# MFi Accessory Hardware Specification

**Release R9**

**2012-02-21**

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

# Contents


Introduction **Introduction 11**

Organization of This Document 11
Apple Device Models 12
General Specification Terms 18
Special Terminology 18
See Also 18

Chapter 1 **Hardware Interfaces 19**

30-Pin Connector 19
Headphone/Microphone Jack 21
Wireless Standards 23
    iPhone Protocols 23
    iPod touch Protocols 23
    iPad Protocols 23

Chapter 2 **Functional Hardware Description 25**

Apple Device Features and Capabilities 25
30-Pin Connector Functions 29
    USB 2.0 29
    Accessory Power 35
    Accessory Detect and Identify 36
    Detecting an Apple Device 37
    UART iPod Accessory Protocol Communication 38
    Apple Device Video Signal Levels 38
    Line Level Input 39
    Line Level Output 41
    Overall Grounding Requirements 42
    Minimizing Crosstalk and Noise 42
Apple Device Power States 43
    Sleep State 45
    Hibernate State 46
    Power Off State 47
Accessory Power Policy 47
    Low Power Mode 47
    Intermittent High Power Mode 48
    Entering Intermittent High Power Mode 48
    Exiting Intermittent High Power Mode 48

Chapter 3 **Command and Data Transports 51**

UART 51
USB 52
  USB Device Mode 52
  USB Host Mode 58
Bluetooth 61
  General Requirements 61
  Setting Up Bluetooth Communication for iAP 62
  Using Bluetooth Transport for iAP 62

Chapter 4 **Accessory Design Requirements 65**

General Requirements 65
  Audio Output From Accessories 65
  Video Output From Accessories 65
  Power Requirements 65
  Car Charger Design Requirements 66
  Battery Pack Design Requirements 66
  Magnetic Fields From Accessories 66
  USB Cables 66
  Cases for Apple Devices 66
iPhone-Specific Requirements 66
  Certification of iPhone Accessories 67
  RF Transmission and Reception Factors 67
  TDMA Noise Factors 67
  Speaker System Design Requirements 67
iPod Out Accessory Requirements 68
AssistiveTouch Accessory Requirements 68
Wi-Fi Network Login Sharing Requirements for Accessories 69

Chapter 5 **Headphone Remote and Mic System 71**

Transmitter Chip 71
  Overview 71
  Part Numbers 72
  Pin Assignments 72
  Maximum Voltage and Current Ratings 74
  Thermal Impedance 74
  Moisture Sensitivity 74
  Electrical Characteristics 74
  Theory of Operation 76
Button Detection Circuitry 80
  Circuit Adjustments 83

Appendix A **iOS Device Power Supply Requirements 85**

Converter Switching Frequency Guidelines 85
Noise Reduction Using a YCAP AC Capacitor 85
Impedance Stability of the Diode Bridge 85

Appendix B **Sample Accessory Circuits 87**

Internal Audio Circuits in Apple Devices 87
Verifying an Accessory's Audio/Video Output Design 89
Sample 1: A Passive Dock Accessory 89
Sample 2: An Apple Device-Powered Accessory 91
Sample 3: A Self-Powered Accessory 94
Avoiding a Warning When a Self-Powered Accessory Is Off 96
Avoiding TX Back-Powering 96

Appendix C **FireWire to USB Reference Design 99**

Converter Requirements 99
Switching Frequency Issues 99
Other Design Issues 100
Typical Design 100

Appendix D **Interfacing With the 3G iPod 105**

Accessory Detection 105
Powering the 3G iPod 105
Connector Usage 105

Appendix E **DisplayPort Digital Video Connectivity 107**

Connection to the Apple Device 107
DisplayPort Output 109
Cable Connections 110

Appendix F **Historical Information 113**

Headphone Jack on the 4G iPod color and 5G iPod 113
38400/57600 BPS Serial Baud Rates 113
FireWire 114
Deprecated ID Resistors 114

**Document Revision History 117**

**Glossary 121**

# Figures and Tables


Introduction **Introduction 11**

| | | |
|---|---|---|
| Table I-1 | iPod, iPhone, and iPad products | 12 |

Chapter 1 **Hardware Interfaces 19**

| | | |
|---|---|---|
| Figure 1-1 | Typical headphone/microphone circuitry in an iPhone accessory | 21 |
| Figure 1-2 | Headphone plug dimensions | 22 |
| Table 1-1 | 30-pin connector pin assignments | 19 |
| Table 1-2 | Headphone plug connections | 22 |

Chapter 2 **Functional Hardware Description 25**

| | | |
|---|---|---|
| Figure 2-1 | D+ and D- connections for USB power sources | 30 |
| Figure 2-2 | Accessory power internal resistance | 35 |
| Figure 2-3 | Accessory Identify and Accessory Detect Signals | 36 |
| Figure 2-4 | Apple device detection | 38 |
| Figure 2-5 | Equivalent input circuits for the iPod classic, 4G nano, 2G touch, and 5G nano | 40 |
| Figure 2-6 | Equivalent input circuits for other Apple devices | 40 |
| Figure 2-7 | Equivalent output circuits for the iPod classic, 2G touch, 4G nano, 5G nano, and future models | 41 |
| Figure 2-8 | Equivalent output circuits for other Apple devices | 42 |
| Table 2-1 | iAP transports | 25 |
| Table 2-2 | 30-pin connector hardware features | 26 |
| Table 2-3 | Additional hardware features | 27 |
| Table 2-4 | USB Host Mode supported device classes | 28 |
| Table 2-5 | USB power supply component specifications | 31 |
| Table 2-6 | USB cable maximum DC resistances | 32 |
| Table 2-7 | Ways to communicate power availability | 33 |
| Table 2-8 | USB Device Vendor Request to set available current from accessory (USB Device Mode only) | 34 |
| Table 2-9 | USB device request timing | 34 |
| Table 2-10 | Resistor-based accessory functions | 37 |
| Table 2-11 | Video signal levels in volts peak-to-peak | 39 |
| Table 2-12 | Audio output line levels | 41 |
| Table 2-13 | Power states and transitions | 43 |
| Table 2-14 | Sleep times before hibernation | 45 |
| Table 2-15 | Maximum current draw in Low Power mode | 47 |
| Table 2-16 | iAP commands that permit Intermittent High Power mode | 48 |
| Table 2-17 | USB events that permit Intermittent High Power mode | 48 |
| Table 2-18 | iAP commands that cancel Intermittent High Power mode | 49 |
| Table 2-19 | USB events that cancel Intermittent High Power mode | 49 |

Chapter 3 **Command and Data Transports 51**

Figure 3-1 Configuration and interface descriptors for Apple devices without USB audio 54
Figure 3-2 Configuration and interface descriptors for Apple devices with USB audio 54
Figure 3-3 Apple device vendor-specific HID report 55
Figure 3-4 Possible report packing scenarios 57
Figure 3-5 Transferring `IdentifyDeviceLingoes` and `ACK` commands over USB 57
Table 3-1 Apple device mark and space levels 51
Table 3-2 Link control byte usage 56
Table 3-3 USB Device Enumeration for Apple devices 59
Table 3-4 Lingoes accessible through Bluetooth 63

Chapter 5 **Headphone Remote and Mic System 71**

Figure 5-1 Transmitter chip package 73
Figure 5-2 Transmitter block diagram 77
Figure 5-3 Startup timing 78
Figure 5-4 Tone mode ACK sequence 79
Figure 5-5 Tone transmit/decode method 80
Figure 5-6 Transmitter circuit 81
Figure 5-7 Microphone circuit 82
Table 5-1 Transmitter chip part numbers 72
Table 5-2 Pin Assignments 72
Table 5-3 Transmitter chip package dimensions 73
Table 5-4 Maximum voltage and current ratings 74
Table 5-5 Electrical characteristics 75
Table 5-6 DETECT Pin Voltage 77
Table 5-7 Circuit components 82

Appendix A **iOS Device Power Supply Requirements 85**

Figure A-1 Typical diode bridge circuit for an AC adapter 86

Appendix B **Sample Accessory Circuits 87**

Figure B-1 Apple device audio circuitry 87
Figure B-2 A passive Apple device dock 90
Figure B-3 An Apple device-powered accessory 92
Figure B-4 A self-powered accessory 94
Figure B-5 Selectively controlling accessory detect and identify 96
Figure B-6 TX line isolation circuit 97

Appendix C **FireWire to USB Reference Design 99**

Figure C-1 Main converter circuit 101

Figure C-2 USB power switching circuit 101
Figure C-3 Converter shut-down circuit 102
Figure C-4 Resistor divider network 102
Table C-1 Unwanted switching frequencies 100
Table C-2 Typical converter circuit components 102

Appendix E **DisplayPort Digital Video Connectivity 107**

Figure E-1 Apple device-to-accessory DisplayPort configurations 110
Table E-1 Repurposing of 30-pin connector pins for DisplayPort 107
Table E-2 Apple device to accessory DisplayPort cable 111
Table E-3 Apple device to accessory Mini DisplayPort cable 111

Appendix F **Historical Information 113**

Figure F-1 Pinout for the headphone jack on the 4G iPod color display and 5G iPod 113
Figure F-2 30-pin to FireWire cable 114
Table F-1 Firmware versions introducing 38400/57600 bps serial baud rates 113
Table F-2 Deprecated RID values 115

# Introduction



This specification covers the hardware requirements for third-party accessories that are designed to work with Apple iPods, iPhones, and iPads.

**IMPORTANT:** This document uses the term "Apple devices" to refer generically to iPods, iPhones, and iPads, all of which support the iPod Accessory Protocol (iAP) interface. Among these products, those that also run iOS (Apple's mobile operating system) are referred to as "iOS devices." Specifications in this document that are designated for iOS devices apply only to those products. Specifications designated for iPods apply only to Apple devices that are not iOS devices.

## Organization of This Document

The specifications in this document are arranged in the following chapters and appendixes:

- "Hardware Interfaces" (page 19) specifies the electrical and mechanical interfaces between accessories and Apple devices.
- "Functional Hardware Description" (page 25) describes the functional characteristics of Apple devices, plus their hardware interfaces.
- "Command and Data Transports" (page 51) describes the UART, USB, and Bluetooth transports for communication between accessories and Apple devices.
- "Accessory Design Requirements" (page 65) summarizes certain hardware design requirements that accessories must meet.
- "Headphone Remote and Mic System" (page 71) describes Apple's technology for sending button press information from a headset accessory to an Apple device through the headphone connector.
- "iOS Device Power Supply Requirements" (page 85) contains design guidelines for third-party developers of DC and AC adapter accessories for iOS devices.
- "Sample Accessory Circuits" (page 87) presents sample schematics for handling audio and video in accessories for Apple devices.
- "FireWire to USB Reference Design" (page 99) provides reference information for FireWire-to-USB DC-to-DC converter designs.

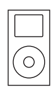
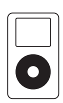
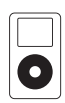
- "Interfacing With the 3G iPod" (page 105) summarizes some of the model-specific design requirements for 3G iPod support.
- "DisplayPort Digital Video Connectivity" (page 107) describes the DisplayPort output planned for future Apple devices.
- "Historical Information" (page 113) provides specifications for past Apple devices.

At the end of this document are a glossary of terms and a document revision history.

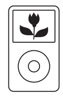
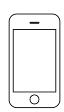
# Apple Device Models

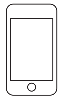
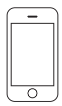
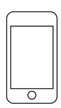
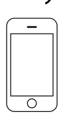
Table I-1 (page 12) lists the Apple devices covered by this specification.

