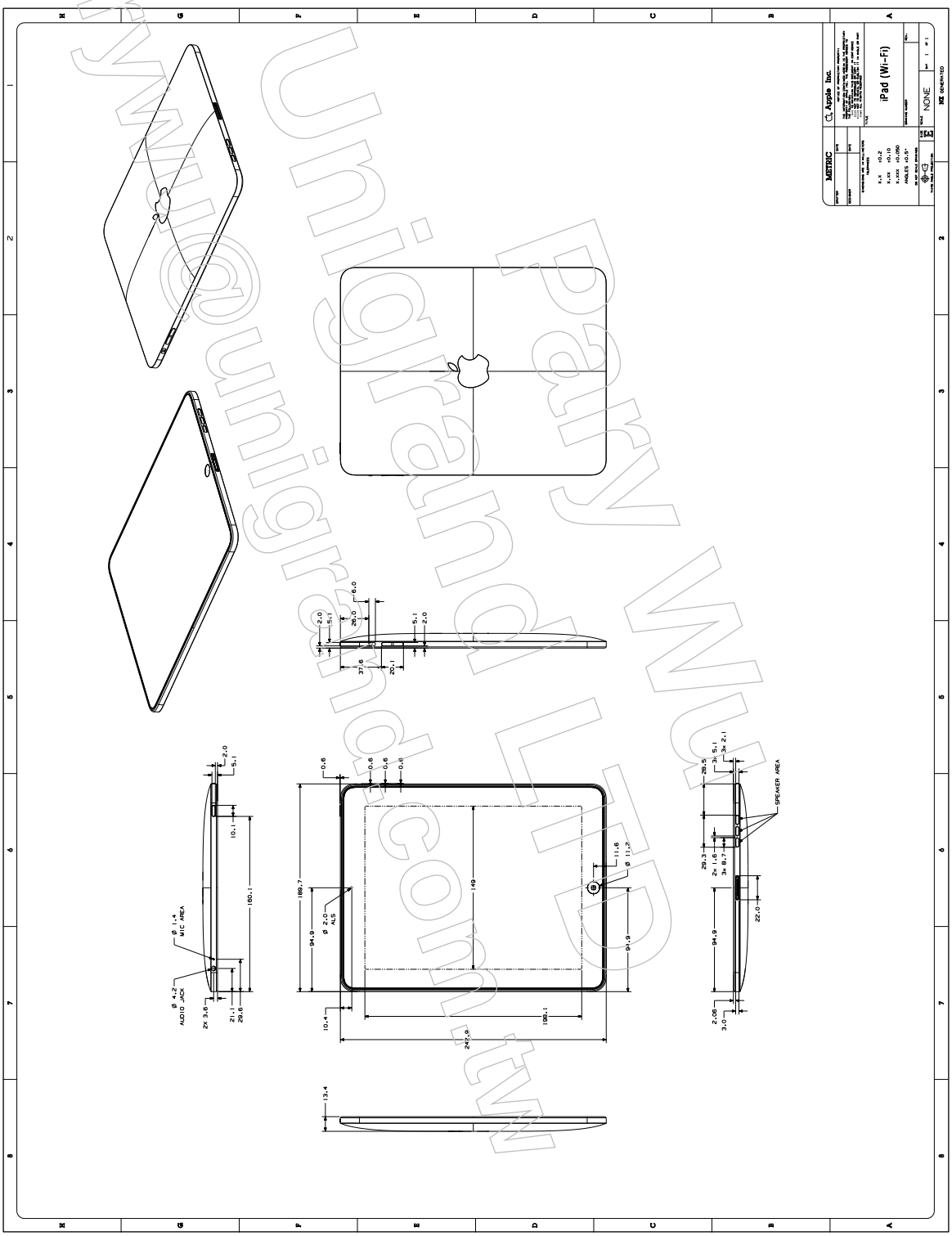
Figure 54-24 iPad 2 Wi-Fi + 3G Dimensional Drawing



Party and Untied
Party@untiedandparty.com

54.25 iPad with Wi-Fi

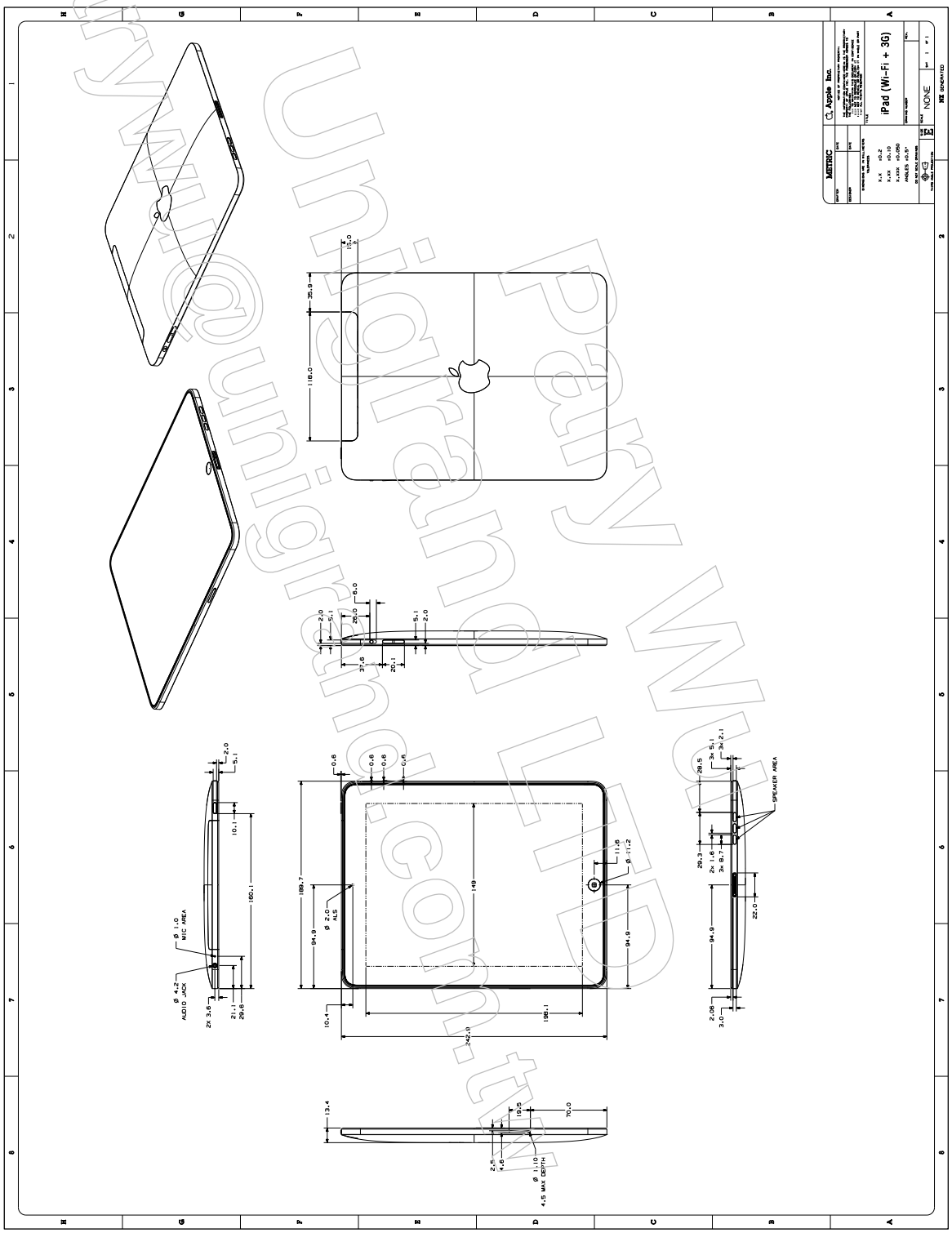
Figure 54-25 iPad Wi-Fi Dimensional Drawing



Party Wu
Unigrand Ltd
partywu@unigrand.com.tw

54.26 iPad with Wi-Fi + 3G

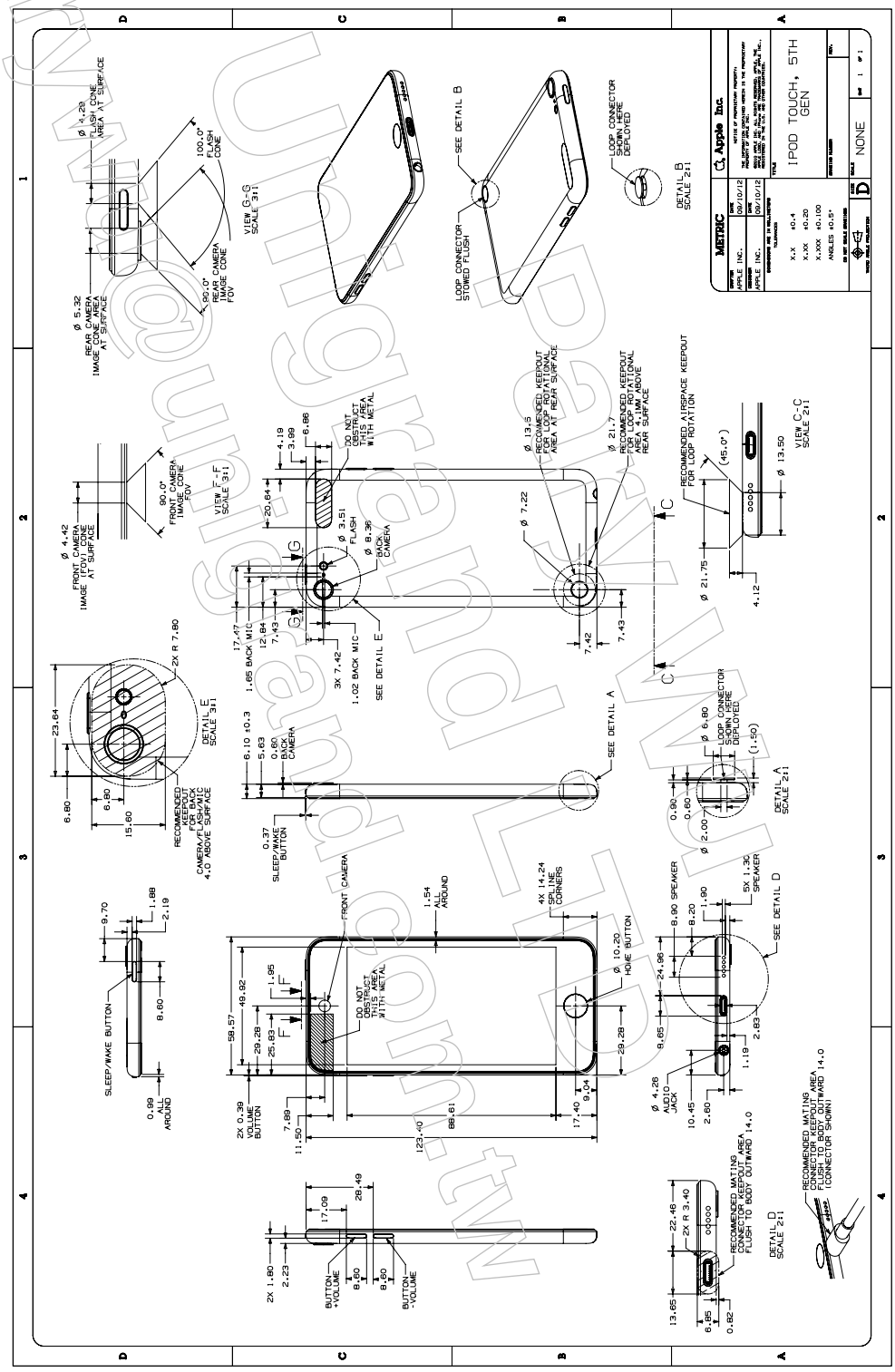
Figure 54-26 iPad Wi-Fi + 3G Dimensional Drawing



Party Without Unigrand Limited

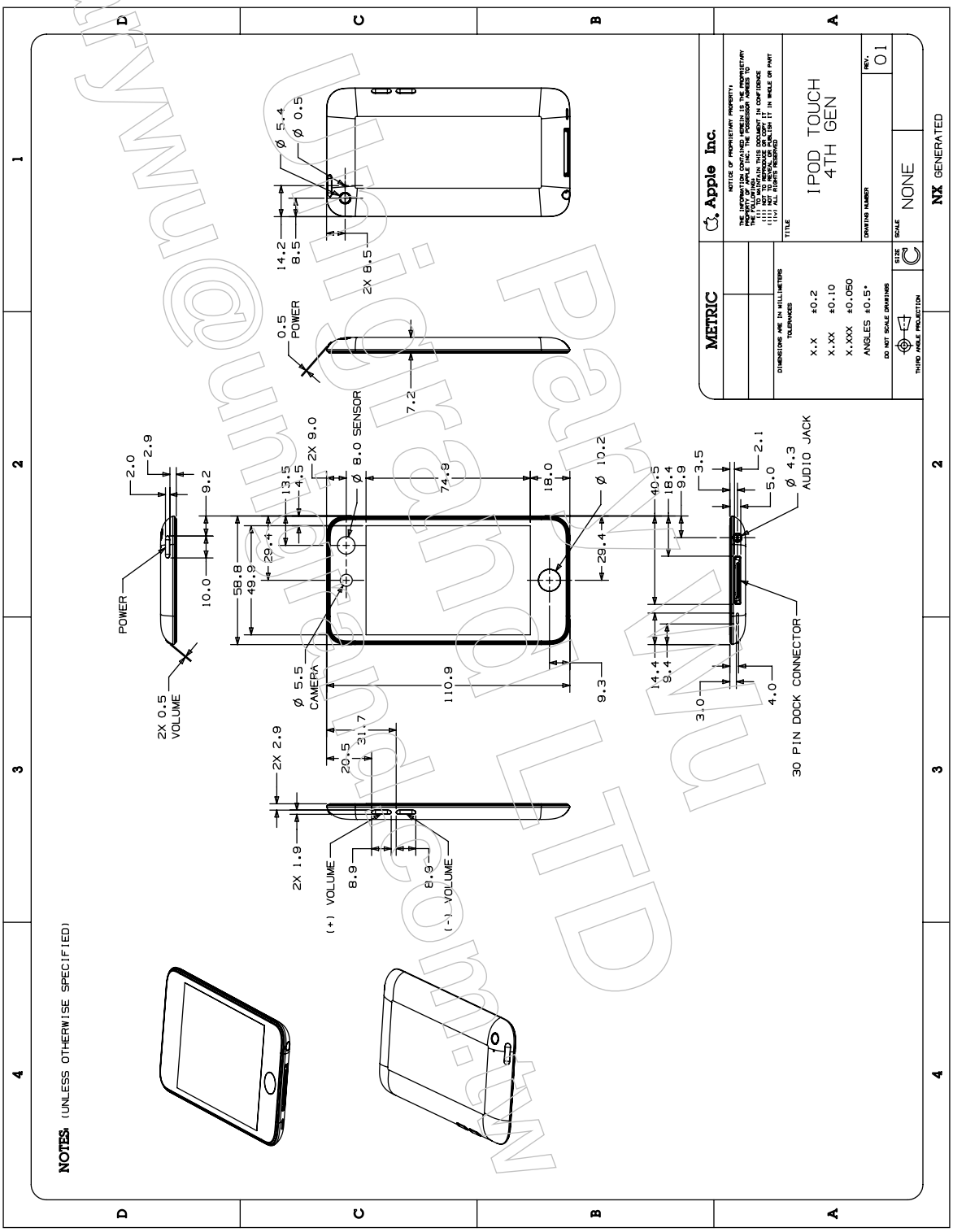
54.27 iPod touch (5th generation)

Figure 54-27 iPod touch 5th gen. Dimensional Drawing



54.28 iPod touch (4th generation)

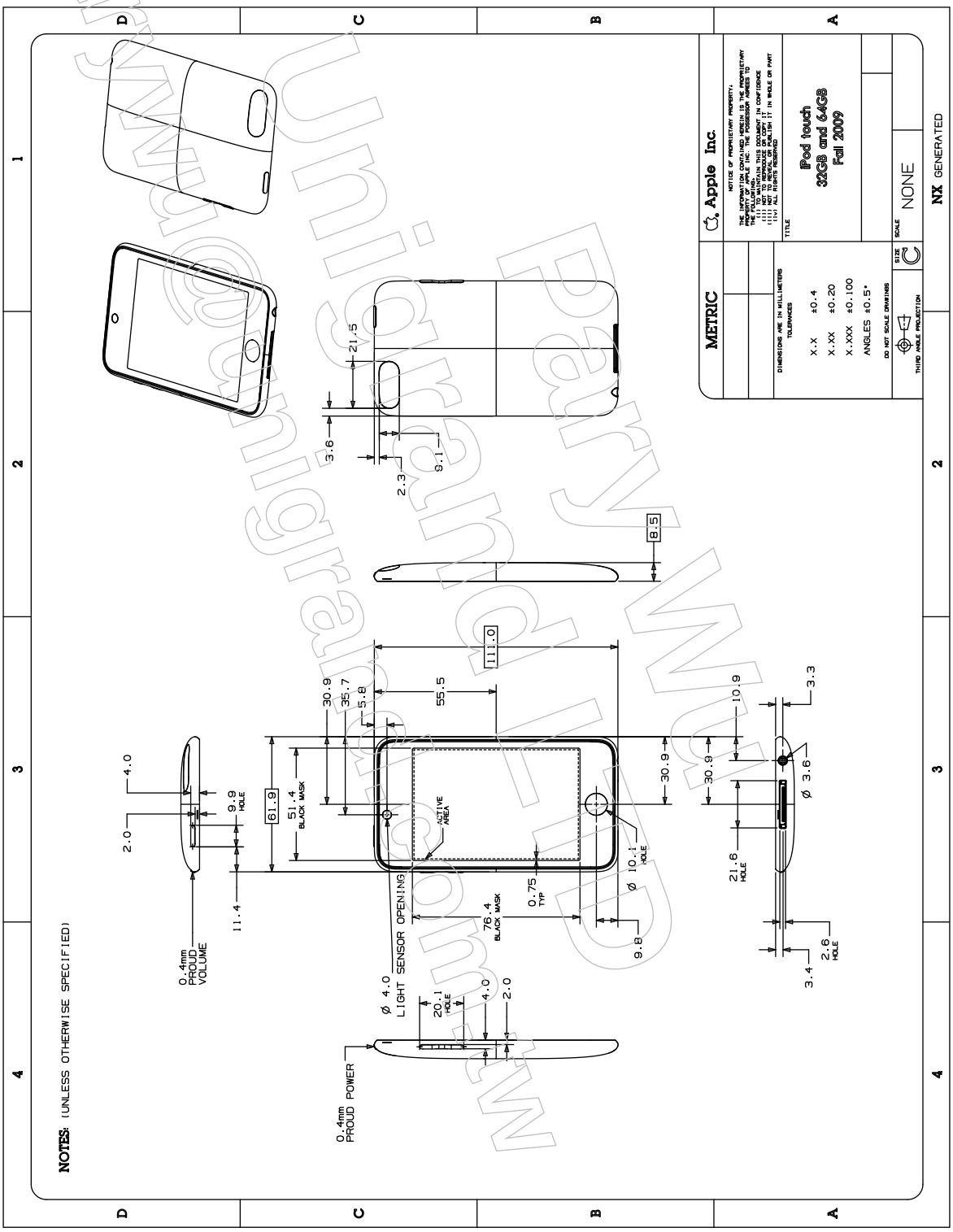
Figure 54-28 iPod touch 4th gen. Dimensional Drawing



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

54.29 iPod touch (3rd generation)

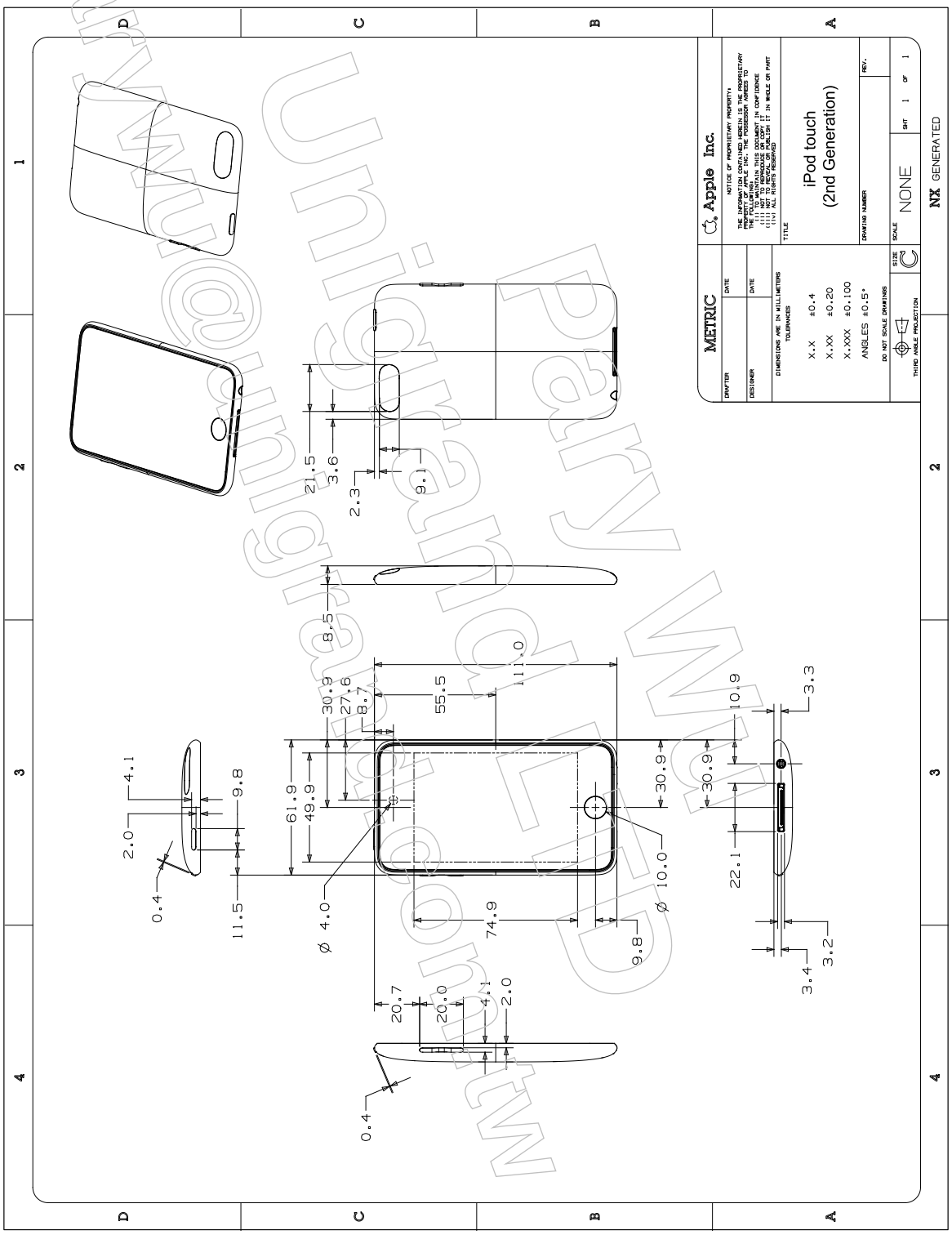
Figure 54-29 iPod touch 3rd gen. Fall '09 32GB and 64GB Dimensional Drawing



Unregistered Party Watermarked

54.30 iPod touch (2nd generation)

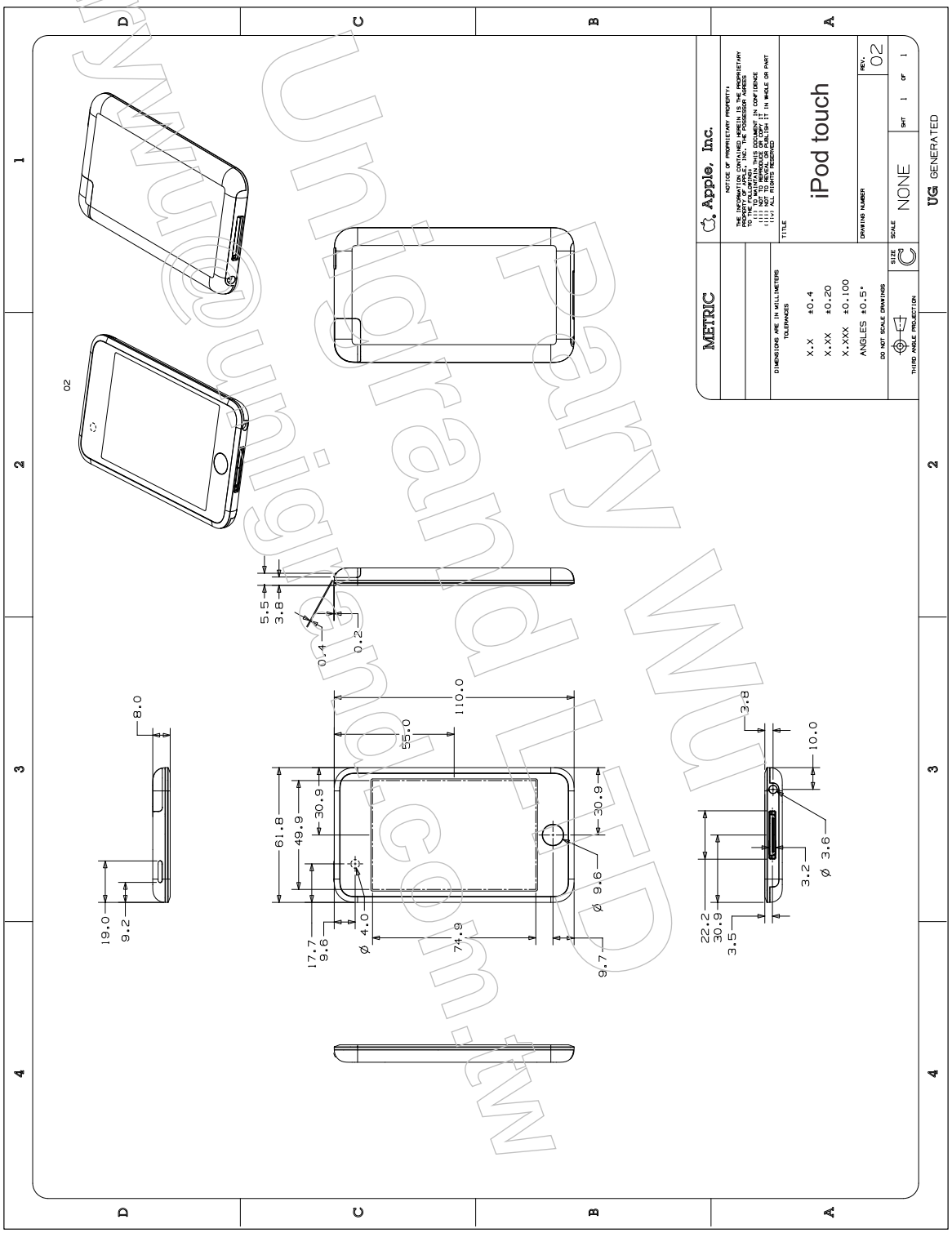
Figure 54-30 iPod touch 2nd gen. 8GB, 16GB, 32GB Dimensional Drawing



Party Unregistered Unregistered Party Viewing Only

54.31 iPod touch

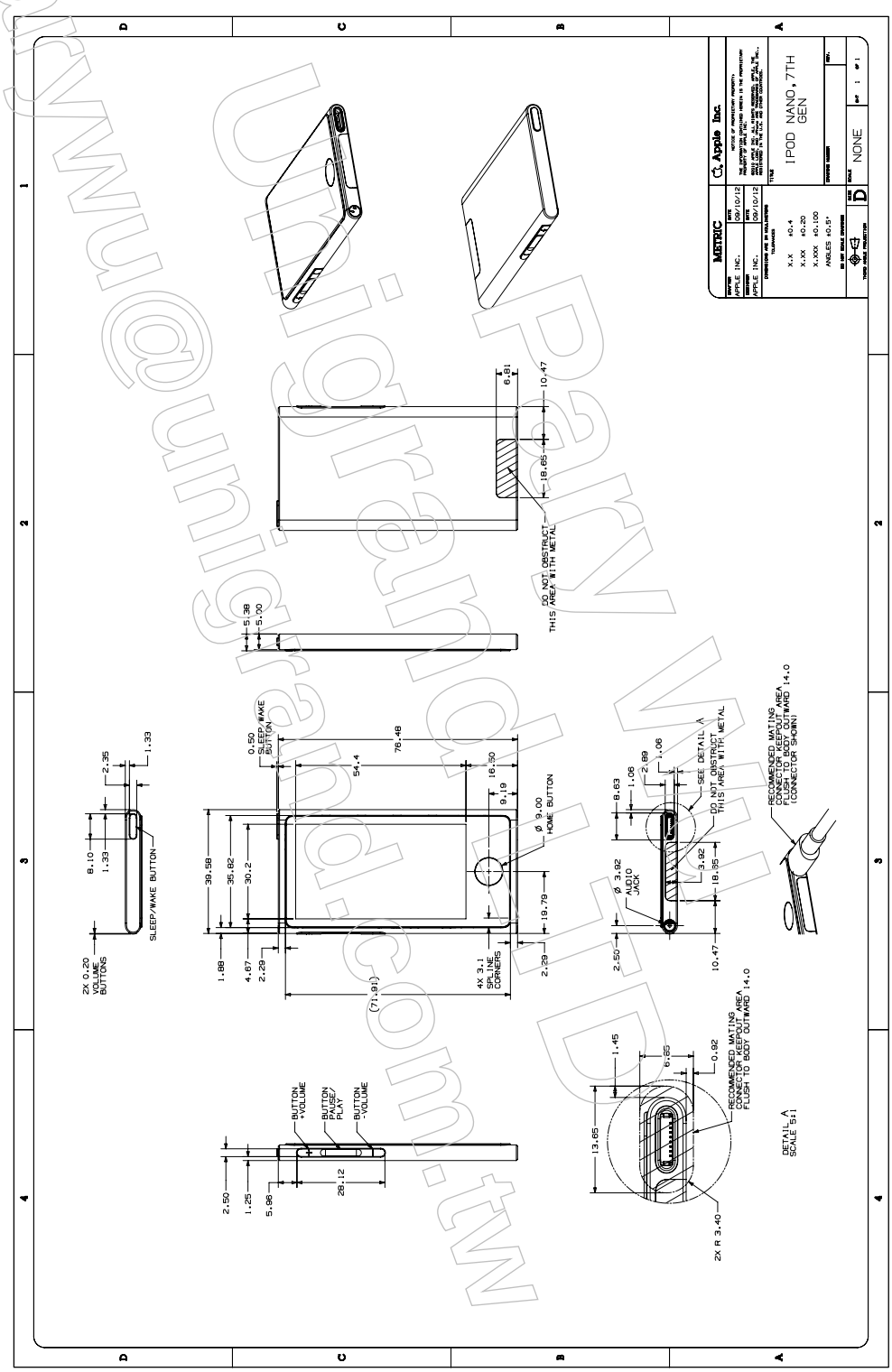
Figure 54-31 iPod touch Dimensional Drawing



Preliminary Unigraphics File
Property of Unigraphics.com.tw

54.32 iPod nano (7th generation)