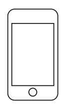
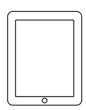
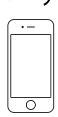
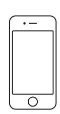
**Table I-1** iPod, iPhone, and iPad products

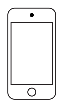
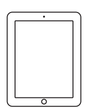
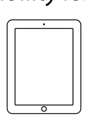
<table>
<tr><th colspan="3">iPod mini</th></tr>
<tr><td rowspan="3"></td><td>Product name: iPod mini</td><td>Shipped: 01/2004</td></tr>
<tr><td colspan="2">Compatibility icon:<br> iPod mini<br>4GB 6GB</td></tr>
<tr><td colspan="2">Connectivity: Dock connector, headphone jack.</td></tr>
<tr><th colspan="3">4G iPod</th></tr>
<tr><td rowspan="3"><br></td><td>Product names: iPod (4th generation), iPod photo, iPod photo (2nd generation), iPod 4th generation (color display)</td><td>Shipped: 07/2004, 10/2004, 02/2005, 06/2005</td></tr>
<tr><td colspan="2">Compatibility icons:<br> iPod<br>4th generation<br>20GB<br> iPod<br>4th generation<br>40GB<br> iPod<br>4th generation (color display)<br>20GB 30GB<br> iPod<br>4th generation (color display)<br>40GB 60GB</td></tr>
<tr><td colspan="2">Connectivity: Dock connector, headphone jack.</td></tr>
</table>

| | | |
|---|---|---|
| **iPod nano** | | |
|  | *Product name:* iPod nano | *Shipped:* 09/2005 |
| | *Compatibility icon:*<br>iPod nano<br>1st generation<br>1GB 2GB 4GB | |
| | *Connectivity:* Dock connector, headphone jack. | |
| **5G iPod** | | |
|  | *Product name:* iPod with video | *Shipped:* 10/2005 |
| | *Compatibility icons:*<br>iPod<br>5th generation (video)<br>30GB<br><br>iPod<br>5th generation (video)<br>60GB 80GB | |
| | *Connectivity:* Dock connector, headphone jack. | |
| **2G nano** | | |
|  | *Product name:* iPod nano (2nd generation) | *Shipped:* 09/2006 |
| | *Compatibility icon:*<br>iPod nano<br>2nd generation (aluminum)<br>2GB 4GB 8GB | |
| | *Connectivity:* Dock connector, headphone jack. | |
| **iPod classic** | | |
|  | *Product name:* iPod classic | *Shipped:* 09/2007 |
| | *Compatibility icons:*<br>iPod classic<br>80GB<br><br>iPod classic<br>160GB (2007) | |
| | *Connectivity:* Dock connector, headphone jack. | |

<table>
<tr><th colspan="3">120 GB classic, 160 GB classic</th></tr>
<tr><td rowspan="3"></td><td>Product names: iPod classic (120GB), iPod classic (160GB)</td><td>Shipped: 09/2008, 9/2009</td></tr>
<tr><td colspan="2">Compatibility icon:<br><br>iPod classic<br>160GB (2009)</td></tr>
<tr><td colspan="2">Connectivity: Dock connector, microphone/headphone jack.</td></tr>
<tr><th colspan="3">3G nano</th></tr>
<tr><td rowspan="3"></td><td>Product name: iPod nano (3rd generation)</td><td>Shipped: 09/2007</td></tr>
<tr><td colspan="2">Compatibility icon:<br><br>iPod nano<br>3rd generation (video)<br>4GB 8GB</td></tr>
<tr><td colspan="2">Connectivity: Dock connector, headphone jack.</td></tr>
<tr><th colspan="3">4G nano</th></tr>
<tr><td rowspan="3"></td><td>Product name: iPod nano (4th generation)</td><td>Shipped: 09/2008</td></tr>
<tr><td colspan="2">Compatibility icon:<br><br>iPod nano<br>4th generation (video)<br>8GB 16GB</td></tr>
<tr><td colspan="2">Connectivity: Dock connector, microphone/headphone jack.</td></tr>
<tr><th colspan="3">iPhone</th></tr>
<tr><td rowspan="3"></td><td>Product name: iPhone</td><td>Shipped: 06/2007</td></tr>
<tr><td colspan="2">Compatibility icon:<br><br>iPhone<br>4GB 8GB 16GB</td></tr>
<tr><td colspan="2">Connectivity: GSM, Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack. Sensors: Still camera.</td></tr>
</table>

| **iPod touch** | | |
|---|---|---|
|  | *Product name:* iPod touch | *Shipped:* 09/2007 |
| | *Compatibility icon:*<br> iPod touch<br>1st generation<br>8GB 16GB 32GB | |
| | *Connectivity:* Wi-Fi, multi-touch display, dock connector, headphone jack. | |
| **iPhone 3G** | | |
|  | *Product name:* iPhone 3G | *Shipped:* 06/2008 |
| | *Compatibility icon:*<br> iPhone 3G<br>8GB 16GB | |
| | *Connectivity:* GSM, Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack. *Sensors:* Still camera, assisted GPS. | |
| **2G touch** | | |
|  | *Product name:* iPod touch (2nd generation) | *Shipped:* 09/2008 |
| | *Compatibility icon:*<br> iPod touch<br>2nd generation<br>8GB 16GB 32GB | |
| | *Connectivity:* Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack. | |
| **iPhone 3GS** | | |
|  | *Product name:* iPhone 3GS | *Shipped:* 6/2009, 6/2010 |
| | *Compatibility icon:*<br> iPhone 3GS<br>8GB 16GB 32GB | |
| | *Connectivity:* GSM, Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack. *Sensors:* Still/video camera, assisted GPS, digital compass. | |

<table>
<tr><th colspan="3">5G nano</th></tr>
<tr><td rowspan="3"></td><td>Product name: iPod nano (5th generation)</td><td>Shipped: 09/2009</td></tr>
<tr><td colspan="2">Compatibility icon:<br><br>iPod nano<br>5th generation (video camera)<br>8GB 16GB</td></tr>
<tr><td colspan="2">Connectivity: Dock connector, microphone/headphone jack.</td></tr>
<tr><th colspan="3">3G touch</th></tr>
<tr><td rowspan="3"></td><td>Product name: iPod touch (3rd generation)</td><td>Shipped: 09/2009</td></tr>
<tr><td colspan="2">Compatibility icon:<br><br>iPod touch<br>3rd generation<br>32GB 64GB</td></tr>
<tr><td colspan="2">Connectivity: Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack.</td></tr>
<tr><th colspan="3">iPad</th></tr>
<tr><td rowspan="3"></td><td>Product names: iPad (Wi-Fi), iPad (Wi-Fi + 3G)</td><td>Shipped: 04/2010</td></tr>
<tr><td colspan="2">Compatibility icon:<br><br>iPad<br>16GB 32GB 64GB</td></tr>
<tr><td colspan="2">Connectivity: GSM (iPad 3G), Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack. Sensors: Assisted GPS, digital compass, accelerometer, ambient light sensor.</td></tr>
<tr><th colspan="3">iPhone 4</th></tr>
<tr><td rowspan="3"></td><td>Product names: iPhone 4 (GSM model), iPhone 4 (CDMA model), iPhone 4S</td><td>Shipped: 06/2010, 02/2011, 10/2011</td></tr>
<tr><td colspan="2">Compatibility icons:<br><br>iPhone 4<br>16GB 32GB<br><br>iPhone 4S<br>16GB 32GB 64GB</td></tr>
<tr><td colspan="2">Connectivity: CDMA or GSM (both in iPhone 4S), Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack. Sensors: Two still/video cameras, assisted GPS, digital compass, 3-axis gyroscope.</td></tr>
</table>

| 6G nano | | |
|---|---|---|
|  | *Product name:* iPod nano (6th generation) | *Shipped:* 09/2010 |
| | *Compatibility icon:*<br> **iPod nano**<br>6th generation<br>8GB 16GB | |
| | *Connectivity:* Multi-touch display, dock connector, microphone/headphone jack. | |
| **4G touch** | | |
|  | *Product name:* iPod touch (4th generation) | *Shipped:* 09/2010 |
| | *Compatibility icon:*<br> **iPod touch**<br>4th generation<br>8GB 32GB 64GB | |
| | *Connectivity:* Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack. *Sensors:* Two still/video cameras, 3-axis gyroscope. | |
| **iPad 2** | | |
|  | *Product names:* iPad 2 (Wi-Fi), iPad 2 (Wi-Fi + 3G) | *Shipped:* 03/2011 |
| | *Compatibility icon:*<br> **iPad 2**<br>16GB 32GB 64GB | |
| | *Connectivity:* CDMA or GSM (iPad 2 (Wi-Fi + 3G)), Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack. *Sensors:* Still camera, still/video camera, assisted GPS, digital compass, accelerometer, 3-axis gyroscope, ambient light sensor. | |
| **iPad (3rd generation)** | | |
|  | *Product names:* iPad with Wi-Fi (3rd generation), iPad with Wi-Fi + 4G (3rd generation) | *Shipped:* 03/2012 |
| | *Compatibility icon:*<br> **iPad** (3rd generation)<br>16GB 32GB 64GB | |
| | *Connectivity:* CDMA or GSM (iPad with Wi-Fi + 4G (3rd generation)), Wi-Fi, Bluetooth, multi-touch display, dock connector, microphone/headphone jack. *Sensors:* Still camera, still/video camera, assisted GPS, digital compass, accelerometer, 3-axis gyroscope, ambient light sensor. | |

# General Specification Terms

Parts of this document contain specification requirements that are incorporated by reference into legal agreements between Apple Inc. and its licensees. The use of the words "must," "should," and "may" in these specifications have the following meanings:

- "Must" means that the specification is an absolute requirement.
- "Must not" means that the specification is an absolute prohibition.
- "Should" means that there may be valid reasons in particular circumstances to ignore the specification, but their full implications must be understood and carefully weighed before choosing to do so.
- "Should not" means that there may be valid reasons in particular circumstances that make the specified action or feature acceptable, but their full implications must be understood and carefully weighed before choosing to include it.
- "May" means that the indicated action or feature does not contravene this specification.

# Special Terminology

Certain terms in this document have the following specific meanings:

- When a data field is marked "Reserved," accessories writing to it must set it to 0 (unless otherwise noted) and accessories reading it must ignore its value.
- "Deprecated" marks an earlier technology that is no longer permitted in new accessory designs.
- "USB Host Mode" is an operating mode in which an Apple device is a USB host and its attached accessory acts as a USB device.
- "USB device mode" is an operating mode in which an accessory is a USB host and its attached Apple device acts as a USB device.

# See Also

For further information, refer to the latest revisions of these additional documents:

- *IEEE 1394a Specification*
- *USB 2.0 High Speed Specification*
- *USB Device Class Definition for Audio Devices*
- *United States RBDS Standard, NRSC-4-A*
- *Bluetooth Core Specification 2.1*
- *VESA DisplayPort Standard*
- *VESA DisplayPort PHY Compliance Test Specification*
- *VESA DisplayPort Link Layer Compliance Test Specification*

# Hardware Interfaces

This chapter describes the connector signals, pin assignments, and wireless standards for the primary hardware interfaces to Apple devices.

## 30-Pin Connector

The 30-pin connector on the bottom end of the Apple device is the product's primary hardware interface. Table 1-1 (page 19) lists its pin assignments. Signal and ground connections to the pins must follow these rules:

- All four digital grounds (pins 1, 2, 15, and 16) must be connected together in the accessory if they are conducted by its cable. Cables that plug into the 30-pin connector should not latch, because latching can result in cable strain and damage to the Apple device.
- Some of these pins are reassigned in Apple products that support DisplayPort; see Table E-1 (page 107).
- Accessories must leave reserved pins unconnected (NC) unless otherwise specified.

**Table 1-1** 30-pin connector pin assignments

| Pin | Signal name | I/O | Function |
|---|---|---|---|
| 1 | DGND | GND | Digital ground in the Apple device and in the accessory. |
| 2 | DGND | GND | Digital ground in the Apple device and in the accessory. |
| 3 | TPA+ | I/O | FireWire signal; **Deprecated** |
| 4 | USB D+ | I/O | USB signal |
| 5 | TPA- | I/O | FireWire signal; **Deprecated** |
| 6 | USB D- | I/O | USB signal |
| 7 | TPB+ | I/O | FireWire signal; **Deprecated** |
| 8 | USB Vbus | I | USB power in. |
| 9 | TPB- | I/O | FireWire signal; **Deprecated** |
| 10 | Accessory Identify | I | See "Accessory Detect and Identify" (page 36) for details. |
| 11 | F/W PWR+ | I | FireWire and charger input power (8 V to 15 V DC) **Deprecated** |
| 12 | F/W PWR+ | I | FireWire and charger input power (8 V to 15 V DC) **Deprecated** |