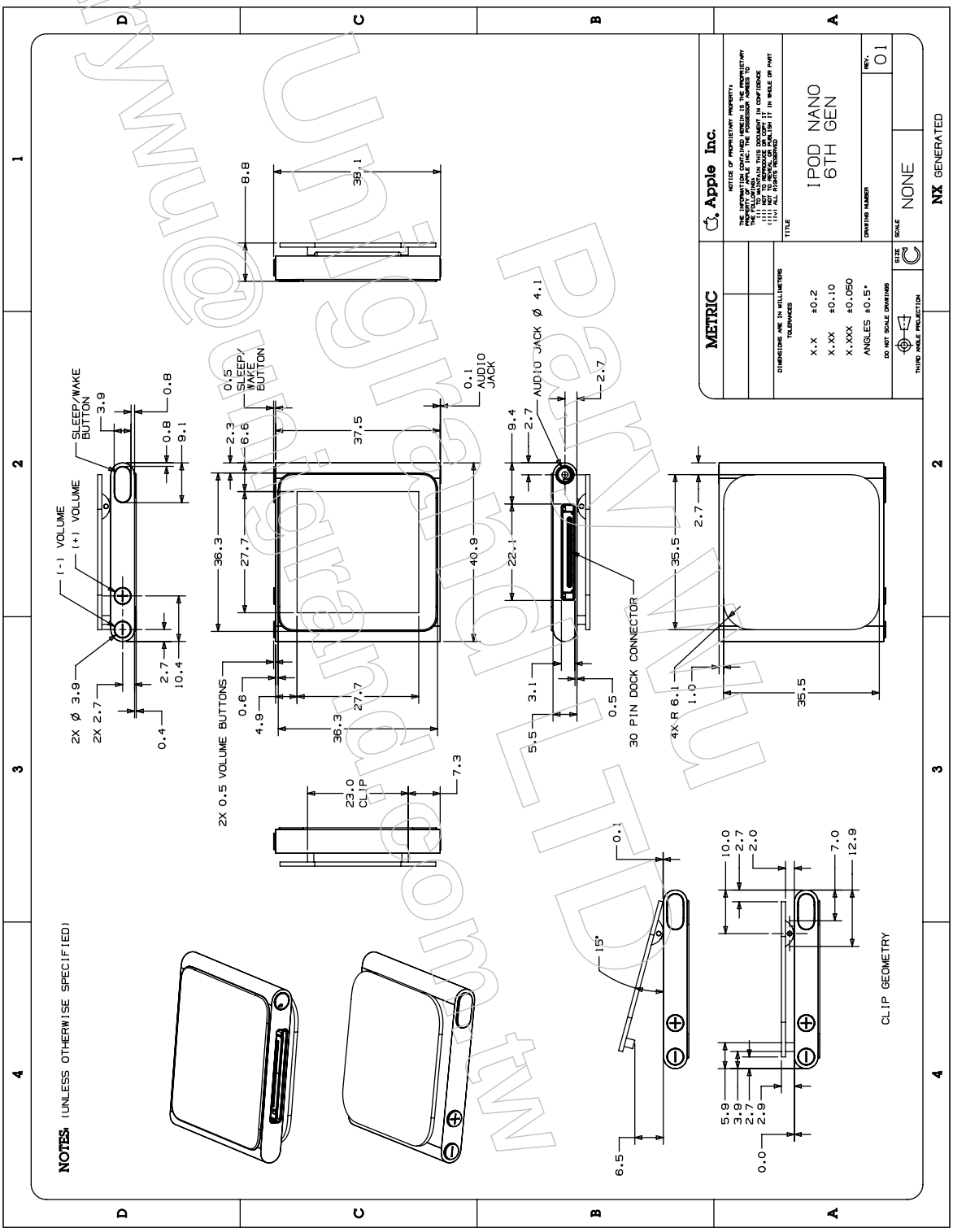
Figure 54-32 iPod nano 7th gen. Dimensional Drawing



Party Unigrand Ltd.
Party@unigrand.com.tw

54.33 iPod nano (6th generation)

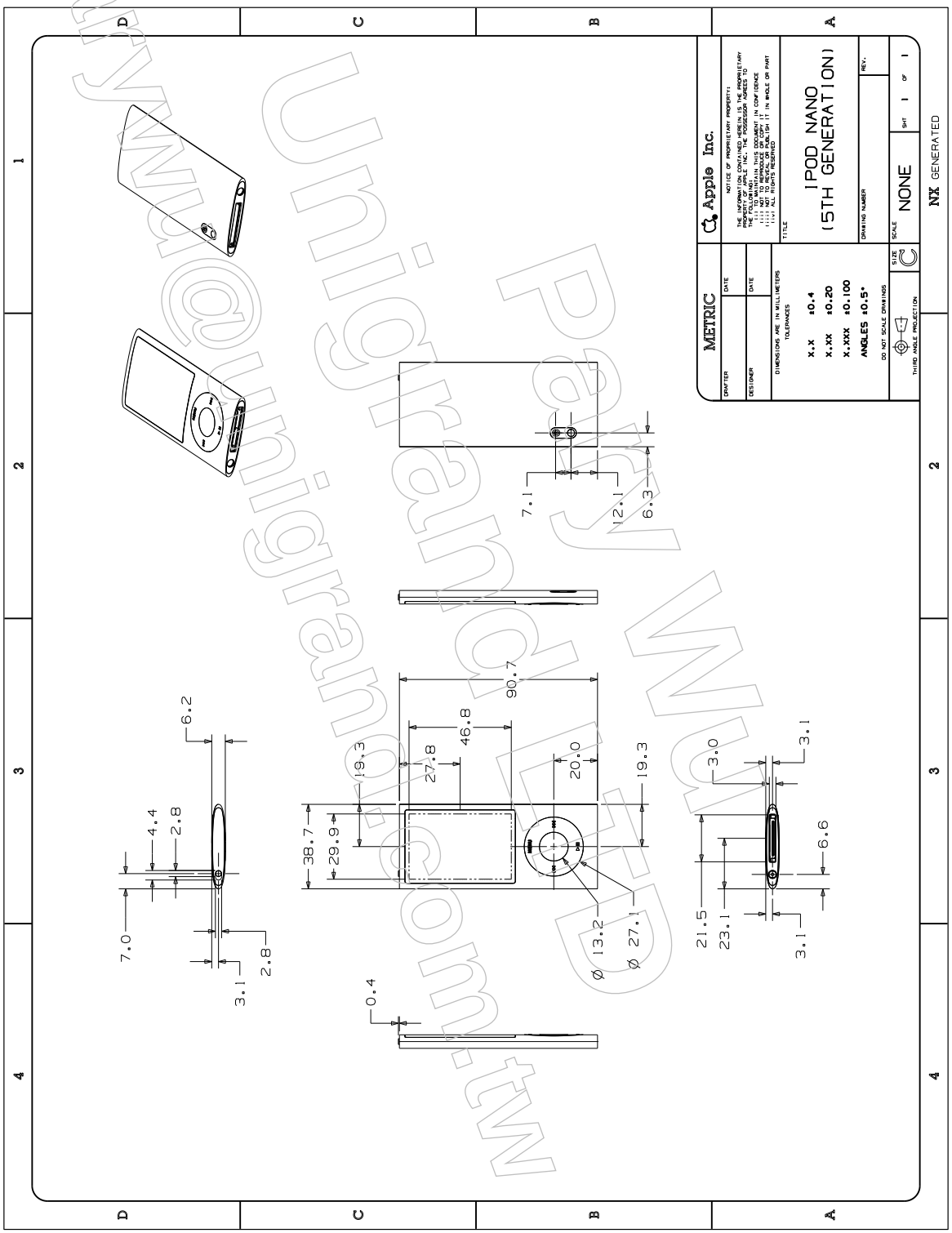
Figure 54-33 iPod nano 6th gen. Dimensional Drawing



Unofficial Parts Guide
Parts not to be reproduced

54.34 iPod nano (5th generation)

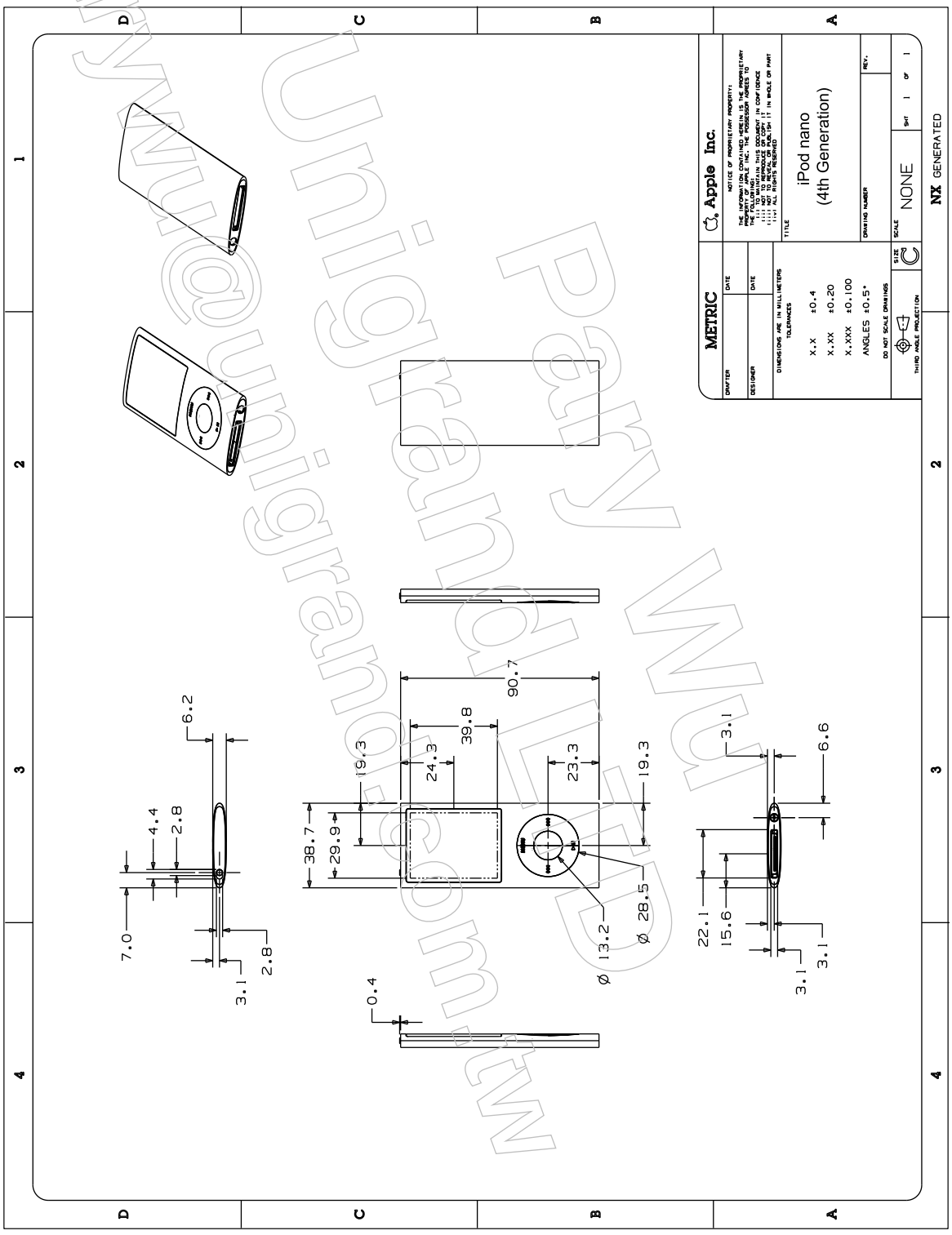
Figure 54-34 iPod nano 5th gen. Dimensional Drawing



Unigrandonline.com
Party Machine

54.35 iPod nano (4th generation)

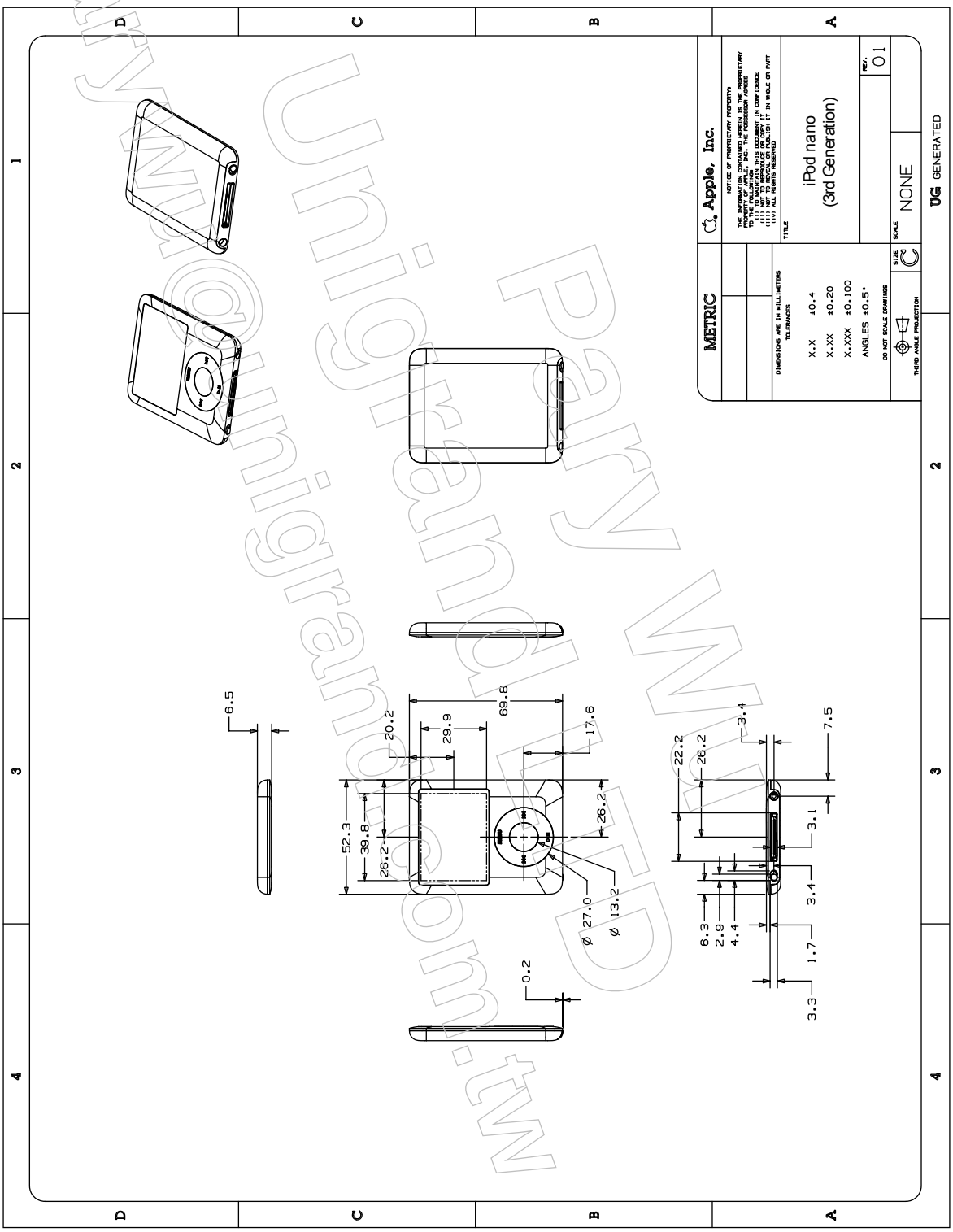
Figure 54-35 iPod nano 4th gen. Dimensional Drawing



Party Unigrand LTD
partyunigrand.com

54.36 iPod nano (3rd generation)

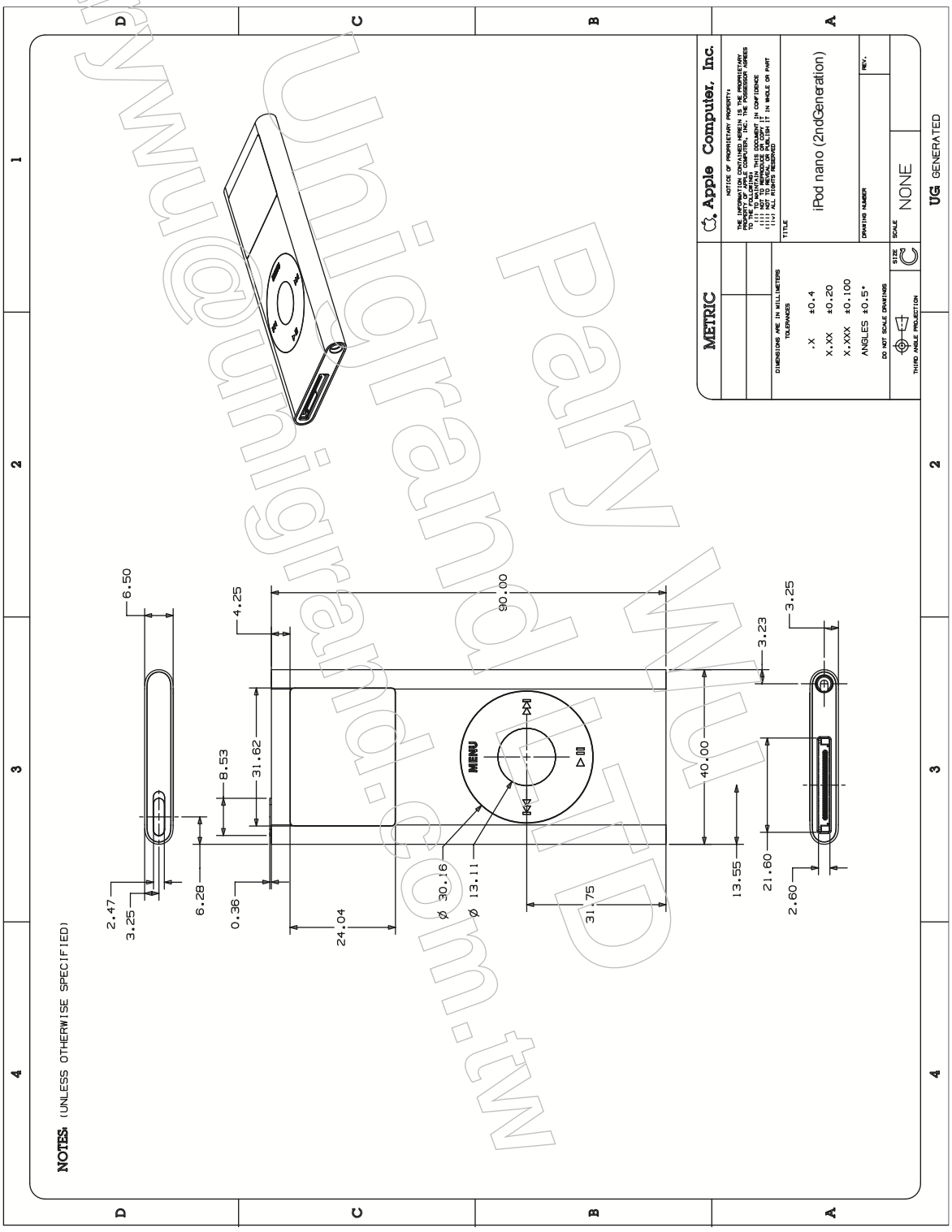
Figure 54-36 iPod nano 3rd gen. Dimensional Drawing



Party Unigrandled
Party Unigrandled.com.tw

54.37 iPod nano (2nd generation)

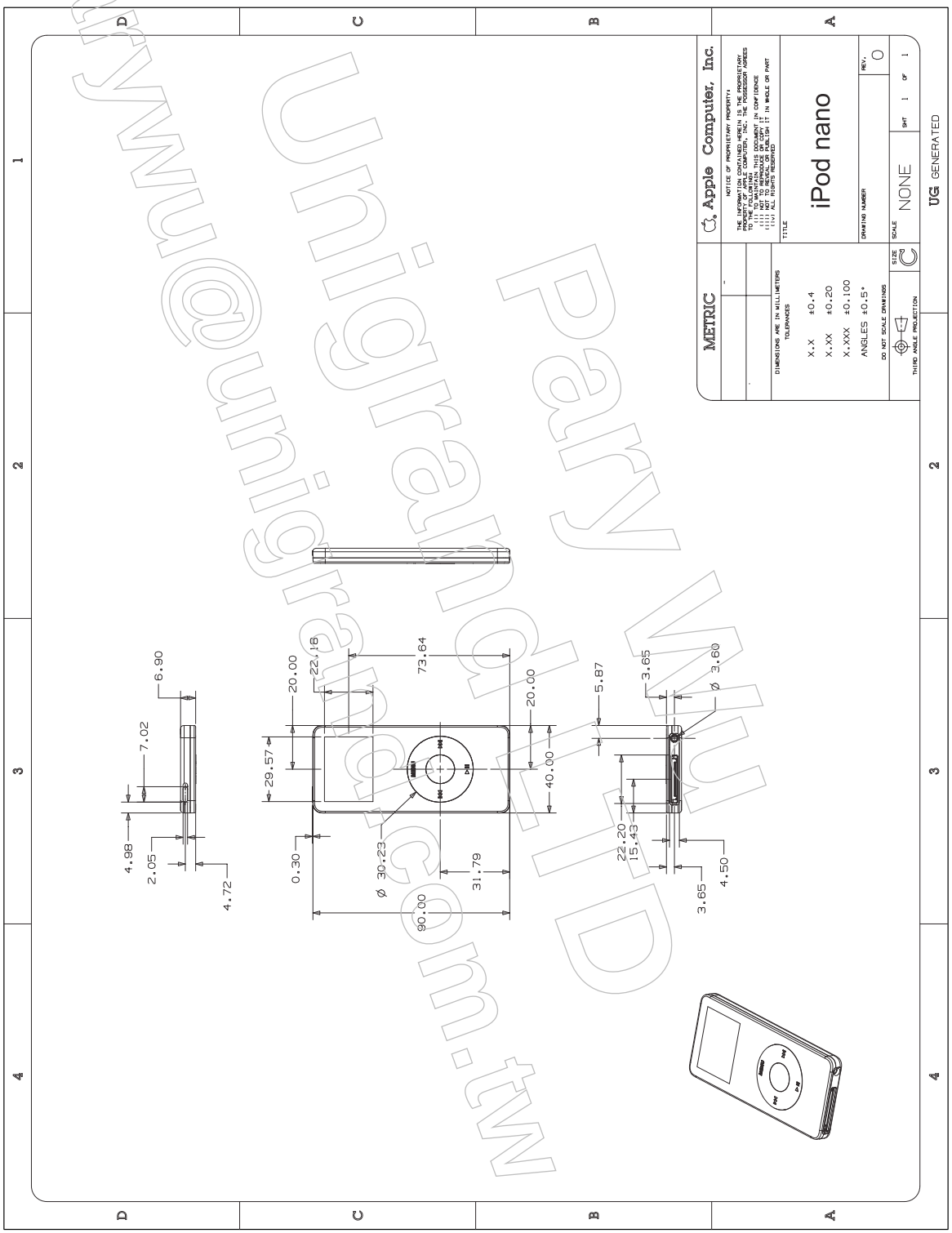
Figure 54-37 iPod nano 2nd gen. Dimensional Drawing



Unigrand
Party Wu
partywu@unigrand.com.tw

54.38 iPod nano

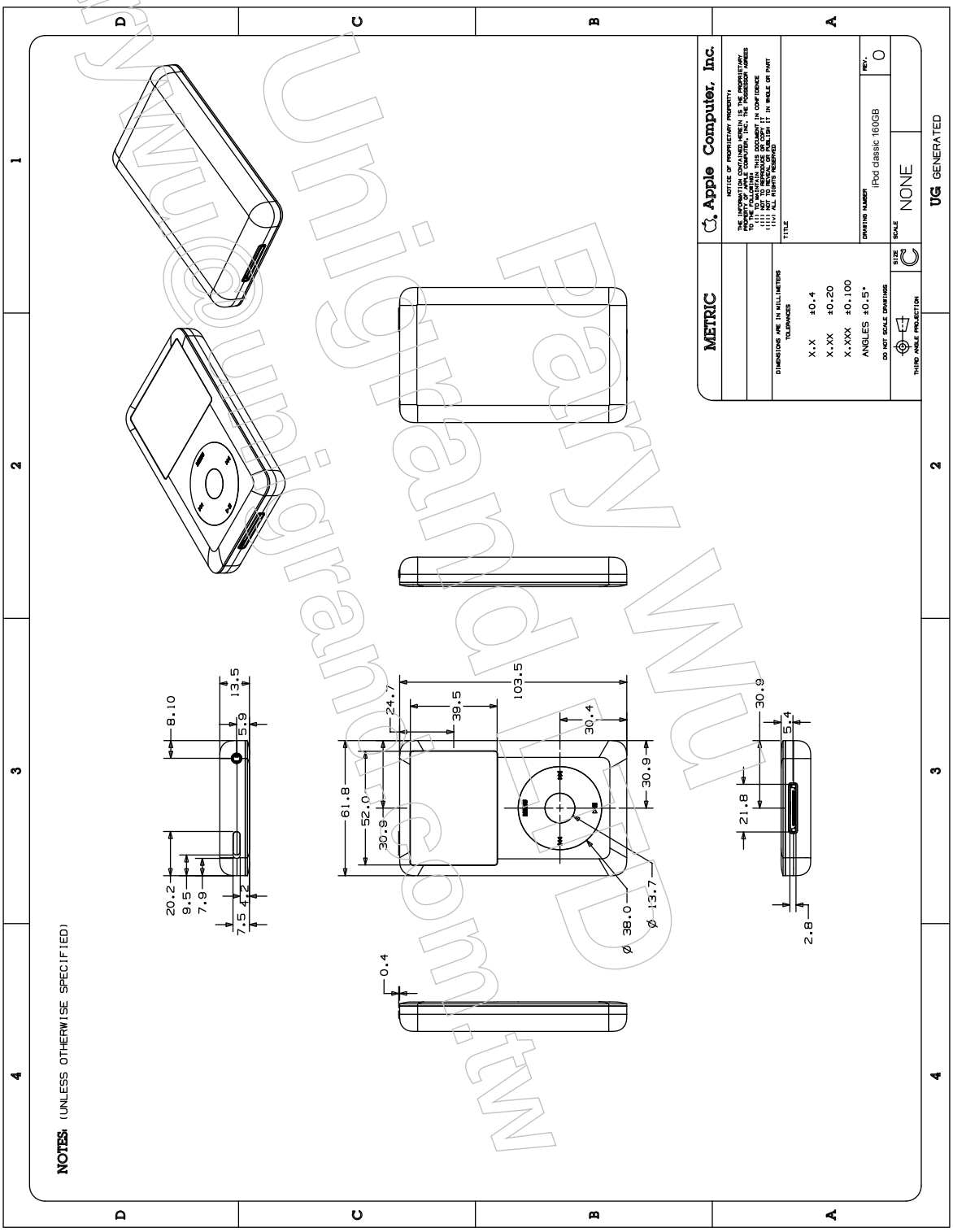
Figure 54-38 iPod nano Dimensional Drawing



Party Unigrand Unigrandtech.com.tw

54.39 iPod classic 160GB

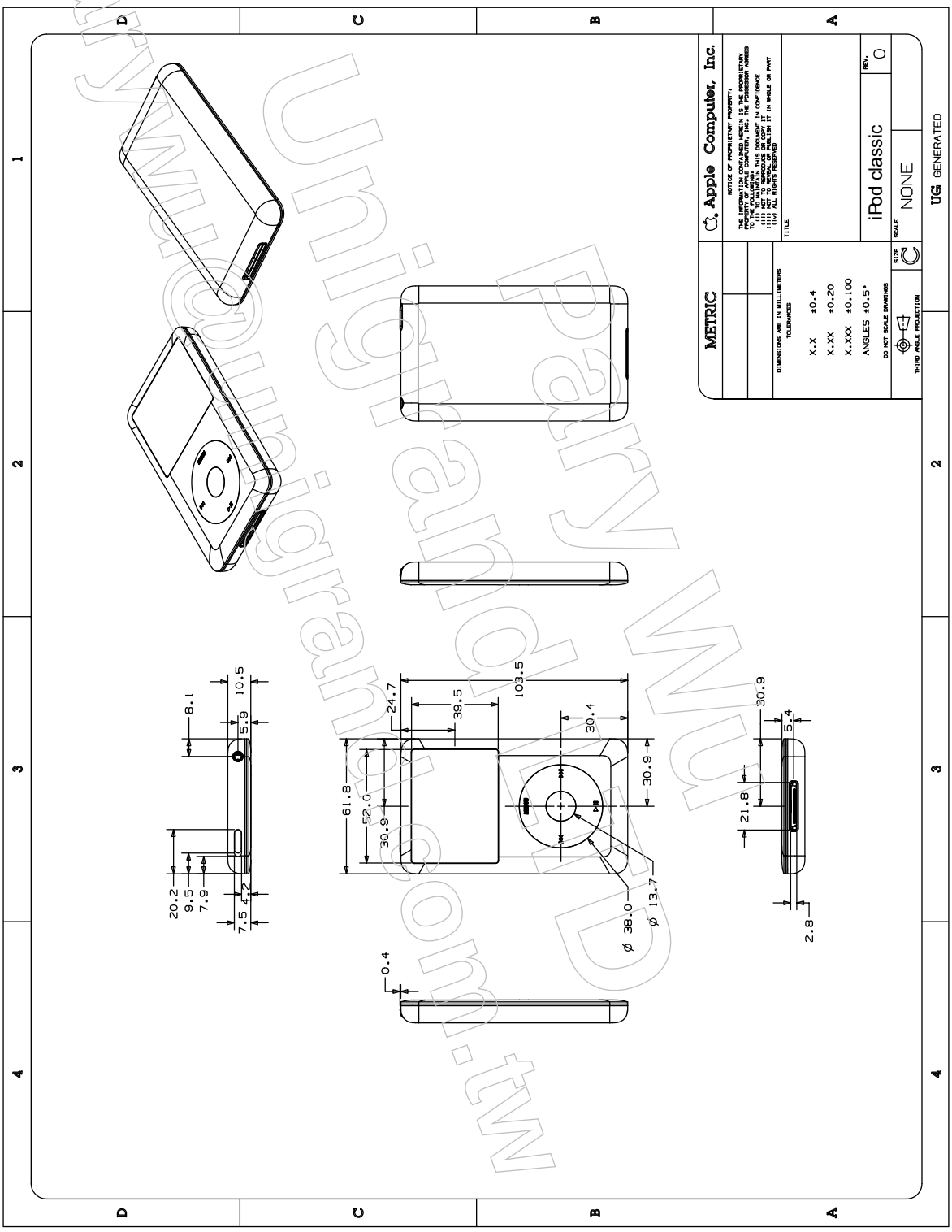
Figure 54-39 iPod classic 160GB Dimensional Drawing