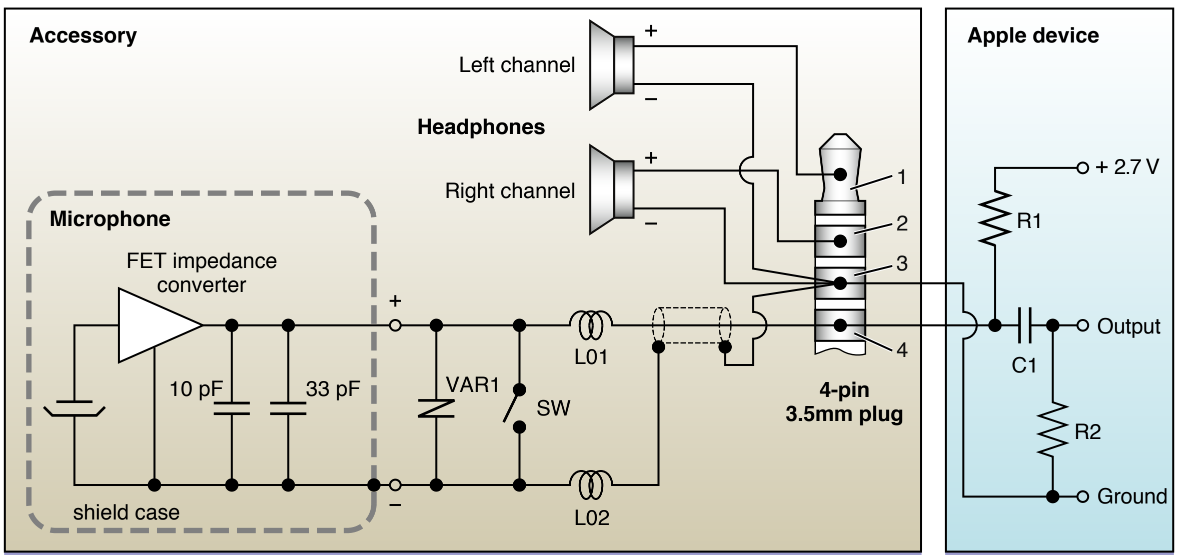
| Pin | Signal name | I/O | Function |
|---|---|---|---|
| 13 | Accessory Power | O | 3.3 V is the nominal output from the Apple device. Nominal current is 5 mA or less (low power mode), with current limited to 100 mA in high power mode. This output must not be used to detect the presence of an Apple device. |
| 14 | Reserved | | Pin must be left disconnected (floating). |
| 15 | DGND | GND | Digital ground in the Apple device and in the accessory. |
| 16 | DGND | GND | Digital ground in the Apple device and in the accessory. |
| 17 | Reserved | | Pin must be left disconnected (floating). |
| 18 | Apple Device RX | I | iPod Accessory Protocol (Data to the Apple device from the accessory) |
| 19 | Apple Device TX | O | iPod Accessory Protocol (Data from the Apple device to the accessory) |
| 20 | Accessory Detect | I | See "Accessory Detect and Identify" (page 36) for details. |
| 21 | S Video Y / Component Video Pr | O | Either the luminance signal of S Video or the (Pr) signal of component video. Only on Apple devices that support the appropriate video signal. |
| 22 | S Video C / Component Video Y | O | Either the chrominance signal of S Video or the luma signal of component video. Only on Apple devices that support the appropriate video signal. |
| 23 | Composite Video / Component Video Pb | O | Either the Composite Video signal or the (Pb) signal of Component Video. Only on Apple devices that support the appropriate video signal. |
| 24 | Remote Sense | I | See "Minimizing Crosstalk and Noise" (page 42). Remote Sense is supported only on Apple devices that support video output; however, it is okay to connect this pin on other Apple devices. |
| 25 | LINE-IN L | I | Left channel audio line level input to the Apple devices listed in Table 2-2 (page 26); **deprecated.** |
| 26 | LINE-IN R | I | Right channel audio line level input to the Apple devices listed in Table 2-2 (page 26); **deprecated.** |
| 27 | LINE-OUT L | O | Line level output from the Apple device for the left channel. |
| 28 | LINE-OUT R | O | Line level output from the Apple device for the right channel. |
| 29 | A/V Return | — | Audio/Video return. This is a signal and must not be grounded inside the accessory. |
| 30 | Apple Device Detect | O | Ground signal for Apple device detection. The accessory must ground this pin if it is not monitoring it to detect Apple device attachment. |

| Pin | Signal name | I/O | Function |
| --- | --- | --- | --- |
| | Chassis | | Mounting tabs: chassis ground for connector shell. |

# Headphone/Microphone Jack

Each Apple device contains a standard stereo headphone jack, but the headphone jack on some models may also be used as a microphone input, with or without a control button. Figure 1-1 (page 21) shows an example of circuitry in a headphone/microphone accessory for the iPhone.

**Figure 1-1** Typical headphone/microphone circuitry in an iPhone accessory



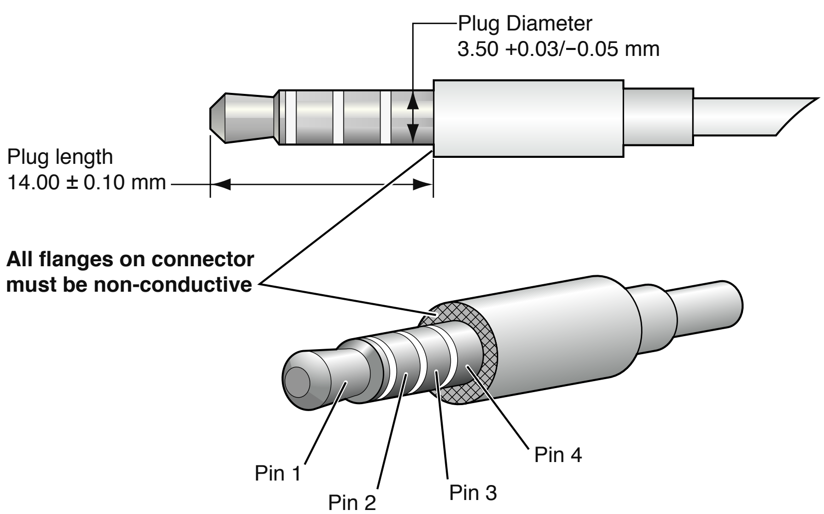

| Component | Value |
| --- | --- |
| Capacitor C1 | 1 µF |
| Resistor R1 | 2.21 kΩ ±1% |
| Resistor R2 | Codec input impedance ≥2 kΩ |
| L01, L02 | 600 Ω @ 100 MHz |
| SW | iPhone action button |
| VAR1 | 12 V varistor |

**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 iPhone 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 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 in Figure 1-1 (page 21), measured under test conditions of $V_s$ = 2.0 V, $R_L$ = 2.2 kΩ, $T_a$ = 20° C, and relative humidity = 65%.

Figure 1-2 (page 22) shows the configuration and critical dimensions of the headphone plug that fits the Apple device's headphone/microphone jack.

**Figure 1-2** Headphone plug dimensions




Other dimensions of the Apple device's headphone/microphone jack conform to the JEITA standard RC 5325A, "4-Pole Miniature Concentric Plugs and Jacks." Table 1-2 (page 22) lists its electrical connections.

**IMPORTANT:** Every accessory cable and every connector that can be inserted into an Apple device's headphone/microphone jack must meet the foregoing requirements.

**Table 1-2** Headphone plug connections

| Pin | Description |
|---|---|
| 1 | Audio left output |
| 2 | Audio right output |
| 3 | Signal return |
| 4 | Microphone input |

Any accessory that passes signals between the jack on an Apple device and another jack on the accessory capable of accepting an Apple device headphone/microphone plug must support the button press technology described in "Headphone Remote and Mic System" (page 71).

# Wireless Standards

iOS devices use these industry standard wireless protocols:

## iPhone Protocols

- GSM/EDGE (850, 900, 1800, 1900 MHz)
- Wi-Fi (802.11 b/g)
- Bluetooth 2.1 + EDR; see Web site iPhone, iPad, iPod touch: Supported Bluetooth profiles for a list of supported profiles.
- (iPhone 3G and 3GS only) UMTS/HSPDA (850, 1900, 2100 MHz)
- (iPhone 3G, 3GS, and 4 (all models)) Assisted GPS
- (iPhone, all models) Wi-Fi (802.11 b/g/n)
- (iPhone 4, all models) UMTS/HSDPA (850, 900, 1900, 2100 MHz)

## iPod touch Protocols

- Wi-Fi (802.11 b/g)
- (2G, 3G, and 4G touch only) Bluetooth 2.1 + EDR

## iPad Protocols

- Wi-Fi (802.11 a/b/g/n)
- Bluetooth 2.1 + EDR
- (iPad 3rd generation only) Bluetooth 4.0
- (iPad 3G only) UMTS/HSPDA (850, 1900, 2100 MHz) Data Only
- (iPad 3G only) GSM (850, 900, 1800, 1900 MHz) Data Only
- (iPad 3G + 4G only) Assisted GPS
- (iPad 4G only) UMTS/HSPA/HSPA+/DC-HSDPA (850, 900, 1900, 2100 MHz) Data Only
- (iPad 4G only) LTE (700, 2100 MHz) Data Only

# Functional Hardware Description

This chapter describes the functional characteristics of Apple devices and their interfaces to accessories, assuming that current firmware is installed.

## Apple Device Features and Capabilities

The following tables mark the high-level capabilities of each of the Apple devices that support the iPod Accessory Protocol (iAP):

- Table 2-1 (page 25) shows the hardware transport protocols by which different models of Apple devices communicate with accessories.
- Table 2-2 (page 26) shows the hardware features available through each model's 30-pin connector.
- Table 2-3 (page 27) shows each model's compatibility with external systems.
- Table 2-4 (page 28) shows the USB device classes that each Apple device model supports when it is in USB Host Mode.

**Table 2-1** iAP transports

| Apple device | UART | USB Device Mode | USB Host Mode | Bluetooth |
|---|---|---|---|---|
| iPod mini | × | | | |
| 4G iPod | × | | | |
| iPod photo, 4G iPod (color display) | × | | | |
| iPod nano | × | × | | |
| 5G iPod | × | × | | |
| 2G nano | × | × | | |
| iPod classic | × | × | | |
| 3G nano | × | × | | |
| iPhone | × | × | | |
| iPod touch | × | × | | |
| 4G nano | × | × | | |
| 120 GB classic | × | × | | |

| Apple device | UART | USB Device Mode | USB Host Mode | Bluetooth |
|---|---|---|---|---|
| 2G touch | × | × | | × |
| iPhone 3G | × | × | | × |
| iPhone 3GS | × | × | × | × |
| 3G touch | × | × | × | × |
| 5G nano | × | × | | |
| 160 GB classic | × | × | | |
| iPad (all models) | × | × | × | × |
| iPhone 4 (all models) | × | × | × | × |
| 6G nano | × | × | × | |
| 4G touch | × | × | × | × |
| iPad 2 (all models) | × | × | × | × |
| iPad (3rd generation) | × | × | × | × |

**Table 2-2** 30-pin connector hardware features

| Apple device | Analog audio out | Analog audio in | Analog video out (S-video/ composite) | Analog video out (component) | Para | USB Device Mode | USB Host Mode |
|---|---|---|---|---|---|---|---|
| iPod mini | × | × | | | Para | × | |
| 4G iPod | × | × | | | Para | × | |
| iPod photo, 4G iPod (color display) | × | × | × | | Para | × | |
| iPod nano | × | × | | | Para | × | |
| 5G iPod | × | × | × | | Para | × | |
| 2G nano | × | × | | | Para | × | |
| iPod classic | × | × | × | × | Para | × | |
| 3G nano | × | × | × | × | Para | × | |
| iPhone | × | × | × | × | Para | × | |
| iPod touch | × | × | × | × | Para | × | |
| 4G nano | × | × | × | × | Para | × | |

| Apple device | Analog audio out | Analog audio in | Analog video out (S-video/ composite) | Analog video out (component) | Para | USB Device Mode | USB Host Mode |
|---|---|---|---|---|---|---|---|
| 120 GB classic | × | × | × | × | Para | × | |
| 2G touch | × | × | × | × | Para | × | |
| iPhone 3G | × | × | × | × | Para | × | |
| iPhone 3GS | × | × | × | × | Para | × | × |
| 3G touch | × | × | × | × | Para | × | × |
| 5G nano | × | × | × | × | Para | × | |
| 160 GB classic | × | × | × | × | Para | × | |
| iPad (all models) | × | | × | × | × | × | × |
| iPhone 4 (all models) | × | | × | × | × | × | × |
| 6G nano | × | | × | | Para | × | ×[1] |
| 4G touch | × | | × | × | × | × | × |
| iPad 2 (all models) | × | | × | × | × | × | × |
| iPad (3rd generation) | × | | × | × | × | × | × |

[1] The 6G nano can enter USB Host mode only if the accessory provides USB power to it.

**Table 2-3** Additional hardware features

| Apple device | Nike+iPod Cardio Equipment support | Headphone Remote and Mic System | USB Device Mode audio |
|---|---|---|---|
| iPod mini | | | |
| 4G iPod | | | |
| iPod photo, 4G iPod (color display) | | | |
| iPod nano | | | × |
| 5G iPod | | | × |
| 2G nano | | | × |
| iPod classic | | | × |
| 3G nano | × | | × |
| iPhone | | | × |

| Apple device | Nike+iPod Cardio Equipment support | Headphone Remote and Mic System | USB Device Mode audio |
|---|---|---|---|
| iPod touch | | | × |
| 4G nano | × | × | × |
| 120 GB classic | | × | × |
| 2G touch | × | × | × |
| iPhone 3G | | | × |
| iPhone 3GS | × | × | × |
| 3G touch | × | × | × |
| 5G nano | × | × | × |
| 160 GB classic | | × | × |
| iPad (all models) | | × | × |
| iPhone 4 (all models) | × | × | × |
| 6G nano | × | × | × |
| 4G touch | × | × | × |
| iPad 2 (all models) | | × | × |
| iPad (3rd generation) | × | × | × |

**Table 2-4** USB Host Mode supported device classes

| Apple device | Audio | HID | MIDI |
|---|---|---|---|
| iPod mini | | | |
| 4G iPod | | | |
| iPod photo, 4G iPod (color display) | | | |
| iPod nano | | | |
| 5G iPod | | | |
| 2G nano | | | |
| iPod classic | | | |
| 3G nano | | | |
| iPhone | | | |

| Apple device | Audio | HID | MIDI |
|---|---|---|---|
| iPod touch | | | |
| 4G nano | | | |
| 120 GB classic | | | |
| 2G touch | | | |
| iPhone 3G | | | |
| iPhone 3GS | | × | × |
| 3G touch | | × | × |
| 5G nano | | | |
| 160 GB classic | | | |
| iPad (all models) | × | × | × |
| iPhone 4 (all models) | × | × | × |
| 6G nano | × [1] | | |
| 4G touch | × | × | × |
| iPad 2 (all models) | × | × | × |
| iPad (3rd generation) | × | × | × |

[1] USB Audio 1.0 output only.

# 30-Pin Connector Functions

This section describes certain functional characteristics of power and signals through the Apple device's 30-pin connector.

## USB 2.0

The 30-pin connector includes a USB interface designed to the USB 2.0 High Speed Specification. For full specifications of the Universal Serial Bus, see http://www.usb.org/developers/docs.

Apple devices are USB-compliant devices that can be connected to a USB-compliant host or to a USB power supply. When connected to an accessory, the Apple device follows the rules in the *USB 2.0 Specification,* Section 7.1.7.3, to determine whether the accessory is signaling on the USB D+/D– lines or providing charging current. A USB accessory should wait at least 2 seconds after it is disconnected before reconnecting, to give the Apple device time to clear its USB stack and apply the USB rules at connection time.

**Note:** Accessories must comply with all published USB specifications. In particular, the connection between the accessory and the Apple device must not show signs of consistent USB data corruption or packet loss.

Most Apple devices provide a USB Interface (iUI) for accessories. The iUI allows the Apple device to be controlled through iAP, using a USB Human Interface Device (HID) interface as a transport mechanism. See "USB" (page 52) for more information.

**Note:** Accessories must not use Apple devices as mass storage devices.

Accessories may contain internal or external power supplies, but their design must follow the guidelines in "iOS Device Power Supply Requirements" (page 85) to avoid interfering with the sensor operation of touch screens. In addition, accessory power supplies must meet the electrical certification requirements described in "Electrical Testing and Certification" in *MFi Accessory Testing Specification*.

The requirements for accessory power supplies depend on whether they provide only power over USB or both power and data over USB. The differences are specified in the following sections.

## USB Power

**Note:** Accessories that have a 30-pin connector and do not implement iAP must provide power to the Apple device.

To be compatible with iPods and iPhones, an accessory must provide a minimum of 4.70 V and 1.0 A; to be compatible with iPads as well, it must provide a minimum of 4.55 V and 2.1 A. These voltages and currents must be measured at the USB Vbus input of the Apple device (pin 8 on its 30-pin connector). If possible, accessories should provide 5.0 V to decrease the Apple device's charging time.

An accessory that powers an Apple device and does not communicate with the Apple device using iAP, nor permit a USB host to connect to the Apple device through the accessory, must connect the USB D+ and D– pins as shown in Figure 2-1 (page 30). 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. Some Apple devices will not charge if an attached power source lacks the necessary resistors.

**Figure 2-1** D+ and D- connections for USB power sources

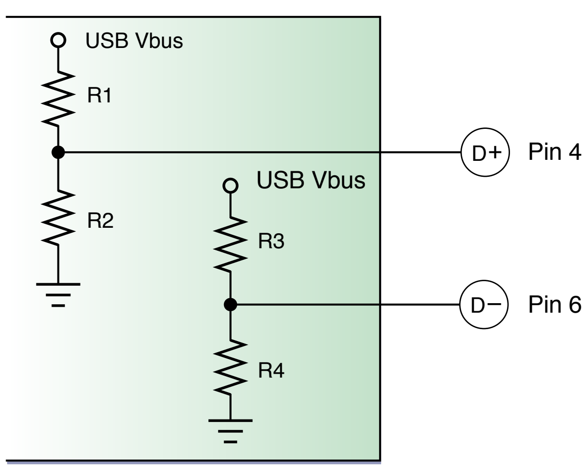


**Table 2-5** USB power supply component specifications

| Specification | 2.1 A at 5.0 V (10.5 W nominal) | 1.0 A at 5.0 V (5.0 W nominal) | 500 mA at 5.0 V |
|---|---|---|---|
| Product applicability | Required for the iPad and iPad-compatible accessories; OK for the iPod and iPhone | Required for the iPod and iPhone | Deprecated (do not use) |
| Resistor R1 (see Figure 2-1 (page 30)) | 43.2 kΩ | 75.0 kΩ | 75.0 kΩ |
| Resistor R2 | 49.9 kΩ | 49.9 kΩ | 49.9 kΩ |
| Resistor R3 | 75.0 kΩ | 43.2 kΩ | 75.0 kΩ |
| Resistor R4 | 49.9 kΩ | 49.9 kΩ | 49.9 kΩ |

**Notes:** The 4G iPod, iPod Classic, and 5G iPod are not compatible with the 10.5 W charger specification. 500 mA power supplies are allowed only for certain battery pack accessories and require explicit approval from Apple.

For safety reasons, an accessory providing power to an Apple device must detect any non-transient current drain of more than 2.5 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.

Every Apple device-compatible connector on an accessory that uses D+/D- resistors, as shown in Figure 2-1 (page 30), must have its own set of resistors. The accessory must be capable of supplying the total current required when all ports are in use, regardless of whether the ports are Apple device-compatible 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 (a) the standard USB type-A connectors supplying 500 mA must be labeled using the USB icon, and (b) the Apple device-compatible USB type-A connectors must be labeled, either singly or in groups, with the text “iPod/iPhone” or “iPad” depending on their device compatibility. Additionally, if the accessory has multiple 30-pin 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.

To be compatible with iOS devices, accessories that supply power must be designed to minimize electrical interference with the touch screen; see “iOS Device Power Supply Requirements” (page 85).

**Note:** Accessories that charge Apple devices must not wait for Accessory Power (pin 13 of the 30-pin connector) to go high before supplying power on USB Vbus (pin 8).

## Power and Data

An accessory that powers an Apple device and communicates with it using iAP, or permits a USB host to connect to it through the accessory, must comply with the requirements in this section in addition to the requirements specified in “USB Power” (page 30).

### Connectors Other Than 30-Pin Dock Connectors

All accessory connectors designed for use with a separate cable that terminates in an Apple 30-pin dock connector must meet the following requirements, regardless of whether the cable is included with the accessory or provided by the end user:

- The connectors must provide at least 4.90 V and 1.0 A power to be compatible with iPods and iPhones.
- They must provide at least 4.97 V and 2.1 A power to be also compatible with iPads.
- They 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).

### Cables Terminating in 30-Pin Dock Connectors

All accessory cables that terminate in an Apple 30-pin dock connector must meet the following requirements:

- The cables must meet or exceed all applicable USB-IF specifications.
- They must test for DC resistances below the limits shown in Table 2-6 (page 32).
- They must exhibit no more inductance than a simple ferrite bead on every circuit.
- To minimize power losses, every cable shield should be connected to the ground conductor at both ends.

**Table 2-6** USB cable maximum DC resistances

| Specification | Maximum DC resistance in milliohms |
|---|---|
| Round-trip $V_{BUS}$ with shield shorted to GND at each end | 200 mΩ |
| Round-trip $V_{BUS}$ without shield shorted to GND | 300 mΩ |
| $V_{BUS}$ conductor alone | 160 mΩ |
| Ground conductor alone | 140 mΩ |
| Braided shield alone | 85 mΩ |

### External Hosts

An accessory connected to an Apple device that either passes Vbus power from a USB host (such as a personal computer), or passes the host's D+ and D– signals, must pass both Vbus and D+/D–. The accessory must ensure that the external USB data signals and Vbus power are passed through to the Apple device without interference, and without violating the USB Specification in such areas as voltage tolerance on the Vbus line and rise time, eye diagrams, and monotonicity requirements on the D+/D– data lines.

The resistors that must be present on the USB D+ and D– lines when passing Vbus power to an Apple device (as specified in "USB Power" (page 30)) may interfere with USB communications on the those lines. An accessory with an external port that can be connected to a personal computer (for Apple device synchronization, for example) must detect the presence of the computer and disable or remove the resistors from the D+/D– lines electronically.

When an accessory disables or removes resistors from the USB D+ and D– lines, it must use a switch with an open capacitance of less than 10 pF; the design goal is 6 pF. This is critical to avoid degrading the signal characteristics of high-speed USB data traffic. In addition, the USB D+/D– lines must maintain a differential impedance of less than 90 Ω with minimal DC resistance.

When an accessory attached to an Apple device is also connected to a personal computer, it must pass USB Vbus power to the Apple device via pin 8 of the 30-pin connector. Some personal computers can supply more than 500 mA through their USB ports; this ensures that if the Apple device needs more power than is available from the accessory, it will be able to get extra power from the computer. Hence the Apple device must be able to draw power from the computer's USB port whenever the accessory is plugged into it. Failure to pass Vbus power through may result in the Apple device not communicating with the computer.

**Note:** The traces and circuits used to pass USB Vbus power must be capable of handling at least 2.5 A. This is the minimum current necessary to guarantee synchronization of an Apple device with iTunes on any computer.

## Informing Apple Devices of Available Power

An accessory that provides power to an Apple device must tell the Apple device how much power is available at all times. The way that the accessory must do this depends on its iAP transport, as shown in Table 2-7 (page 33).

**Table 2-7** Ways to communicate power availability

| iAP transport | Way to communicate available power |
|---|---|
| None | Use D+/D- resistors, as shown in Figure 2-1 (page 30). |
| UART | Send the iAP command `SetAvailableCurrent` or use D+/D- resistors as shown in Figure 2-1 (page 30). `SetAvailableCurrent` is recommended. |
| USB Host mode | Send the iAP command `SetAvailableCurrent`. |
| USB Device mode | If the accessory grounds Accessory Detect, send the iAP command `SetAvailableCurrent`. If the accessory does not ground Accessory Detect (such as one that uses a USB-to-30-pin dock connector cable), send USB Vendor Requests as specified in "USB Device Mode Vendor Requests" (page 33). |
| Bluetooth | Use D+/D- resistors, as shown in Figure 2-1 (page 30). |

### USB Device Mode Vendor Requests

The USB device request shown in Table 2-8 (page 34) tells an Apple device in USB Device mode how much power is available. The device request may be sent after the first 8 bytes of the Apple device's USB device descriptor are received; this lets the maximum packet size of EP0 be determined. For details of this and other parts of the USB interface, see http://www.usb.org/developers/docs.

**Note:** When an Apple device is connected or powered on, it can momentarily draw more than 1 A from the accessory. Accessories must comply with the USB specification for voltage and inrush current.

**Table 2-8** USB Device Vendor Request to set available current from accessory (USB Device Mode only)

| Field | Value | Description |
|---|---|---|
| `bmRequestType` | 0x40 | Host-to-device request, vendor-defined type, device is recipient |
| `bRequest` | 0x40 | Vendor-defined power request |
| `wValue` | Typical values: 0, 100, or 500 | Extra current (in mA) the Apple device may draw in suspend state |
| `wIndex` | Recommended value for Apple devices: +500<br>Recommended value for iPads: +1600 | Extra current (in mA) the Apple device may draw when not in suspend state. Recommended value of +500 allows 1 A configured; +1600 allows 2.1 A configured; –500 allows nothing configured. |
| `wLength` | 0 | No data transfer |

**Note:** The combined value of the extra current and the specification current is rounded down to the next lower useful value (0, 0.5, 2.5, 100, 500, 1000).

The time at which the USB device request specified in Table 2-8 (page 34) takes effect varies by the USB Product ID of the Apple device, as shown in Table 2-9 (page 34). USB Product IDs are discussed in "Choosing an Apple Device USB Configuration" (page 53).