2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

54.40 iPod classic 80GB

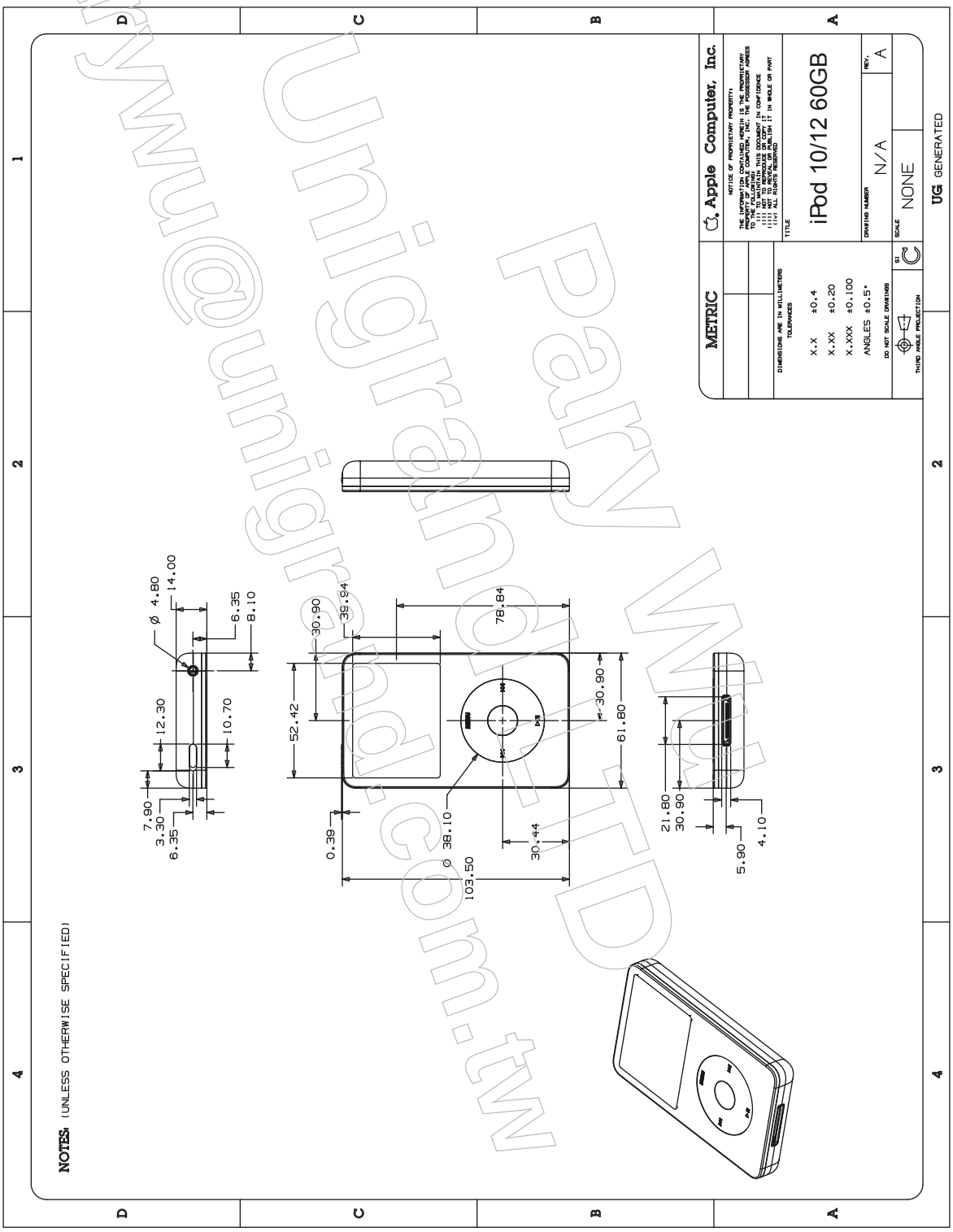
Figure 54-40 iPod classic 80GB Dimensional Drawing



Party Wu
Unigiandroid@gmail.com

54.41 iPod (5th generation) 60GB/80GB

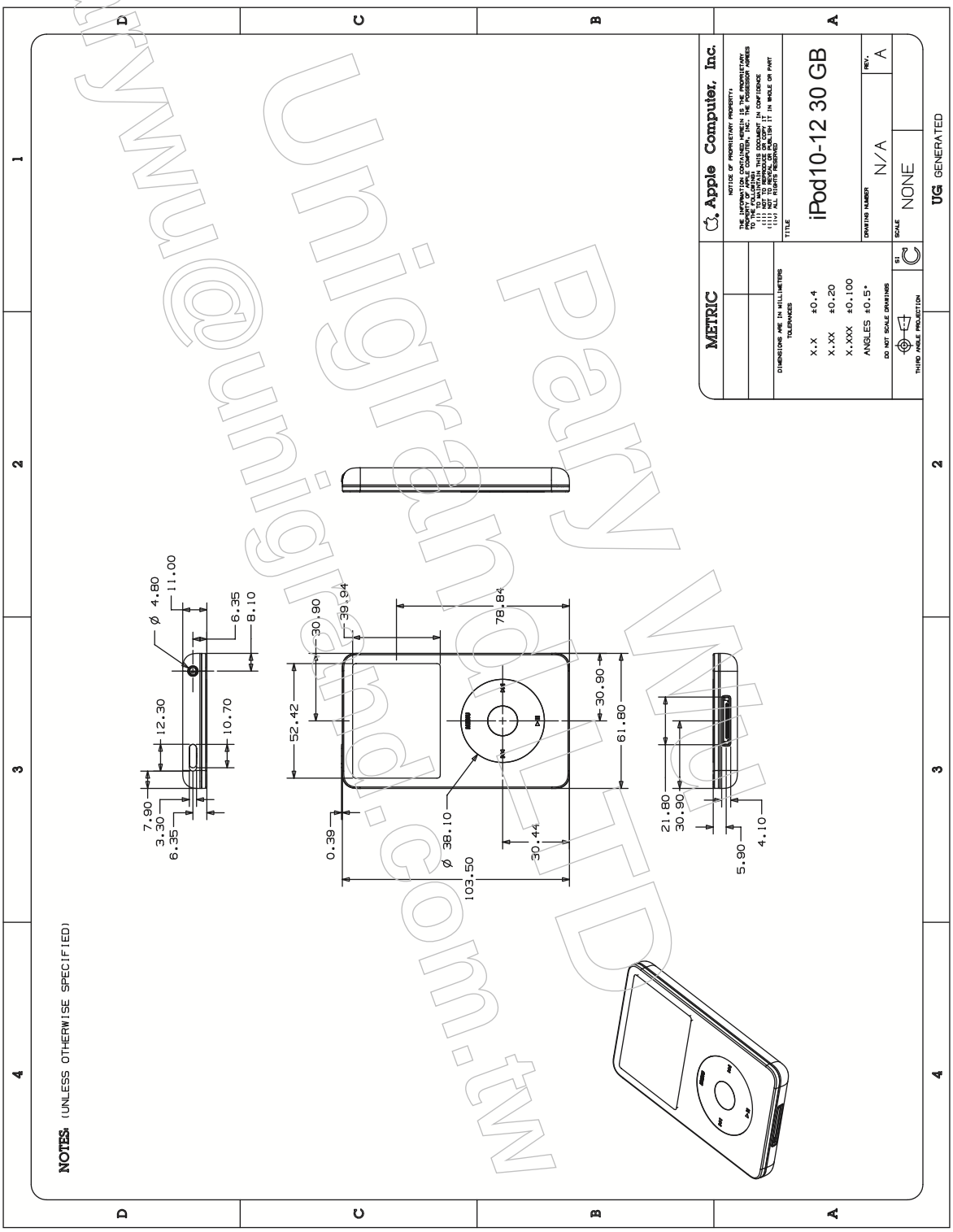
Figure 54-41 iPod 5th gen. 60GB/80GB Dimensional Drawing



Party Wu@unigration.com.tw
Party Wu@unigration.com.tw

54.42 iPod (5th generation) 30GB

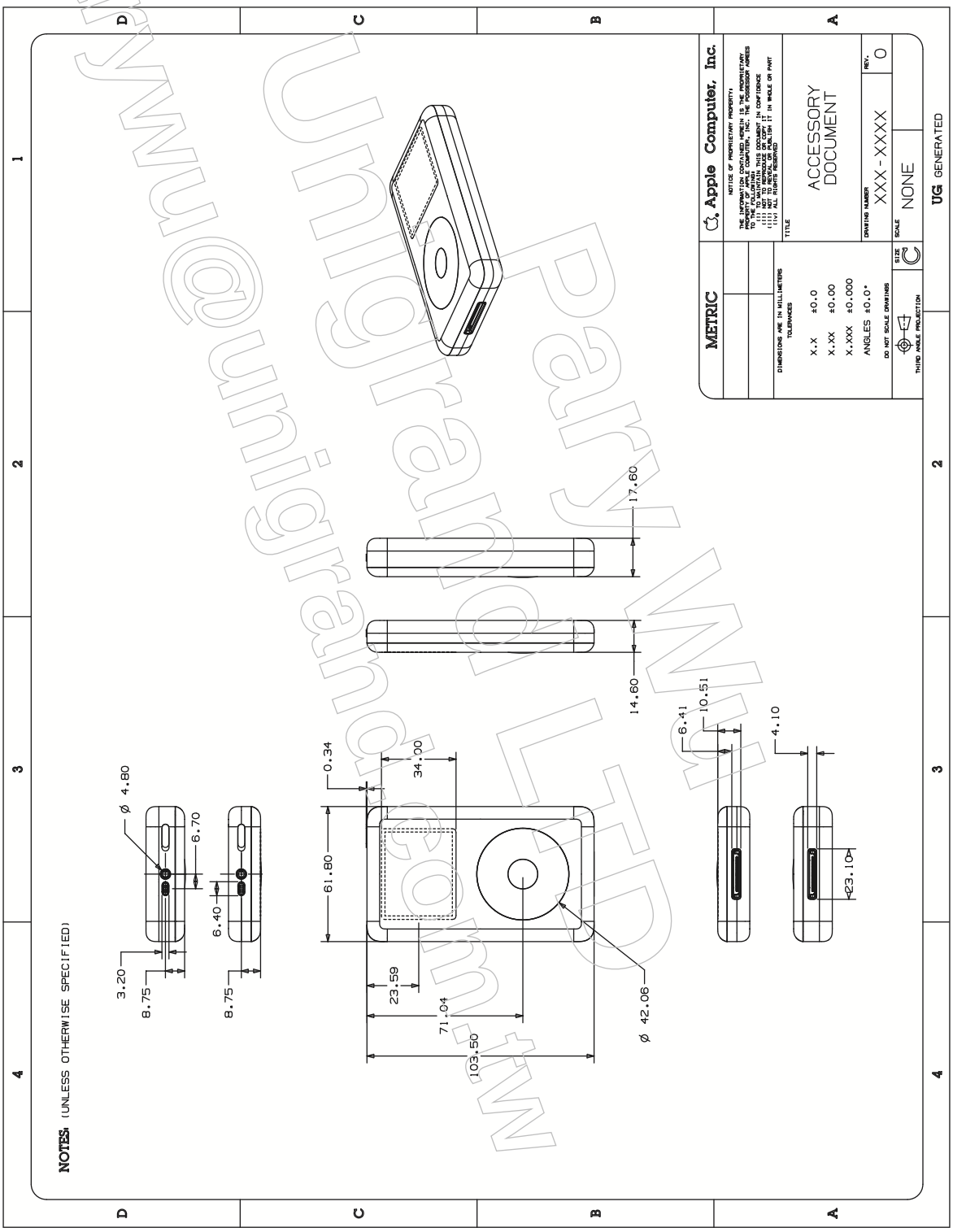
Figure 54-42 iPod 5th gen. 30GB Dimensional Drawing



Party Unigrand LTD
Party@unigrand.com.tw

54.43 iPod (4th generation)

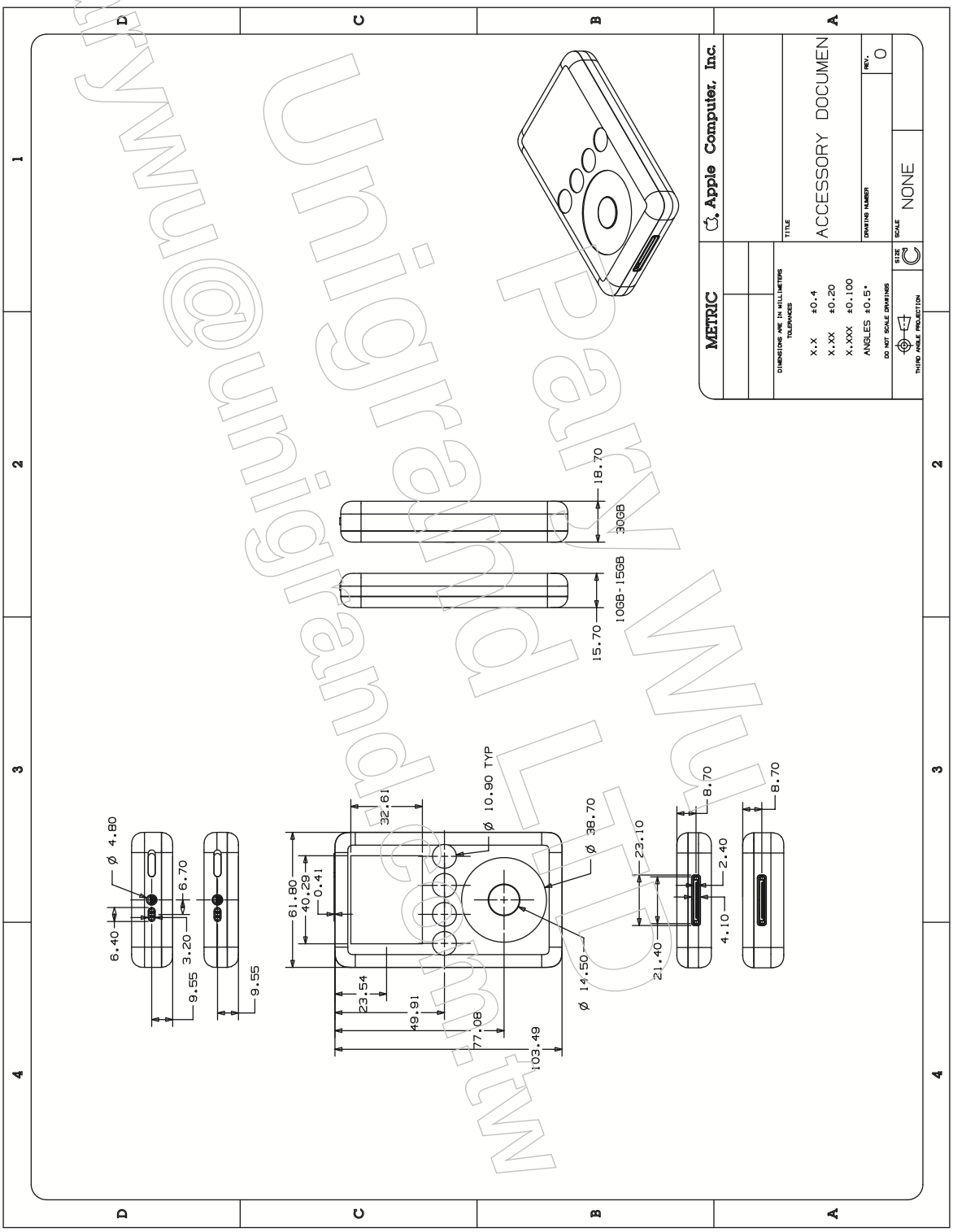
Figure 54-43 iPod 4th gen. Dimensional Drawing



Party Wall Unigrand Ltd
partywall@unigrandteam.com

54.44 iPod (3rd generation)