**Table 2-9** USB device request timing

| USB product ID | Product description | Request timing |
|---|---|---|
| 0x1200–0x1260 | Older Apple devices | All requests take effect at configuration time and are cancelled by sleep. |
| 0x1261–0x128F | iPod 3G nano, Classic, and newer Apple devices | Requests for extra current take effect as soon as they are sent; requests for extra suspend current take effect at suspend time. Neither request is cancelled by the iPod sleeping or waking up. |
| 0x1290–up | iPhone, iPod touch, iPhone 3G, 2G touch, iPhone 3GS | Currently, all requests take effect at configuration time and are cancelled by sleep. This behavior may change to that of newer Apple devices in the future. |

# Accessory Power

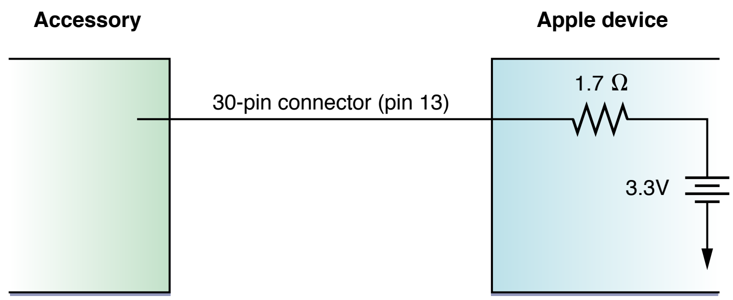
By default, the Accessory Power pin (pin 13 of an Apple device's 30-pin connector) supplies 5 mA (low power mode) at 3.0 V to 3.3 V ±5% (2.85 V to 3.465 V). Accessories requesting more power must meet all the requirements specified in "Accessory Power Policy" (page 47). When accessories are inactive, they must consume 5 mA or less (low power mode).

**Note:** Accessories must not use the Accessory Power output to detect the presence of an Apple device.

Current that the Apple device supplies on the Accessory Power line of the 30-pin connector is switched off for at least 2 seconds during the Apple device's bootstrap process. This is done to reset Apple device-powered accessories and make them identify themselves to the Apple device 80 ms after power resumes. Self-powered accessories must stop sending iAP packets when their power turns off and identify themselves to the Apple device 80 ms after their power turns back on.

Accessory power is returned through pins 1, 2, 15, and 16 of the 30-pin connector. The internal resistance of the Accessory Power supply in a typical Apple device is shown in Figure 2-2 (page 35).

**Figure 2-2** Accessory power internal resistance




## Avoiding RX Back-Powering

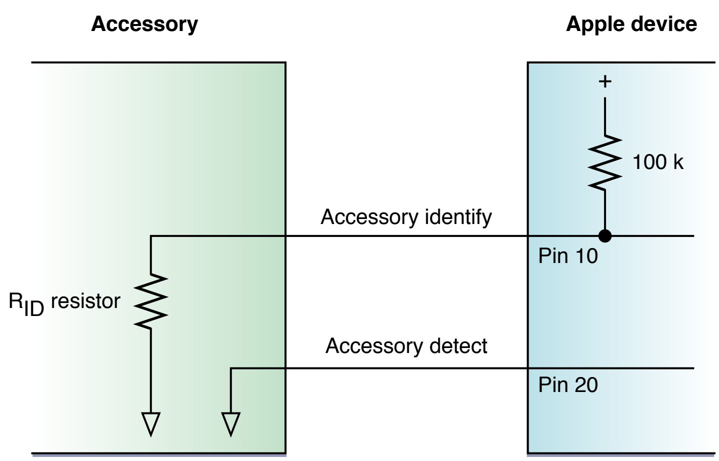
When the current that the Apple device supplies on the Accessory Power line of the 30-pin connector is off, the Serial Receive block in the Apple device's UART may also be off. To avoid back-powering the Apple device, the accessory must put line RX (pin 18) of the Apple device's 30-pin connector into a Hi-Z state, using tri-state logic, when the Accessory Power line goes low. Any attempt by the accessory to drive the UART serial line high when the Apple device's Serial Receive block is off turns on protection diodes in the Apple device. This condition wastes power in the accessory and can cause adverse behavior in the Apple device. For a similar warning about the accessory, see "Avoiding TX Back-Powering" (page 96).

## Accessory Detect and Identify

**Note:** Some of the material in this section does not apply to accessories that need to support the 3G iPod. See "Interfacing With the 3G iPod" (page 105).

Apple devices are notified when an accessory is physically attached and the Accessory Detect pin is grounded. Many accessories must also present a valid Accessory Identify resistor ($R_{ID}$) between pin 10 and ground, to invoke specific Apple device behavior and/or use iAP commands over certain iAP transport protocols. These pins and their connections inside the accessory are shown in Figure 2-3 (page 36). Table 2-10 (page 37) specifies which resistor values are required for specific combinations of iAP transports and Apple device behavior. Accessories must not try to invoke other combinations.

**Figure 2-3** Accessory Identify and Accessory Detect Signals




Self-powered accessories must simulate a physical disconnect by floating both Accessory Detect and Accessory Identify pins if the accessory is turned off but not physically detached from the Apple device. However, accessories must not simulate a disconnect in response to the current that the Apple device supplies on the Accessory Power line of the 30-pin connector going low, or other changes in the state of the Apple device.

If an accessory is self-powered and can enter an Off state while still remaining physically attached to the Apple device, it must comply with these three requirements:

- Upon entering the Off state, the accessory must immediately float both Accessory Detect and Accessory Identify pins.
- Upon entering the On state, the accessory must immediately ground Accessory Detect and present a valid Accessory Identify resistor.
- The accessory must not enter the Off state in response to Accessory Power going low or any other changes in the state of the Apple device.

**Notes:** Unless otherwise specified in Table 2-10 (page 37), accessories must perform the Identify Device Preferences and Settings (IDPS) process specified in the latest release of *MFi Accessory Firmware Specification*. This process must begin within 2 seconds after the accessory grounds Accessory Detect and presents a valid $R_{ID}$ resistor.

The tolerance of $R_{ID}$ resistors must be 1% or less. Accessories must not use resistor values other than those shown in the table. The resistor value must not change when the accessory is connected to an Apple device.

All accessories must use iAP commands to receive analog audio from an Apple device.

**Table 2-10** Resistor-based accessory functions

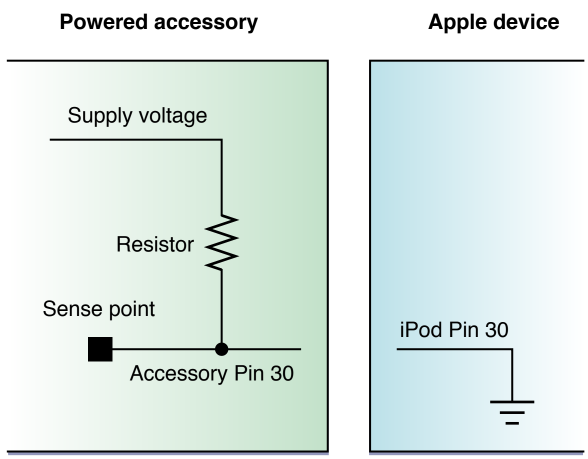
| iAP transport | Resistor value | Functional behavior and requirements |
|---|---|---|
| No iAP transport | None | All charger accessories. Do not ground Accessory Detect (pin 20). No IDPS process is required. |
| No iAP transport; power only | 255 kΩ | The Apple device draws power from the accessory but does not charge its internal battery. Its user interface displays the battery level as full, instead of charging. The iPhone requires firmware version 1.1.3 or later. The accessory must ground Accessory Detect (pin 20). No IDPS process is required. |
| iAP over UART | 549 kΩ | The accessory must ground Accessory Detect (pin 20). |
| iAP over Bluetooth | None | Do not ground Accessory Detect (pin 20). |
| iAP over USB; USB Host mode | 28 kΩ | The accessory must ground Accessory Detect (pin 20). It may use UART transport only long enough to detect support for USB Host mode and enter that mode as specified in *MFi Accessory Firmware Specification*. |
| iAP over USB; USB Device mode | None | The accessory must use a standard USB `Set_Configuration` request to select the iAP HID Interface. It should also ground Accessory Detect (pin 20); see "Informing Apple Devices of Available Power" (page 33) for the consequences of not grounding Accessory Detect. |

## Detecting an Apple Device

If a powered accessory needs to detect that an Apple device has been plugged in, even when the Apple device has no battery power or is hibernating, it must tie its pin 30 to its supply voltage through a resistor and examine the state of the signal on the line. The signal state will be high when no Apple device is attached. The signal state will be pulled low when an Apple device is attached because pin 30 on the Apple device is tied to ground.

> **Note:** The accessory must obtain digital ground by connecting pins 1, 2, 15 and 16 together, or as many of these pins as are available to it.

**Figure 2-4** Apple device detection




> **Note:** The elapsed time it takes an Apple device to bootstrap and be fully operational varies by model type, media types present, storage capacity, and the starting boot state (for example, Hibernate or Reset). An Apple device does not receive or respond to any iAP commands while the current that the Apple device supplies on the Accessory Power line of the 30-pin connector is switched off. To use iAP, accessories that communicate via UART must always monitor the Accessory Power line and wait until 80 ms after it has gone high before trying to communicate with the attached Apple device. See "Accessory Signaling and Initialization" in *MFi Accessory Firmware Specification*. Accessories that do not communicate via UART, such as USB hosts or USB or Bluetooth devices, need not monitor the Accessory Power line.

## UART iPod Accessory Protocol Communication

Accessories using the iPod Accessory Protocol (iAP) over UART transport use two pins, Apple Device RX and Apple Device TX, to communicate to and from the Apple device. See "UART" (page 51) for electrical details.

Attaching a UART serial accessory to the 30-pin connector of a 3G iPod makes any accessories attached to the 9-pin Audio/Remote connector inactive, because the 3G iPod shares the UART serial port between these two connectors. The iPod mini, 4G iPod, iPod photo, and 4G iPod (color display) models have two UART serial ports; in these models, plugging a UART serial accessory to the 30-pin connector does not deactivate the UART serial port on the 9-pin Audio/Remote connector.

## Apple Device Video Signal Levels

Video signal levels from an Apple device's 30-pin connector are shown in Table 2-11 (page 39). These levels assume that the video outputs are properly load-terminated into 75 Ω ±1%.

By default, an Apple device sends only a Composite Video signal. Some Apple devices also send S-Video signals by default. Use the General lingo command `SetiPodPreferences` to select the appropriate signal for your accessory.

To receive video signals from most Apple devices, an accessory must be authenticated. The iPod photo, 4G iPod (color display) and 5G iPod are exempt from this requirement.

**Table 2-11** Video signal levels in volts peak-to-peak

| Description | Minimum | Typical | Maximum |
| --- | --- | --- | --- |
| Composite Video Amplitude | 0.30 | | 1.0 |
| Composite Video Burst Amplitude (NTSC) | 0.21 | 0.28 | 0.33 |
| Composite Video Sync Amplitude (NTSC) | 0.28 | 0.28 | 0.33 |
| S-Video Luminance Amplitude | 0.30 | | 1.0 |
| S-Video Chrominance Amplitude | 0.0 | | 0.70 |
| S-Video Chrominance Burst Amplitude (NTSC) | 0.21 | 0.28 | 0.33 |
| Composite Video Burst Amplitude (PAL) | 0.27 | 0.30 | 0.33 |
| Composite Video Sync Amplitude (PAL) | 0.27 | 0.30 | 0.33 |
| S-video Burst Amplitude (PAL) | 0.27 | 0.30 | 0.33 |
| Component Video Luminance (“Y”) | 0.3 | | 1.0 |
| Component Video Chrominance (“Pr”) | 0.0 | | 0.7 |
| Component Video Chrominance (“Pb”) | 0.0 | | 0.7 |

**Note:** All accessories that output video from the 30-pin connector must play the associated audio stream in sync with the video.

**Note:** iPhones with firmware version 1.1.1 or greater support composite or component video output on pins 21, 22, and 23. iPhones with firmware versions 1.0 through 1.0.2 do not support video outputs.

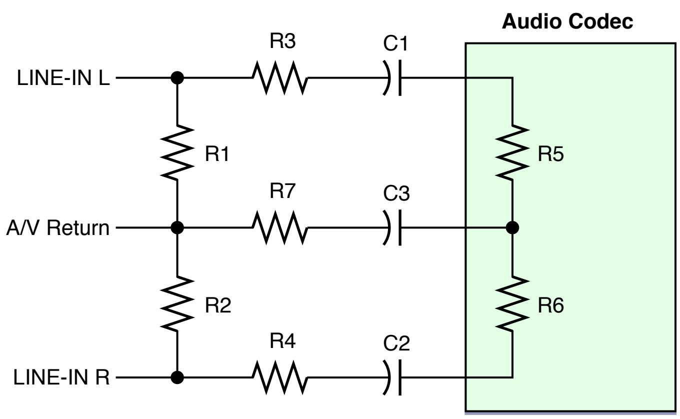
## Line Level Input

The 30-pin connector of the iPod models listed in Table 2-2 (page 26) supports both stereo and mono analog audio inputs, with an input line level of 0.600 Vrms.

Mono audio input is through the left channel. Figure 2-5 (page 40) shows a reference schematic for Apple device line-in impedance on the iPod classic, 4G nano, and 2G touch. The corresponding reference schematic for other Apple devices is shown in Figure 2-6 (page 40).

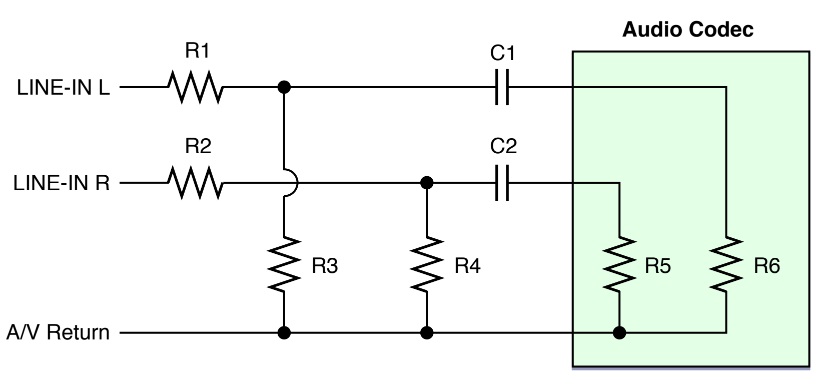
> **Note:** The A/V Return pin is not a ground and must not be tied directly to the digital ground (DGND); treat it as a signal. Left and right audio are referenced to A/V Return.

**Figure 2-5** Equivalent input circuits for the iPod classic, 4G nano, 2G touch, and 5G nano




| C1 | C2 | C3 | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|---|---|---|---|---|---|---|---|---|---|
| 4.7 µF | 4.7 µF | 4.7 µF | 100 kΩ | 100 kΩ | 100 Ω | 100 Ω | 40 kΩ | 40 kΩ | 100 Ω |

**Figure 2-6** Equivalent input circuits for other Apple devices




| Apple device models | C1 | C2 | R1 | R2 | R3 | R4 | R5 | R6 |
|---|---|---|---|---|---|---|---|---|
| 1G nano | 1 µF | 1 µF | 100 Ω | 100 Ω | 100 kΩ | 100 kΩ | 40 kΩ | 40 kΩ |
| 3G iPod | 10 µF | 10 µF | 100 Ω | 100 Ω | 100 kΩ | 100 kΩ | 30 kΩ | 30 kΩ |
| Other Apple device models | 10 µF | 10 µF | 100 Ω | 100 Ω | 100 kΩ | 100 kΩ | 22 kΩ | 22 kΩ |

## Line Level Output

Stereo audio output from an Apple device is at a fixed level; it is not adjustable. The output levels for various Apple devices are shown in Table 2-12 (page 41).

**Table 2-12** Audio output line levels

| Apple device models | Output level |
|---|---|
| 3G iPod, 4G iPod, iPod mini, iPod photo, 4G iPod (color display), 5G iPod, 1G iPod touch, iPhone, iPhone 3G, iPhone 3GS, iPad (all models), iPhone 4 (all models), iPad 2 (all models), iPad (3rd generation) | 0.900 Vrms |
| iPod classic, 4G nano, 2G touch, 5G nano | 0.800 Vrms |
| iPod nano, 2G nano, 3G nano | 0.700 Vrms |

Figure 2-7 (page 41) shows a reference schematic for line-out and headphone-out impedance on the iPod classic, 2G touch, 4G and 5G nanos, and all future Apple device models. The corresponding reference schematic for other Apple device models is shown in Figure 2-8 (page 42). These values are approximate and may change in the future without notice.

**Note:** Accessories that need line-out enabled from an Apple device must request it using iAP commands. For details of setting line-out and other preferences using the Identify Device Preferences and Settings process, see "Accessory Identification and Authentication" in *MFi Accessory Firmware Specification.*

**Figure 2-7** Equivalent output circuits for the iPod classic, 2G touch, 4G nano, 5G nano, and future models

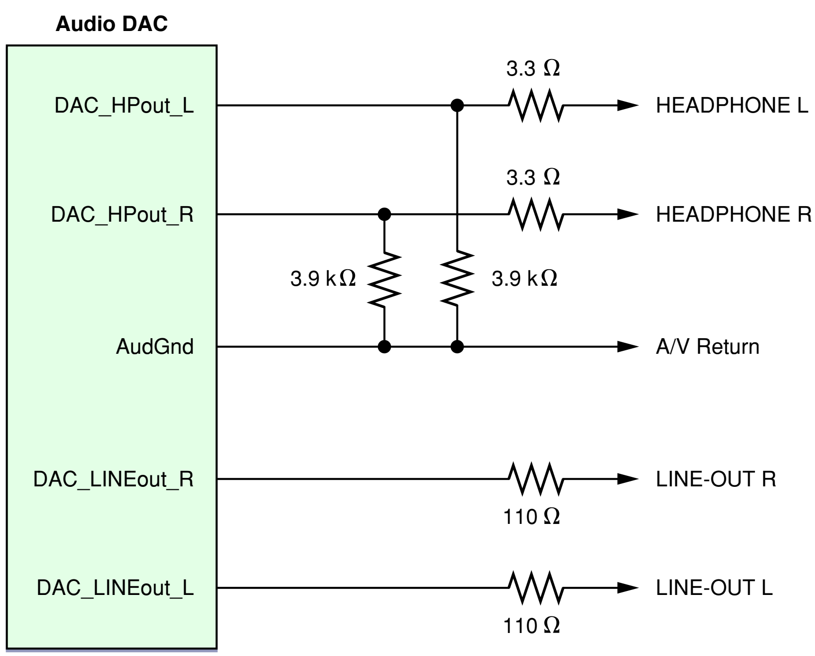
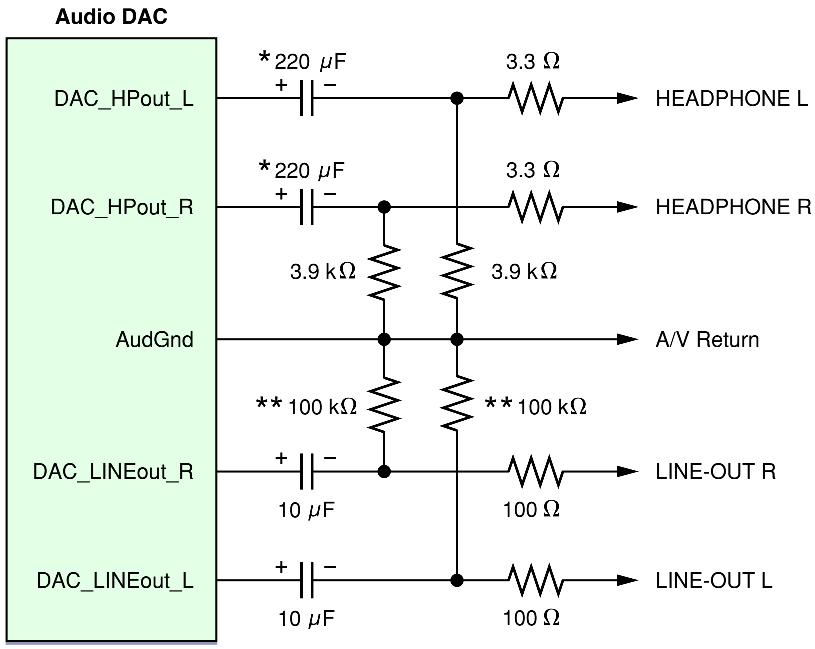


**Figure 2-8** Equivalent output circuits for other Apple devices




* 100 μF on 3G, mini (pre-June 2005), 4G, and photo iPods
** 10 kΩ on some models

**Note:** Under some circumstances, the 2G iPod nano, the 2G touch, and the iPod classic disable their line-out circuits to save power. Line-out is disabled only if a headphone is detected in the headphone jack, no accessory identify resistor is detected, **and** power is not attached to the Apple device.

## Overall Grounding Requirements

Chassis ground is tied to the specified pins; see "Hardware Interfaces" (page 19).

**IMPORTANT:** Digital ground must not be tied to A/V Return.

## Minimizing Crosstalk and Noise

Accessories that receive audio or video signals from an Apple device must be designed to minimize crosstalk and extraneous noise. Both the accessory's circuitry and physical trace layout must take these issues into account.

Apple devices with video output capability use single-ended analog audio and video signals. To conserve pins on connectors and conductors in cables, the audio and video signals share a common return path. To mitigate video-to-audio crosstalk, active differential amplifiers are used in the audio path. The differential amplifier stage is configured to sense a low side return voltage representing the video-to-audio crosstalk. This signal is then applied to the load in such a way that minimum noise current flows through the load. For sample schematics and advice on circuit layout, see "Sample Accessory Circuits" (page 87).

In Apple devices without video capability, pins 21, 22, and 23 are not connected internally. See Table 1-1 (page 19) for descriptions of these pins.

# Apple Device Power States

To optimize music playback time and best utilize the internal battery and external power sources (such as a powered dock, power brick, computer, or other powered accessory), Apple devices support several different power states. These power states affect accessories, particularly those that are powered by the Apple device. The Apple device can transition between power states as a result of the inactivity of its UI, its internal battery-conservation actions, or accessory actions.

The iPod power states are Power On, Sleep, and Hibernate. The power states of iOS devices are Power On, Hibernate, and Power Off. iOS devices enter the Hibernate state directly from the Power On state; however, they generate a Sleep state notification immediately prior to the Hibernate notification. This Sleep state notification ensures compatibility with accessories that always expect to receive one before a Hibernate notification.

The Power On state consumes the most power. Sleep consumes less power, although parts of the iPod are still powered in order to respond to iAP commands. Hibernate is the lowest powered state and is used to preserve internal battery power for extended periods of inactivity, such as days or weeks. Off is a completely unpowered state. Table 2-13 (page 43) describes the power states for all Apple devices.

**Table 2-13** Power states and transitions

| Power state | Power On | Sleep | Hibernate | Off |
|---|---|---|---|---|
| **Apple devices** | All devices | iPods | All devices | iOS devices |
| **Display state** | ON | OFF | OFF | OFF |
| **Attributes** | The Apple device UI may be active, allowing users to interface with the menus, listen to music, view images, and so forth. If the UI is inactive—that is, if no track is playing and there is no user input for 2 minutes—the Apple device displays a large battery icon (either "Charged" or "Charging"). | The iPod UI is inactive—no track is playing—and the UI state is preserved. The iPod can respond immediately to front panel buttons or to iAP simple remote buttons. Attaching some accessories may also wake the device. | The Apple device is inactive (no track is playing). It does not respond to iAP packets over any iAP transports. | The iOS device is completely powered down. It does not respond to iAP packets over any iAP transports. |

| Power state | Power On | Sleep | Hibernate | Off |
|---|---|---|---|---|
| **Accessory power** | ON<br>Typical current that the accessory draws from the Apple device on the Accessory Power line of the 30-pin connector must be 5 mA or less (low power mode). Accessories using the intermittent high power option can consume up to a total of 100 mA while music is playing or during other high power modes, such as voice recording. | ON<br>Maximum current draw is 5 mA or less (low power mode). Simple remote accessories must consume less than 10 μA. High power mode is disabled. | OFF | OFF |
| **iAP transports** | All iAP transports are usable. | The iPod will not transition into this state if attached to an active USB host accessory or to a USB accessory supplying $V_{BUS}$ power to the iPod. | None. | None. |
| **Transitions** | To the Sleep state, when any of the following occurs:<br>■ The simple remote "Off" button is pressed or released.<br>■ The play/pause button is pressed and held for 2 seconds.<br>■ The Apple device is idle (no track is playing) for 2 minutes.<br>■ The Apple device is displaying the battery icon and the external power source is removed.<br>The Apple device will not transition out of the Power On state when an external power source is connected or a track is playing. | To the Power On state, when any of the following occurs:<br>■ An external power source is connected.<br>■ The user presses a button on the front panel of the iPod.<br>■ A Simple Remote lingo command other than "Off" is sent.<br>To the Hibernate state when the appropriate amount of time expires. See Table 2-14 (page 45) for the duration of the Sleep state before transitioning to the Hibernate state. | To the Power On state, when any of the following occurs:<br>■ An external power source is connected.<br>■ The Menu or Select button on the front panel of the Apple device is pressed.<br>■ The Accessory Detect pin goes from floating to grounded; see "Accessory Detect and Identify" (page 36).<br>■ The Accessory sends Bluetooth data to the Apple device. | To the Power On state, when external power is provided to an iOS device. |

| Power state | Power On | Sleep | Hibernate | Off |
|---|---|---|---|---|
| **Transition notification (see Note below)** | To Sleep: 0x03<br>To Hibernate: 0x03 and 0x02 | To Power On: 0x04<br>To Hibernate: 0x01 or 0x02 | None | None |

At each transition the Apple device sends a notification to the accessory, specifying the transition type. See General lingo command 0x23, `NotifyiPodStateChange`, in *MFi Accessory Firmware Specification*. No notifications are sent over Bluetooth.