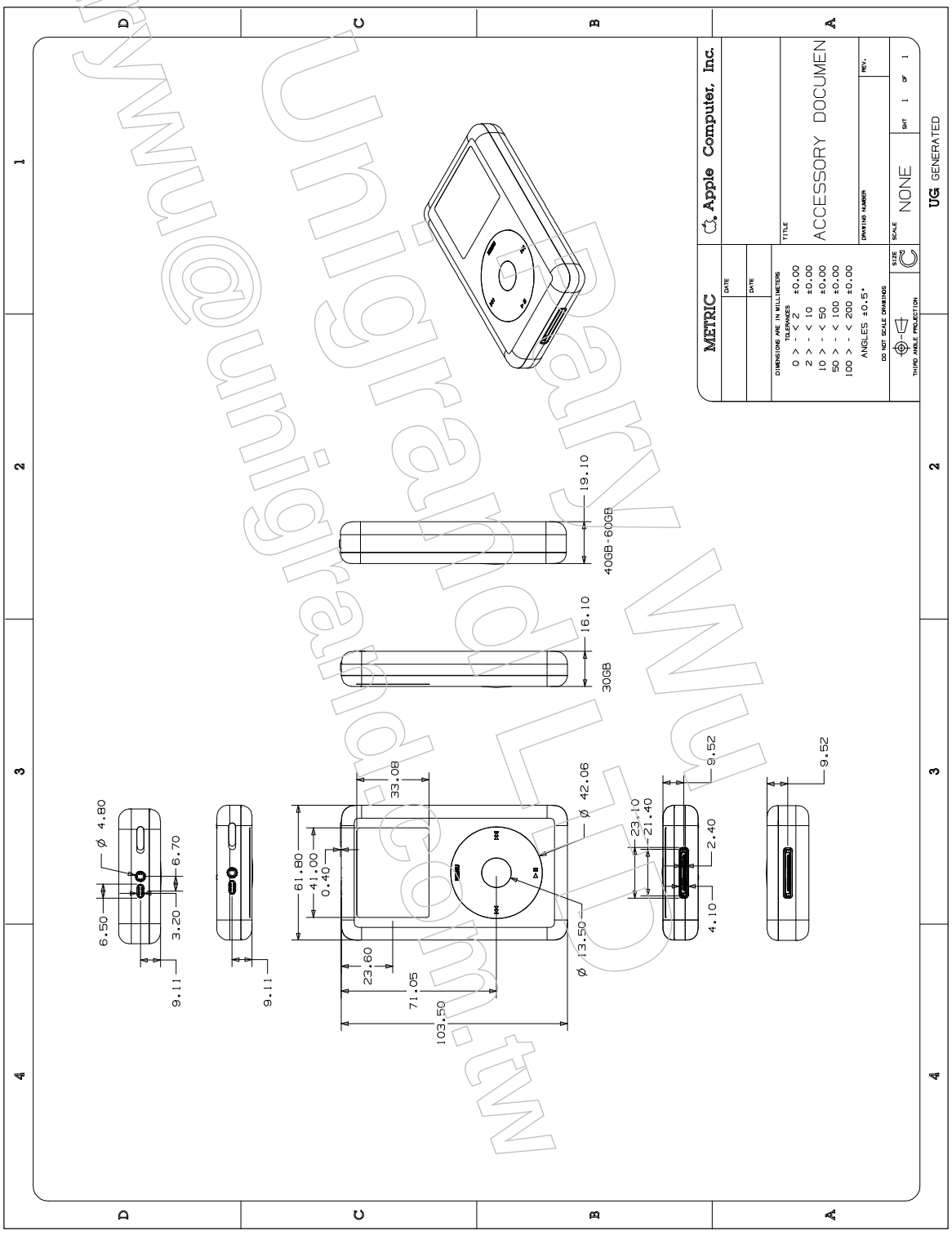
Figure 54-44 iPod 3rd gen. Dimensional Drawing



Partywu@unigrandltd.com Partywu Unigrand Ltd

54.45 iPod photo 30GB/60GB

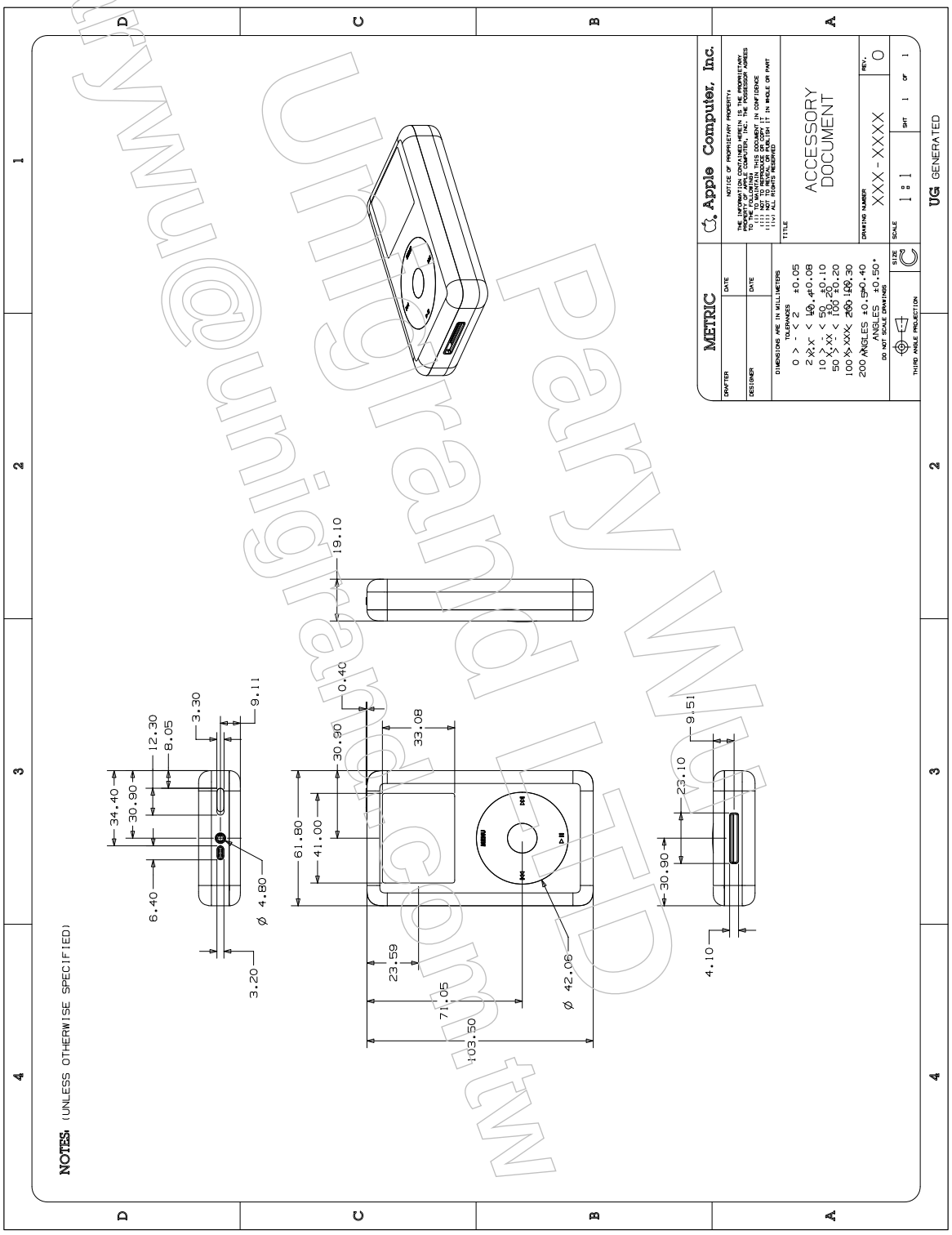
Figure 54-45 iPod photo 30/60GB Dimensional Drawing



Party Wu
partywu@unigrandlead.com.tw
Unigrandlead Co.,Ltd.

54.46 iPod photo

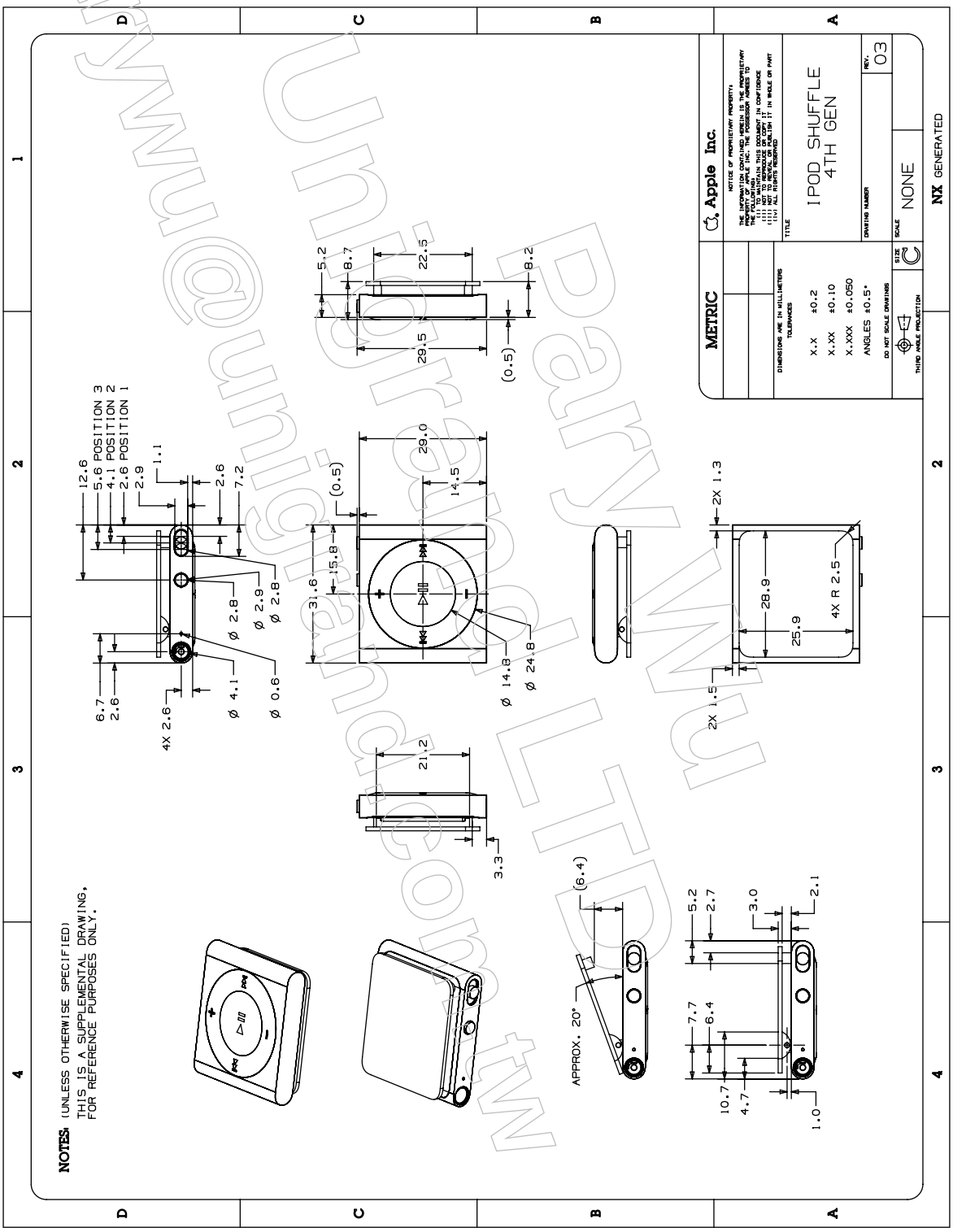
Figure 54-46 iPod photo Dimensional Drawing



Pary Wu
Unigrand Ltd
Pary@unigrand.com

54.47 iPod shuffle (4th generation)

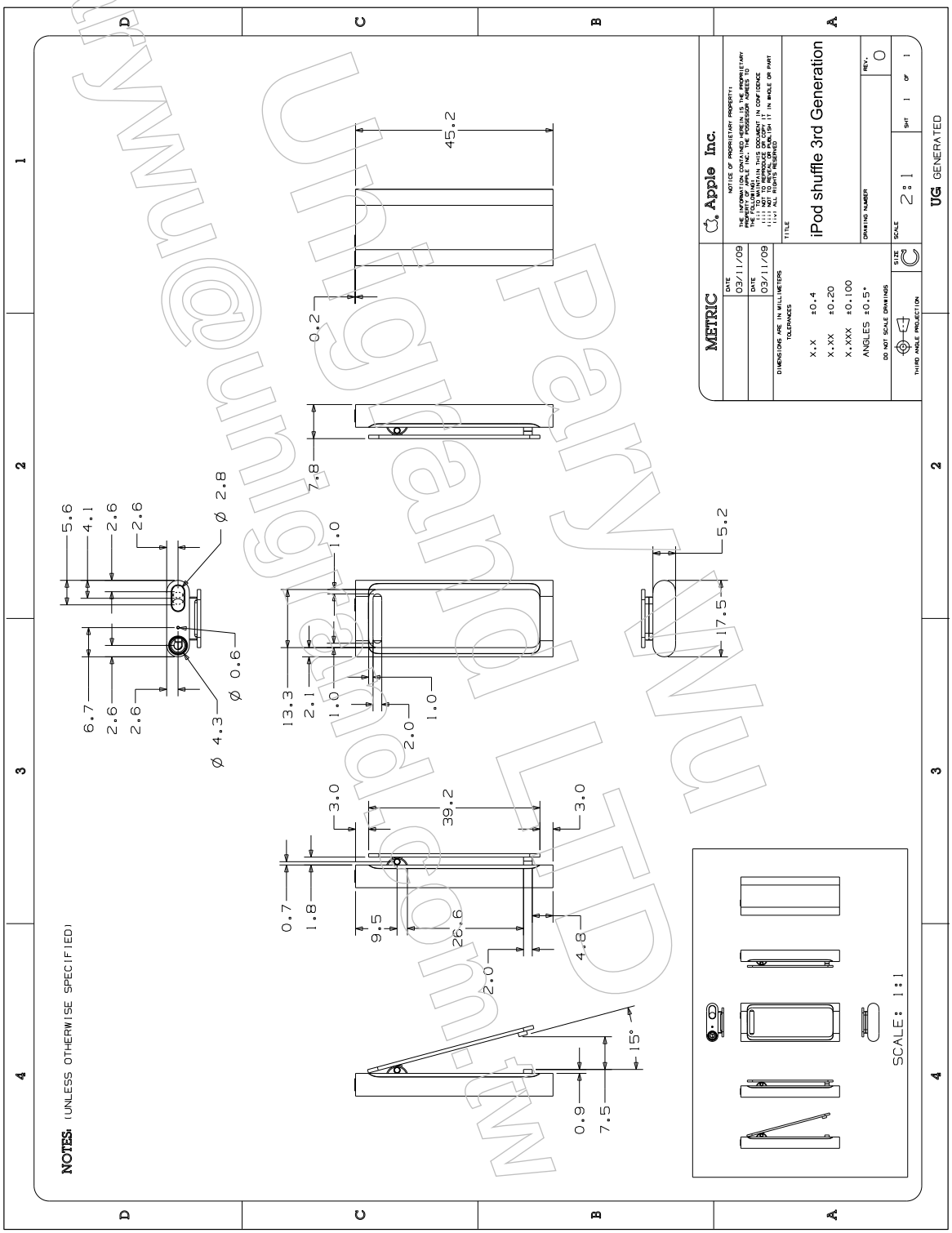
Figure 54-47 iPod shuffle 4th gen. Dimensional Drawing



Party Wu
Unterprise LTD
party@unterprise.com.tw

54.48 iPod shuffle (3rd generation)