The current that the Apple device supplies on the Accessory Power line of the 30-pin connector is completely shut off when an Apple device enters the Hibernate state. When waking from Sleep state the iPod requires the accessory to identify itself. See General lingo command 0x00, `RequestIdentify`, in *MFi Accessory Firmware Specification*. On reset or power up, the accessory must not consume more than 5 mA (low power mode). The minimum time between an Apple device notification of entering a power-off mode (such as Hibernate), and entry into that state is 100 ms. The minimum time between an Apple device notification of ending recording or playback mode and the associated reduction in the current that the Apple device supplies on the Accessory Power line of the 30-pin connector is also 100 ms.

**Note:** When waking from Sleep state the iPod requires the accessory to identify itself. See General lingo command 0x00, `RequestIdentify`, in *MFi Accessory Firmware Specification*. Accessories that communicate via UART, including self-powered ones such as automotive head units or docks, must always monitor the Accessory Power line to detect sleeping/waking transitions, and they must wait until 80 ms after it has gone high before trying to communicate with the attached Apple device. See "Accessory Signaling and Initialization" in *MFi Accessory Firmware Specification*.

USB and Bluetooth accessories need not monitor the Accessory Power line. USB host accessories must wait for the USB enumeration process to finish and retry communicating with the Apple device until communication is established. After that, retries are need only after command timeouts. USB accessories that place an Apple device in Host mode can initiate communication after the Apple device has started polling the interrupt IN pipe; see "USB Host Mode" (page 58).

## Sleep State

Some Apple devices remain in the Sleep state for various periods, as shown in Table 2-14 (page 45).

**Table 2-14** Sleep times before hibernation

| Apple device model | Time |
|---|---|
| 3G iPod | 36 hours |
| iPod mini | 36 hours |
| 4G iPod | 36 hours |
| iPod photo, 4G iPod (color display) | 36 hours |
| 1G iPod nano | 14 hours |

| Apple device model | Time |
|---|---|
| 5G iPod | 14 hours |
| 2G nano | 30 minutes |
| iPod classic | 30 minutes |
| 3G nano | 30 minutes |
| 4G nano | 10 minutes |
| 5G nano | 10 minutes |
| 120 GB classic | 30 minutes |
| All other Apple devices | N/A |

The trend in Apple device design is for shorter sleep times before the device hibernates. In the 2G iPod nano, iPod classic, and 3G nano, the time is 30 minutes. Accessories must not be designed to draw power from iPods in Sleep state, because such power will be shut off when the Apple device hibernates. With the iPhone, device power may be shut off at any time.

## Hibernate State

In Hibernate mode, the Apple device turns off the current that the Apple device supplies on the Accessory Power line of the 30-pin connector and only responds to iAP commands over Bluetooth.

iOS devices enter the Hibernate state directly from the Power On state. With USB and UART accessories they generate a Sleep state notification immediately before the Hibernate notification, to be compatible with accessories that expect such behavior. See General lingo command 0x23, `NotifyiPodStateChange`, in *MFi Accessory Firmware Specification*. With wireless iAP accessories, iOS devices do not generate a sleep notification. Every Apple device from the iPod nano onward preserves menu selections and playback context during hibernation.

**Note:** For accessories that identify themselves as supporting the Simple Remote or Remote UI lingoes, the time until Hibernate mode is extended. If the simple remote accessory sends button status commands to an Apple device every hour or so, the Apple device will never hibernate.

The transition to Hibernate state behaves differently in the 3G iPod, iPod mini, and 4G iPod than in iPhones and later Apple devices. In hibernation, these four early products lose their menu selection and playback environment information; when they transition to the Power On state, their UIs display only the topmost menu. iPhones and later Apple devices save menu selection and playback environment information. When they transition back to the Power On state, the previous menu and track selection are restored, so users can resume their currently playing playlist and track position exactly where they left off.

Self-powered accessories can wake a hibernating Apple device by providing USB power, and it will stay awake as long as USB power is applied. The only other way to wake a hibernating Apple device remotely is to create a transition on the Accessory Detect pin (pin 20 of the 30-pin connector) from floating to ground. The Accessory Detect pin must remain floating for at least 200 ms before transitioning to ground. Accessories must not toggle the Accessory Detect line during normal, powered operation.

## Power Off State

iOS devices can be turned off by the user. When in the Off state, an accessory can turn them on only by grounding the Accessory Detect pin, as described below. The following actions are ineffective:

- Menu button presses
- Signals from a headphone accessory (see "Headphone Remote and Mic System" (page 71))
- Bluetooth messages
- Incoming phone calls

# Accessory Power Policy

Any accessory that is connected to an Apple device's 30-pin connector and draws power from the Apple device must comply with the requirements in this section.

All accessories must operate in either Low Power mode or Intermittent High Power mode. These power modes are specified in the next sections.

## Low Power Mode

Table 2-15 (page 47) specifies how much current an accessory in Low Power mode may draw from an Apple device, depending on which iAP transport the accessory is using.

**Table 2-15** Maximum current draw in Low Power mode

| iAP Transport | Maximum Current Draw | Notes |
|---|---|---|
| None | 0 mA | |
| UART | 5 mA | |
| USB Device mode | 0 mA | |
| USB Host mode | 10 mA | Some Apple devices support a maximum current draw of only 5 mA. Accessories that need to draw more than 5 mA must verify that the Apple device supports 10 mA. See General lingo command 0x4B, `GetiPodOptionsForLingo`, in *MFi Accessory Firmware Specification*. |
| Bluetooth | 0 mA | |

All accessories must start the authentication process in Low Power mode. During the Authentication process, an accessory may draw the current required to power the Apple Authentication Coprocessor in addition to the Low Power mode limit specified in Table 2-15 (page 47). The authentication process begins at the rising edge of the Apple device's Accessory Power output and ends 500 ms after the Apple device acknowledges successful authentication. See "Accessory Signaling and Initialization" in *MFi Accessory Firmware Specification*.

## Intermittent High Power Mode

When an accessory is in Intermittent High Power mode it may draw up to 100 mA from the Apple device for limited periods, but it is strongly advised that it draw less than 30 mA. At other times it must operate in Low Power mode, as specified in "Low Power Mode" (page 47). This limit applies to the totality of accessories and accessory features that are permitted to operate in Intermittent High Power mode at one time.

## Entering Intermittent High Power Mode

An accessory may enter Intermittent High Power mode on receipt of any of the iAP commands listed in Table 2-16 (page 48).

**Note:** The iAP commands and events cited in this and the next section are described in *MFi Accessory Firmware Specification*.

**Table 2-16** iAP commands that permit Intermittent High Power mode

| Lingo | iAP Command | State | Notes |
|---|---|---|---|
| General | `OpenDataSession-ForProtocol` | | Used with External Accessory Framework |
| Microphone | `iPodModeChange` | Begin audio recording mode | Deprecated |
| | `iPodModeChange` | Begin audio playback mode | Deprecated |
| RF Tuner | `SetTunerCtrl` | Turn RF tuner accessory power draw on | |
| Location | `SetDevControl` | Accessory GPS radio power, Power on | |

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 USB events listed in Table 2-17 (page 48).

**Table 2-17** USB events that permit Intermittent High Power mode

| USB Device Type | Event |
|---|---|
| Audio | The Apple device selects a nonzero bandwidth interface setting |
| MIDI | The Apple device starts polling a MIDI Streaming IN endpoint |

## Exiting Intermittent High Power Mode

All accessories must enter Low Power mode within 1 second after all of the iAP commands that permitted Intermittent High Power mode are negated by one or more of the iAP commands listed in Table 2-18 (page 49).

**Table 2-18** iAP commands that cancel Intermittent High Power mode

| Lingo | iAP Command | State | Notes |
|---|---|---|---|
| General | `CloseDataSession` | | Used with an External Accessory Framework |
| | `NotifyiPodStateChange` | Sleep or Hibernate | Negates all other iAP commands |
| Microphone | `iPodModeChange` | End audio recording mode | Deprecated |
| | `iPodModeChange` | End audio playback mode | Deprecated |
| RF Tuner | `SetTunerCtrl` | Turn RF tuner accessory power draw off | |
| Location | `SetDevControl` | Accessory GPS radio power, Power off | |

When an Apple device is in USB Host mode, accessories in Intermittent High Power mode must also enter Low Power mode within 1 second after all of the USB commands that permitted Intermittent High Power mode are negated by one or more of the USB events listed in Table 2-19 (page 49).

**Table 2-19** USB events that cancel Intermittent High Power mode

| USB Device Type | Event |
|---|---|
| Audio | The Apple device selects a zero bandwidth interface setting |
| MIDI | The Apple device stops polling a MIDI Streaming IN endpoint |

# Command and Data Transports

An accessory may exchange iAP commands and data of specific types with an Apple device using UART, USB Device Mode, USB Host Mode, or Bluetooth transports. This chapter specifies the accessory hardware required to support these transport layers.

**Note:** To determine which transports are supported by specific Apple devices, see Table 2-1 (page 25).

Every accessory must restrict its iAP traffic to one transport at a time. If an accessory needs to move its iAP connection from one transport to another, it must cease all iAP traffic on the first transport and then re-identify on the second before resuming iAP communication. Except for the General and Simple Remote lingoes, lingoes can be used by only one connected accessory at a time. Attempting to use a lingo that is already in use by another accessory will cause a “Maximum number of accessory connections already reached” error during identification. For further information, see General lingo command 0x02 (`iPodAck`) in *MFi Accessory Firmware Specification*.

## UART

The iPod Accessory Protocol builds on the RS-232 serial specification; however, the Apple device’s signaling levels are not standard. The RS-232 standard specifies that a mark is –7 V and a space is +7 V. In the Apple device protocol, a mark is 2.500 through 3.465 V and a space is 0 through 0.8 V. Voltages and currents are shown in Table 3-1 (page 51), where positive output currents flow out of the Apple device.

**Table 3-1** Apple device mark and space levels

| Description | Symbol | Conditions | MIN | MAX | Units |
|---|---|---|---|---|---|
| Input Voltage High | $V_{IH}$ | | 2.500 | 3.465 | V |
| Input Voltage Low | $V_{IL}$ | | 0.000 | 0.800 | V |
| Input Current | $I_I$ | $V_I$ = 0 V or 3.0 V | | ±30 | µA |
| Output Voltage High | $V_{OH}$ | $I_{OH}$ = 100 µA | 2.500 | 3.465 | V |
| Output Voltage Low | $V_{OL}$ | $I_{OL}$ = –100 µA | 0.000 | 0.500 | V |

The accessory must not rely on the Apple device to hold the accessory RX level to the mark state when the UART transport is idle. When idle, the accessory’s RX line should be pulled up to the Accessory Power level (pin 13 of the 30-pin connector) through a minimum value of 10 kΩ; 100 kΩ is recommended.

Accessories that use an open-collector or open-drain UART driver on their TX line should include a pull-up resistor to the Apple device's Accessory Power (pin 13 of the 30-pin connector). The resistor value should be chosen to meet the UART timing requirements described below. The maximum mark voltage shown in Table 3-1 (page 51) must be measured with the Apple device not connected, and mark signals must be sent only after the current that the Apple device supplies on the Accessory Power line of the 30-pin connector is present.

Accessories must communicate with all Apple devices at a nominal baud rate of either 19200 bps or 57600 bps and maintain their baud rates within ±2% of the chosen rate over the entire temperature range of the accessory. The temperature range of the accessory must be greater than or equal to the temperature range of the Apple device (0–35° C). Once an accessory has started communicating with an Apple device, it cannot change its baud rate.

**Note:** Current Apple devices are capable of automatic baud rate detection (autobauding) between 9600 bps and 24000 bps, as shown in "General Apple Device Features" in *MFi Accessory Firmware Specification*. However, this feature may not be available in future Apple devices; newly designed accessories must communicate only at either 19200 bps or 57600 bps.

When using UART transport, every packet must begin with a sync byte (0xFF) to ensure that the automatic baud rate detection is accurate. If the Apple device may be sleeping, another sync byte must also be sent 5 ms before the packet to wake it.

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.

# USB

Apple devices are USB 2.0–compliant devices that can interact with third-party accessories in either of two mutually exclusive modes:

- As a USB device, described in "USB Device Mode" (page 52).
- As a USB host, described in "USB Host Mode" (page 58).

**Note:** The firmware requirements for using USB Host Mode and USB Device Mode are set forth in *MFi Accessory Firmware Specification*.

## USB Device Mode

When used as a USB device, an Apple device supports two modes of operation:

- Mass storage device. This is the default configuration when attached to a typical USB host such as a PC or Macintosh. This mode is used for syncing music and other content, transferring files, and so forth.

- An iAP-enabled USB device that uses the iPod USB Interface (iUI). The accessory must select this mode before it can be used.

**Note:** Newly-designed accessories must not use an Apple device as a mass storage device. Accessories should always use the iUI to interact with an Apple device.