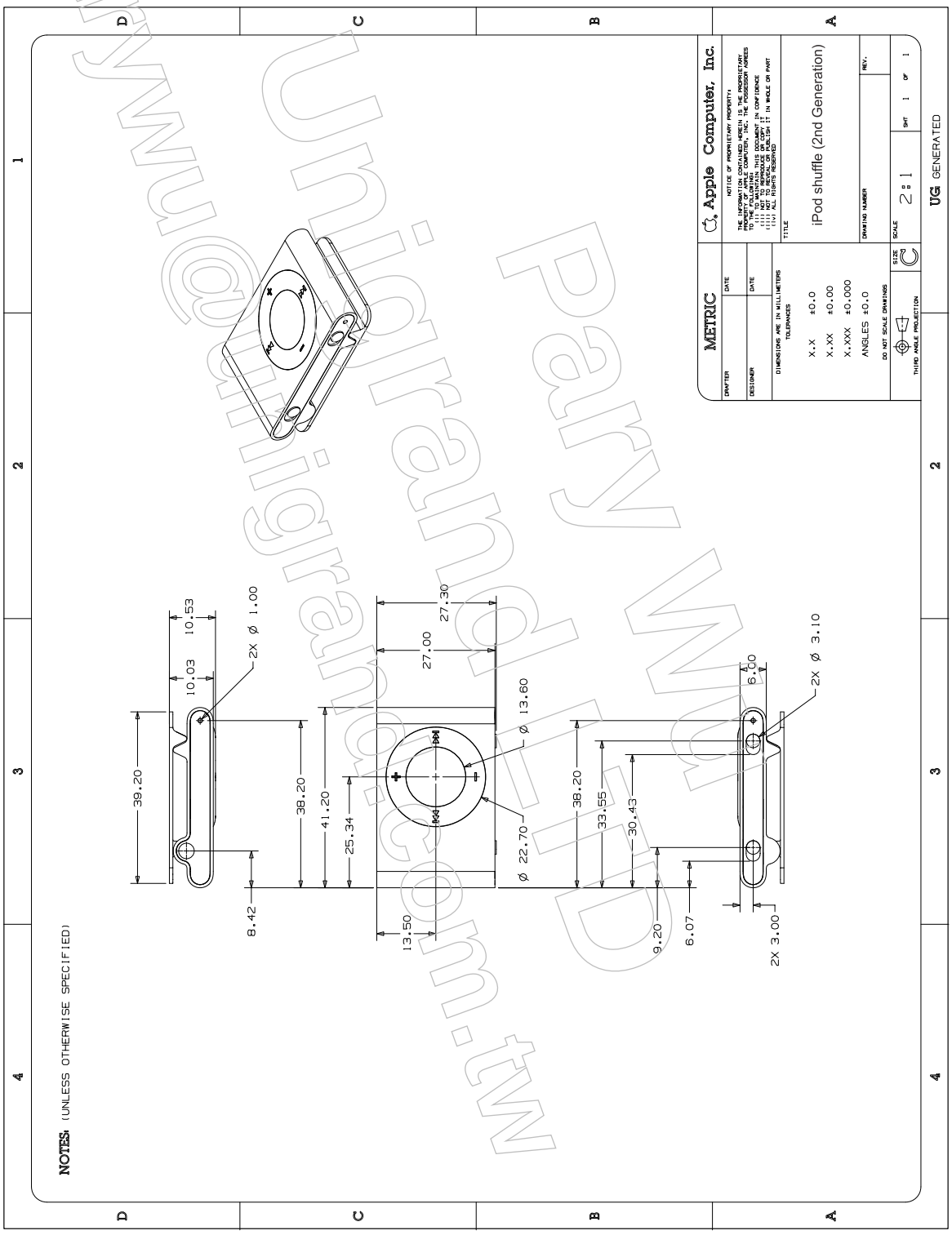
Figure 54-48 iPod shuffle 3rd gen. Dimensional Drawing



Partywu@unionland.com.tw Unigrand LTD

54.49 iPod shuffle (2nd generation)

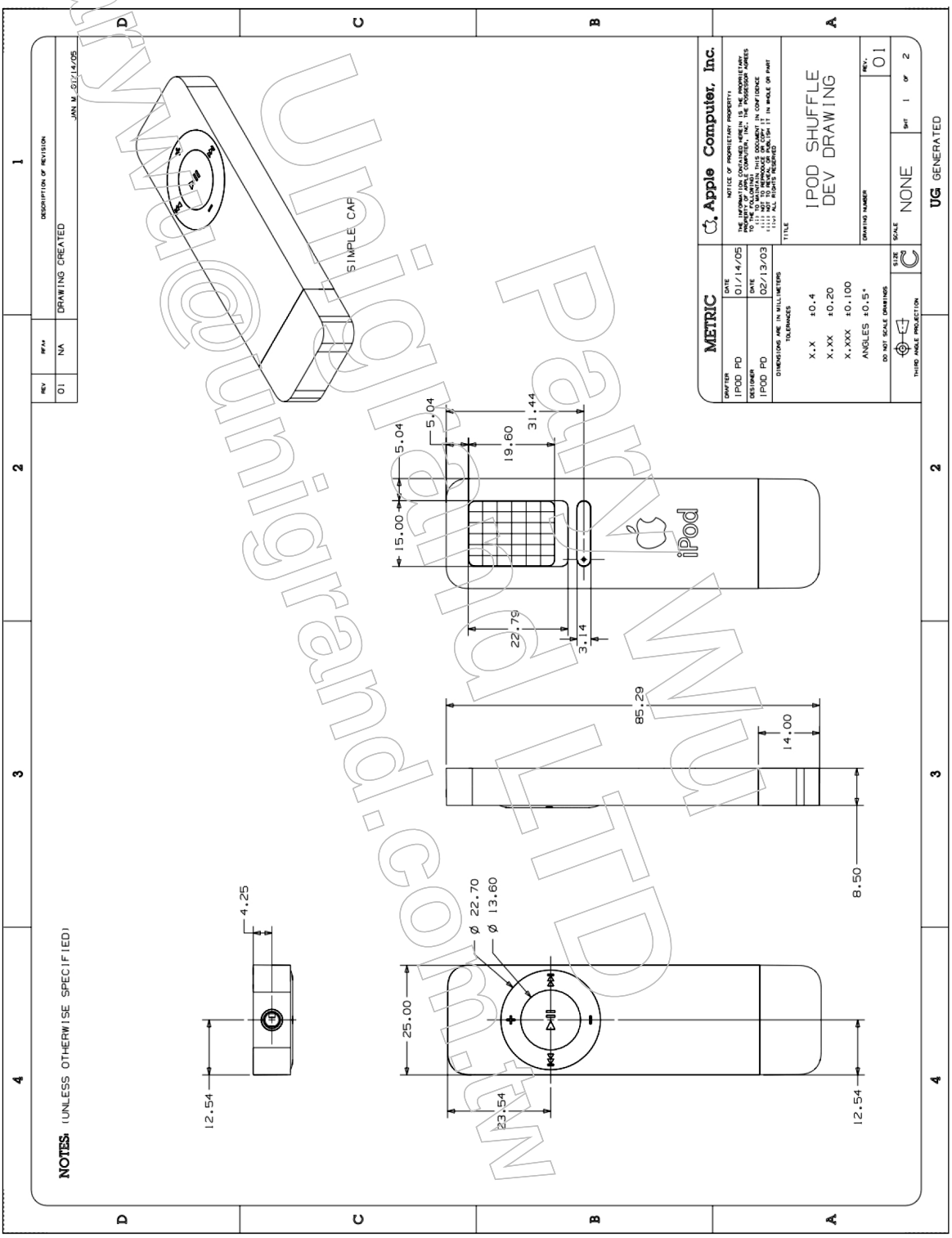
Figure 54-49 iPod shuffle 2nd gen. Dimensional Drawing



Party Wu Unigrandi
partywuunigrandi.com.tw

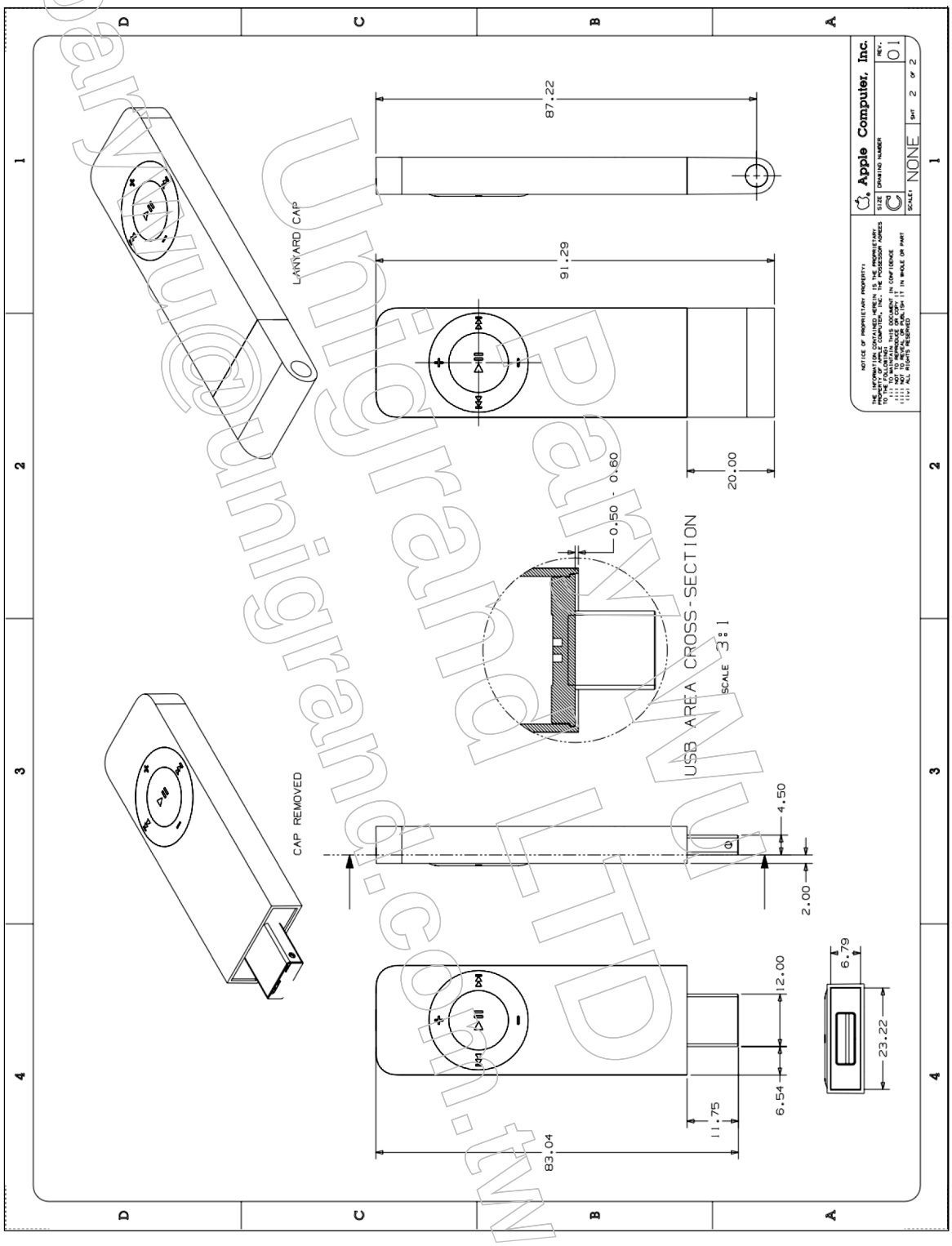
54.50 iPod shuffle

Figure 54-50 iPod shuffle Dimensional Drawing (1 of 2)



Patent Pending Unigrand.com

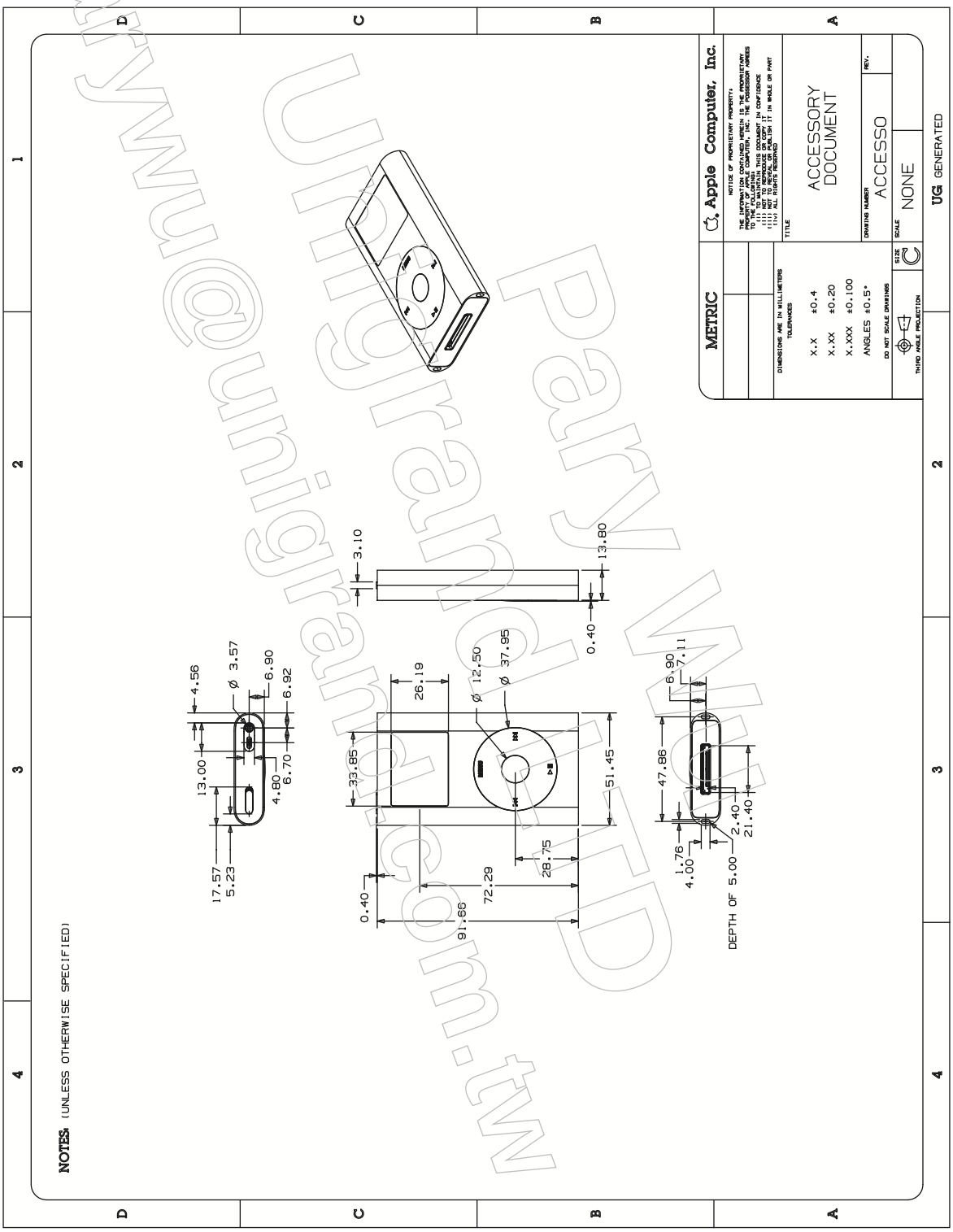
Figure 54-51 iPod shuffle Dimensional Drawing (2 of 2)



partner@unigrand.com.tw
Unigrand LTD

54.51 iPod mini

Figure 54-52 iPod mini Dimensional Drawing



Party Unigrandtech
party@unigrandtech.com.tw

55. Revision History

This table describes the changes to the MFi Accessory Interface Specification.

Date	Notes
2014-11-24	*Release R18*
	Updated Reference Designs & Development Kits (page 44)
	Updated iAP2 (page 45)
	Added Apple USB Power Adapters (page 48)
	Updated Apple Lightning Audio Module (page 80)
	Updated Apple Lightning Connector (page 114)
	Added Apple Lightning Receptacle (page 172)
	Added Apple Lightning Receptacle Controller (page 180)
	Updated Accessory Authentication over iAP2 Usage (page 202)
	Updated Accessory Identification Requirements (page 205)
	Updated Sniff Mode for Low Power Consumption (page 216)
	Updated Notifications (page 222)
	Added Play/Pause Button (page 223)
	Added CarPlay (page 244)
	Updated Device Protection (page 337)
	Updated Color (page 339)
	Updated Environmental (page 342)
	Updated Test Procedures (page 343)
	Updated Multiple Audio Connections (page 366)

Unofficial Document

	Updated Product Design (page 404)
	Updated HID (Human Interface Device) (page 407)
	Updated HID Game Controller Requirements (page 416)
	Updated Joystick (page 428)
	Updated App Match for Controller-Enabled Games (page 430)
	Updated HID Keyboard Requirements (page 446)
	Updated HID Media Playback Remote Requirements (page 456)
	Updated Media Library Playback Requirements (page 473)
	Updated Low Power Mode Requirements (page 487)
	Updated Siri Eyes Free Mode (page 500)
	Updated Improving Voice Recognition (page 501)
	Added User Interaction with Siri Eyes Free in a Vehicle (page 504)
	Added Test Procedures (page 507)
	Updated USB Role Switch Usage (page 529)
	Added Vehicle Status (page 532)
	Updated Control Session (page 562)
	Updated Accessory Identification (page 579)
	Updated Vehicle Status (page 625)
	Updated Location (page 601)
	Updated Device Dimensional Drawings (page 636)
2014-09-19	*Release R17*
	Added Reference Designs & Development Kits (page 44)
	Updated Cables with USB Connectors (page 50)
	Updated Cables with Non-USB Connectors (page 51)
	Updated RF Transmission and Reception (page 52)

Paid Content by WuTID
Unauthorized Distribution Prohibited

Updated Connector Versions (page 115)
Updated Connector Pad/Pin Configuration (page 125)
Updated All Accessories (page 131)
Updated Cable Accessories (page 133)
Updated Lightning (C48)/Lightning (C68) Cables (page 136)
Updated Dock Accessories (page 144)
Updated Lightning (C48)/Lightning (C68) Docks (page 149)
Updated Dongle Accessories (page 153)
Updated Form-Fitting Accessories (page 153)
Updated Test Procedures (page 155)
Updated Apple Lightning Audio Module (page 80)
Updated Cable (page 88)
Updated accConfigurationInformation (0x03) (page 98)
Updated Accessory Identification of Power Capabilities (page 207)
Updated App Launch Usage (page 211)
Updated External Accessory Protocol (page 213)
Updated Cases (page 336)
Updated Access to the Headphone Jack and 30-pin or Lightning Connector (page 336)
Added Device Protection (page 337)
Added Cover Glass Contact (page 337)
Added Dock Compatibility (page 337)
Updated Magnetic Interference (page 338)
Updated Touch ID Sensor (page 338)
Updated Environmental (page 342)
Added Near Field Communication (NFC) (page 343)

Paid@Pay Wu[ID]
Paid@Pay Wu[ID]

Updated Apple Device Compatibility (page 343)
Updated Apple Device Configurations (page 346)
Updated Device Notifications (page 365)
Updated Digital Audio Requirements (page 366)
Moved Readiness for Audio Streaming (page 366)
Moved Multiple Audio Connections (page 366)
Moved Audio Input Source Switching (page 367)
Moved Copy Protection of Digital Audio Output (page 367)
Moved USB Host Mode Audio (page 367)
Moved USB Device Mode Audio (page 369)
Updated iAP2 EA Session Requirements (page 373)
Updated Test Procedures (page 379)
Updated Headphone Jack (page 382)
Updated Test Procedures (page 386)
Updated Button Detection Circuitry Adjustments (page 403)
Updated Test Procedures (page 404)
Updated App Match for Controller-Enabled Games (page 430)
Updated HID Keyboard Requirements (page 446)
Updated Standard NMEA Sentences (page 462)
Updated Media Library Access (page 472)
Updated Power (page 480)
Moved USB Host Mode Passthroughs (page 514)
Moved USB Signal Integrity (page 514)
Updated USB Embedded Host Implementation (page 522)
Updated Accessory Identification (page 579)

2014-11-24 | Copyright © 2014 Apple Inc. All Rights Reserved.

	Updated App Launch (page 590)
	Updated Device Notifications (page 596)
	Updated External Accessory Protocol (page 597)
	Updated Media Library Access (page 602)
	Updated Now Playing (page 612)
	Updated Power (page 617)
	Updated Device Dimensional Drawings (page 636): Added dimensional drawings for iPhone 6 and iPhone 6 Plus
2014-07-17	*Release R16*
	Updated Accessory, Device, and Product (page 41)
	Updated Multiple Simultaneous iAP2 Connections (page 49)
	Updated Physical Configuration (page 71)
	Updated Connector Pad/Pin Configuration (page 125)
	Updated Apple Lightning Audio Module (page 80)
	Updated Audio Routing (page 226)
	Updated Headphone Jack (page 382)
	Updated HID Requirements (page 407)
	Updated HID Game Controller Requirements (page 416)
	Updated Switches and Displacement Triggers (page 423)
	Updated Bluetooth Button (page 429)
	Added Bluetooth Button (page 438)
	Updated HID Keyboard Requirements (page 446)
	Updated Location Information (page 460)
	Updated Power Usage (page 488)
	Updated Improving Voice Recognition (page 501)

Paid Content Protected

	Updated Table 43-1 (page 517)
	Updated Cumulative Acknowledgement Timeout (page 543)
	Updated Maximum Cumulative Acknowledgements (page 544)
	Updated Sleep (page 545)
	Updated Synchronization (page 547)
	Updated TransferIdentifier (page 568)
	Updated Media Library Access (page 602)
	Updated Now Playing (page 612)
	Updated Telephony (page 621)
2014-05-09	*Release R15*
	Removed "Accessory Test Procedures" and distributed its contents across feature chapters
	Removed "Transports"
	Updated Apple Device Detection (page 48)
	Updated Connector Versions (page 115)
	Updated Connector Pad/Pin Configuration (page 125)
	Added App Match (page 212)
	Added iAP2 (page 229)
	Updated External Accessory Protocol (page 372)
	Updated HID Game Controller (page 416)
	Updated HID Keyboard (page 446)
	Updated Location Information (page 460)
	Added Test Procedures (page 490)
	Added Serial (page 494)
	Added USB (page 514)
	Added USB Device Mode (page 522)

Paywu Digital LTD
paywu@digitalpaywu.com.tw

	Added USB Host Mode (page 527)
	Updated VoiceOver Usage (page 535)
	Updated iAP2 Link (page 538)
2014-04-07	*Release R14*
	Updated Direct User Action (page 43)
	Updated USB Signal Integrity (page 514)
	Updated Cables with USB Connectors (page 50)
	Updated Cables with Non-USB Connectors (page 51)
	Updated Integrated Non-USB Receptacles (page 51)
	Added User Supplied Cables and Power Supplies (page 52)
	Updated Connector Pad/Pin Configuration (page 125)
	Updated Shielding (page 137)
	Added Apple Lightning Audio Module (page 80)
	Added Headphones (page 378)
	Updated Headphone Remote and Mic (page 387)
	Updated Joystick (page 428)
	Updated Bluetooth Button (page 429)
	Added App Match for Controller-Enabled Games (page 430)
	Added HID Headphone Remote (page 441)
	Updated HID Keyboard Requirements (page 446)
	Updated Media Library Access Requirements (page 472)
	Updated Media Library Updates Usage (page 474)
	Updated USB Device Mode (page 522)
	Updated Media Library Access (page 602)
	Updated Telephony (page 621)

Paid Version Unregistered Version

	Updated Test Procedures (page 155)
	Added Test Procedures (page 112)
	Added Test Procedures (page 343)
	Added Test Procedures (page 379)
	Added Test Procedures (page 454)
2014-03-06	*Release R13*
	Updated Connector Pad/Pin Configuration (page 125)
	Updated HID Game Controller Requirements (page 416)
	Updated USB Role Switch Requirements (page 529)
	Updated Signal Integrity Tests (C10A, C10B, C48A, C48B, C68A, C68B) (page 159)
	Added DC Resistance Tests (C10, C48, C68) (page 161)
	Added Test Procedures (page 404)
	Updated Test Procedures (page 435)
2014-02-04	*Release R12*
	Added Adapters and Proxies (page 45)
	Updated Cables with USB Connectors (page 50)
	Updated Cables with Non-USB Connectors (page 51)
	Updated Integrated Non-USB Receptacles (page 51)
	Updated Connector Versions (page 115)
	Updated Lightning (C12) dimensions (page ?)
	Updated Connector Mechanical Requirements (page 131)
	Updated Shielding (page 137)
	Updated Lightning (C48)/Lightning (C68) Accessory Testing Requirements (page 155)
	Added Apple 30-pin Connector (page 185)
	Updated Cases (page 336)

Paid Unofficially by Wu Weiqiang (weiqiang@ti.com)

	Updated External Accessory Protocol Requirements (page 372)
	Updated Headphone Remote and Mic Requirements (page 387)
	Updated HID Keyboard Requirements (page 446)
	Updated Power Requirements (page 480)
	Added "Accessory Non-Power Source Usage"
	Added Providing Power to the Apple Device (page 488)
	Updated Bluetooth (page 216)
	Updated Human Interface Device (page 599)
	Updated Now Playing (page 612)
	Added Accessory Test System (page 635)
	Updated Over Voltage Protection Test (page 156)
2013-12-20	*Release R11*
	Added iBeacon (page 50)
	Updated Lightning (C12) dimensions (page ?)
	Updated Lightning (C48) dimensions (page ?)
	Updated iPad mini Form-Fitting Extended Gamepad Sample (page ?)
	Updated HID extended gamepad control layout (page ?)
	Updated Global Navigation Satellite System (GNSS) Mode (page 461)
	Updated Location Information Examples (page 469)
	Updated Flow Control (page 550)
	Updated iPad 2 with Wi-Fi + 3G (page 661)
2013-12-11	*Release R10*
	Updated Cables with USB Connectors (page 50)
	Updated Cable Encapsulation (page 135)
	Updated Shielding (page 137)

Paid@Pay Wu Pingan(Ningbo)Co.,Ltd

	Updated Form-Fitting Accessories (page 153)
	Updated Headphone Jack (page 382)
	Updated Cable Accessories (page 159)
	Release R9
	Updated Applicability (page 41)
	Added Development Tools and Emulators (page 44)
	Updated Cables with Non-USB Connectors (page 51)
	Moved Apple Authentication Coprocessor 2.0C (page 54)
	Updated App Launch Requirements (page 210)
	Updated Connector Versions (page 115)
	Updated Connector Pad/Pin Configuration (page 125)
	Added Lightning (C48)/Lightning (C68) Cables (page 136)
	Updated EA Native Transport (USB Host Mode) Requirements (page 373)
2013-11-01	Updated HID Requirements (page 407)
	Renamed HID Assistive Switch Control (page 410)
	Updated Shoulder Button (page 427)
	Updated Bluetooth Button (page 429)
	Added Test Procedures (page 435)
	Updated All Accessories (page 155)
	Added Cable Accessories (page 159)
	Added iPad mini 2 & 3 Wi-Fi (page 650)
	Added iPad mini 2 & 3 Wi-Fi + Cellular (page 651)
	Added iPad Air Wi-Fi (page 652)
	Added iPad Air Wi-Fi + Cellular (page 653)
2013-09-17	*Release R8*

Paid Content Wu Electronics LTD wuelectronics.com

Updated Applicability (page 41)
Updated Requirements, Recommendations, and Permissions (page 40)
Updated Detachable Connector Assemblies (page 45)
Updated Multiple Simultaneous iAP2 Connections (page 49)
Added Removable Storage (page 52)
Updated Cables with Non-USB Connectors (page 51)
Added Integrated USB Receptacles (page 51)
Updated Connector Versions (page 115)
Added Table 5-1 (page 116)
Added Accessory Identification of Manufacturing Information (page 206)
Updated AssistiveTouch (page 214)
Added Bluetooth (page 216)
Added Bluetooth Accessory Identification (page 233)
Added Bluetooth Headset Battery Level Indication (page 237)
Added Cases (page 336)
Added Device Notifications (page 365)
Updated External Accessory Protocol (page 372)
Updated Headphone Jack (page 382)
Added Extension Cables (page 383)
Updated Headphone Remote and Mic (page 387)
Updated HID Requirements (page 407)
Added HID Game Controller (page 416)
Updated Table 33-4 (page 451)
Updated Keyboard Example HID Report Descriptor (page 452)
Added HID Assistive Switch Control (page 410)

Paid@Digi-Key.com Purchased by Wu Digi-Key Corporation

	Updated Location Information (page 460)
	Updated Now Playing Updates Usage (page 479)
	Updated Power (page 480)
	Added Siri (page 496)
	Added Telephony (page 512)
	Updated Table 46-1 (page 530)
	Updated Serial (page 494)
	Updated USB Host Mode (page 527)
	Updated Extended Acknowledgement Payload (page 544)
	Updated None (page 565)
	Updated iAP2 Control Session Messages (page 577)
	Updated Coprocessor 2.0C Overview (page 54)
	Added Device Dimensional Drawings (page 636)
2013-07-26	*Release R7*
	Updated Cables with USB Connectors (page 50)
	Updated Connector Versions (page 115)
	Updated Dongle Accessories (page 153)
	Updated Headphone Remote and Mic (page 387)
2013-06-04	*Release R6*
	Update Introduction (page 40)
	Updated General Requirements and Recommendations (page 44)
	Updated Accessory Identification (page 205)
	Updated Apple Lightning Connector (page 114)
	Updated Digital Audio (page 366)
	Updated External Accessory Protocol (page 372)

Patent Pending Wu Electronics LTD info@elinkuk.com

	Updated Headphone Jack (page 382)
	Moved material from HID (Human Interface Device) (page 407) chapter into new chapters HID AssistiveTouch Pointer (page 413), HID Keyboard (page 446), and HID Media Playback Remote (page 456).
	Updated Location Information (page 460)
	Updated Media Library Access (page 472)
	Updated Power (page 480)
	Updated "Transports"
	Updated iAP2 Link (page 538)
	Updated iAP2 Sessions (page 561)
	Updated iAP2 Control Session Messages (page 577)
2013-02-20	*Release R5*
	Updated General Requirements and Recommendations (page 44)
	Updated Apple Lightning Connector (page 114)
	Updated Accessory Authentication (page 201)
	Updated Accessory Identification (page 205)
	Updated Device Authentication (page 362)
	Updated Digital Audio (page 366)
	Updated HID (Human Interface Device) (page 407)
	Updated Location Information (page 460)
	Updated Media Library Access (page 472)
	Updated Power (page 480)
	Updated USB Role Switch (page 529)
	Updated "Transports"
	Updated iAP2 Link (page 538)
	Updated iAP2 Sessions (page 561)

Apple Confidential
Apple iPod Accessory Protocol Interface Specification (Part Number 4359)
Apple Inc.

	Updated "Accessory Test Procedures"
2012-12-21	*Release R4*
	Updated General Requirements and Recommendations (page 44)
	Updated Apple Lightning Connector (page 114)
	Updated Media Library Access (page 472)
	Updated Now Playing Updates (page 478)
	Updated Power (page 480)
	Updated iAP2 Sessions (page 561)
2012-11-26	*Release R3*
	Added "Accessory Test Procedures"
	Updated General Requirements and Recommendations (page 44)
	Updated Accessory Authentication (page 201)
	Updated Device Authentication (page 362)
	Updated Headphone Jack (page 382)
	Updated Power (page 480)
2012-10-30	*Release R2*
	Changed specification title
	Added Apple Authentication Coprocessor 2.0C (page 54)
	Updated Apple Lightning Connector (page 114)
	Updated Accessory Authentication (page 201)
	Updated HID (Human Interface Device) (page 407)
	Updated External Accessory Protocol (page 372)
	Updated Power (page 480)
	Updated iAP2 Link (page 538)
	Updated iAP2 Sessions (page 561)

Paid Wu Electronics LTD
paid@electronics.com

2012-09-23	*Release R1*

Pary Wu
Unigrand LTD
paywu@unigrand.com.tw

Apple Inc.
Copyright © 2014 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Offline copy. Trademarks go here.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Pary Wu
Unigrand LTD
unigrand.com.tw
Reserved