The two USB Device modes of operation (mass storage and iUI) are each represented by a USB Configuration. When an Apple device is attached to USB, the USB host (the accessory) must select one of the configurations and set it as the active configuration during the bus initialization. Alternatively, an accessory can use a resistor to make the iUI the default configuration, as shown in "Accessory Detect and Identify" (page 36).

Although Apple devices support Full-Speed as well as High-Speed bus operation, High-Speed hosts should be used for better data throughput.

## Choosing an Apple Device USB Configuration

The initialization and configuration of an attached USB device is documented in the USB 2.0 specification. This document does not cover this topic in detail, but instead provides information specific to Apple devices. To distinguish an Apple device, a USB host can check the device descriptor of attached USB devices for the following fields:

- Vendor ID: 0x05AC
- Product ID: 0x12*nn*

Note that product IDs vary, depending on the type of Apple device. There will be an expanding list of Apple devices that support iUI. Although an Apple device may make string descriptors available that identify its manufacturer, product name, serial number, configuration, and so on, the accessory must not use these strings to determine whether the connected USB device is an Apple device. The strings may change in future Apple devices. For iPod shuffle identification, see Apple's *iPod shuffle Interface Specification*.

**IMPORTANT:** A USB Device mode accessory must use the Vendor ID and *only* the most significant byte of the Product ID to detect the presence of an Apple device. It can complete the identification by checking for the presence of an iUI configuration on the detected Apple device.

Figure 3-1 (page 54) shows the USB configuration and interface descriptors in Apple devices that do not support USB audio. Figure 3-2 (page 54) shows the USB Configuration and interface descriptors in Apple devices that do support USB audio.

**Figure 3-1** Configuration and interface descriptors for Apple devices without USB audio

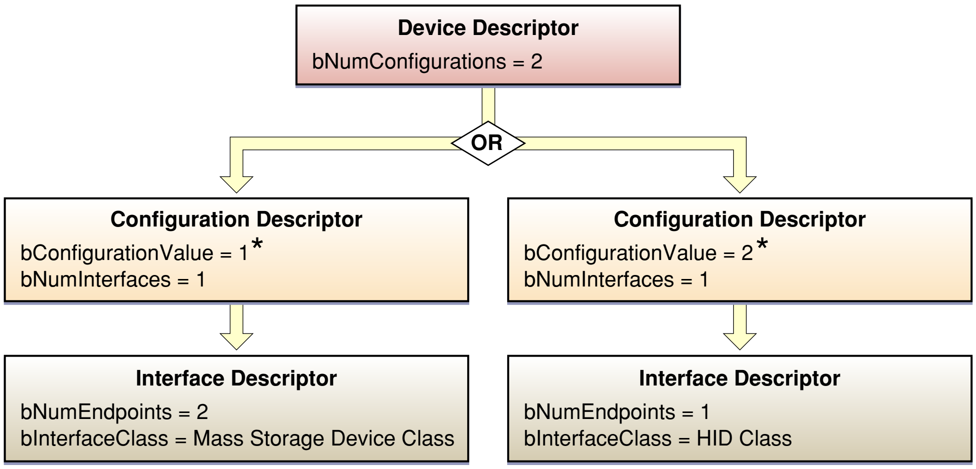
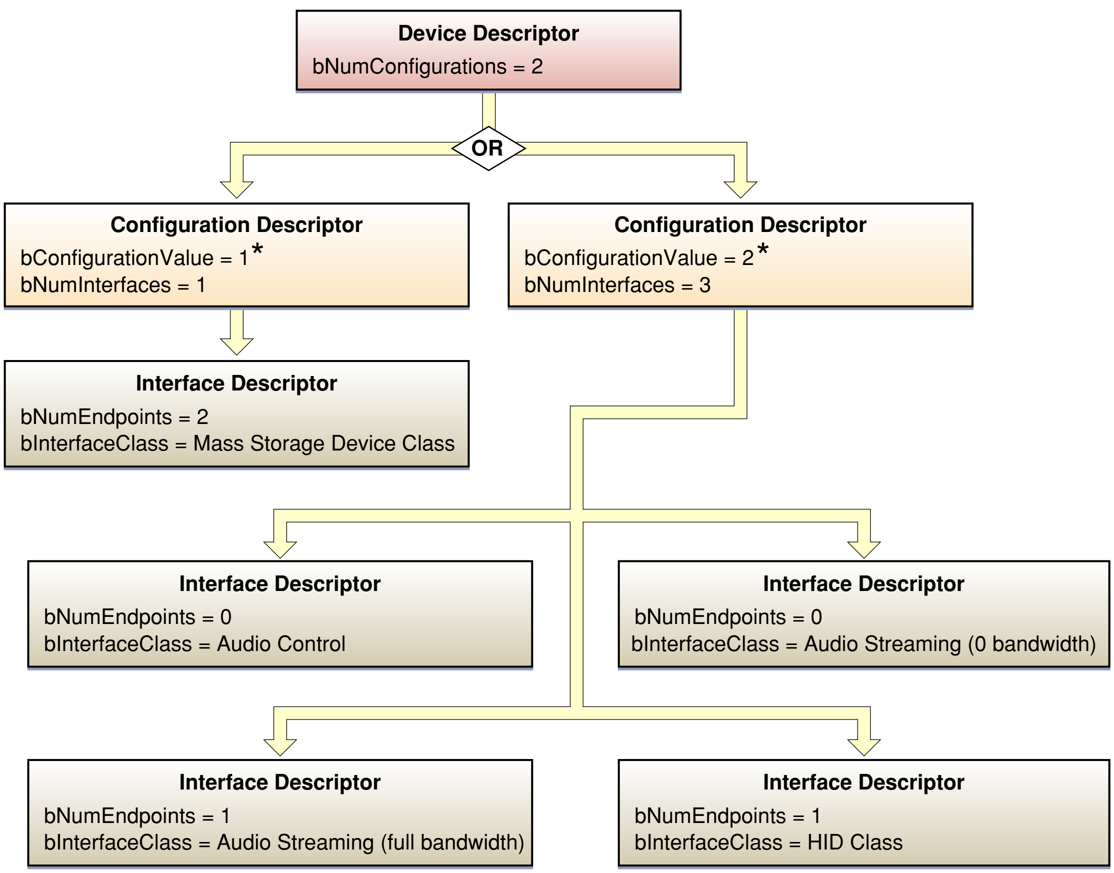



**Figure 3-2** Configuration and interface descriptors for Apple devices with USB audio




* These numbers distinguish different USB configurations.

## iPod USB Interface (iUI) Configuration

The iUI configuration allows an Apple device to communicate via iAP over USB and enables USB digital audio on supported models. The USB Human Interface Device (HID) interface is the transport 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.

**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 USB host devices 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 host (output reports) and some of which are used to transport data to the host (input reports). In order to send data to an Apple device, a host chooses one or more appropriately sized HID reports in which to embed the iAP packet and sends this to the Apple device HID interface with USB `Set_Report` commands. The Apple device reassembles the iAP packet and processes it. The process is repeated in reverse when the Apple device sends responses or iAP packets to the host. In this case, the data is sent on an interrupt pipe 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.

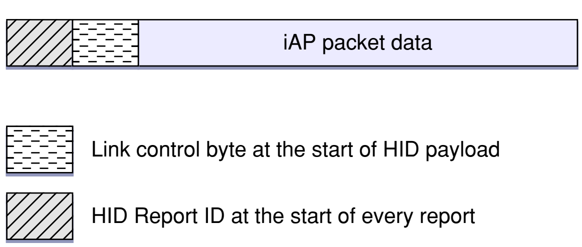
**IMPORTANT:** Accessories must always request and parse the HID report descriptor each time an Apple device is connected or the accessory resets USB, because the HID Report ID and size descriptions may change.

## HID as a Transport

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 an Apple device's vendor-specific HID report is a link control byte (LCB), followed by iAP packet data. An example is shown in Figure 3-3 (page 55).

**Figure 3-3** Apple device vendor-specific 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 should 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, as described in Table 3-2 (page 56).

**Table 3-2** 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.<br>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.<br>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.

Figure 3-4 (page 57) shows the different report packing scenarios that are possible, including one packet per report and multiple reports per packet.

**Figure 3-4** Possible report packing scenarios

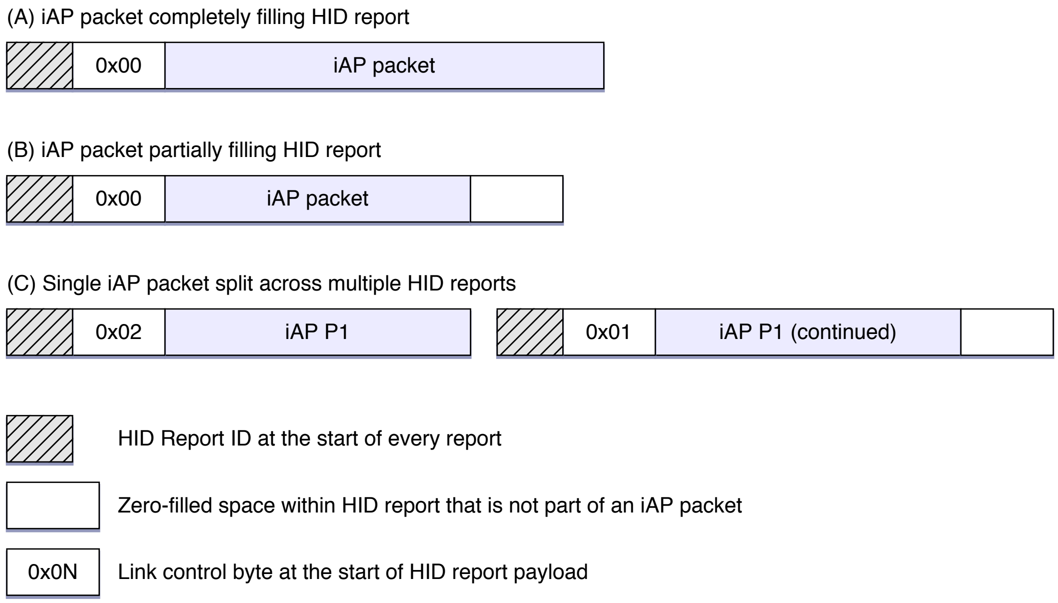
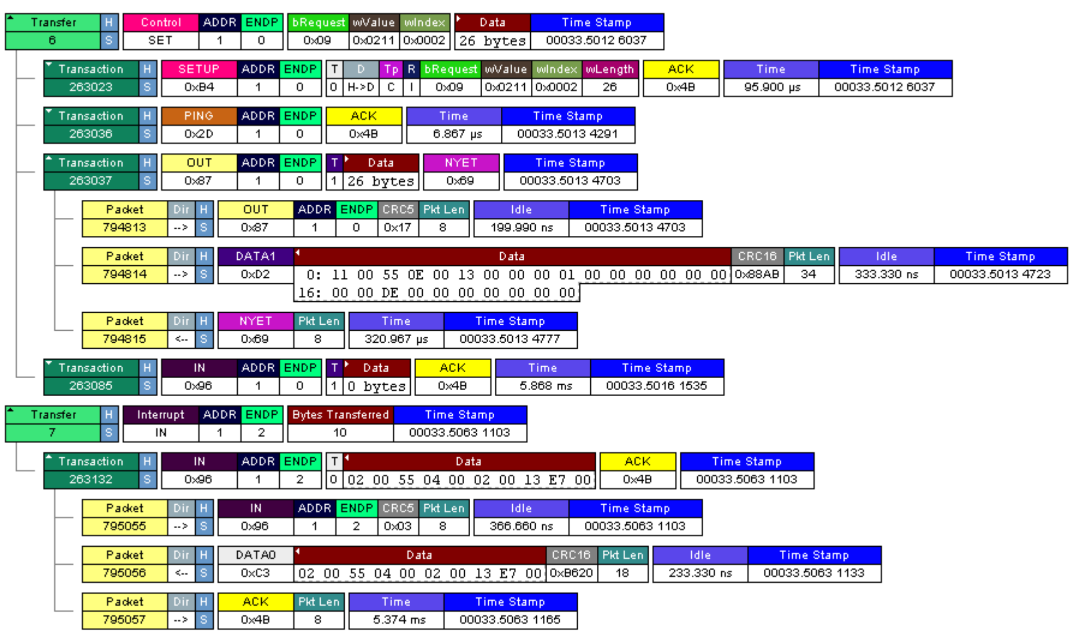



Figure 3-5 (page 57) illustrates a sample USB transport interchange that uses iAP to send an `IdentifyDeviceLingoes` command and receive back an `ACK` command. `IdentifyDeviceLingoes` is described in "Command 0x13: IdentifyDeviceLingoes" in *MFi Accessory Firmware Specification,* Lingo 0x00; the corresponding `ACK` command is described in "Command 0x2: ACK" in *MFi Accessory Firmware Specification,* Lingo 0x00. For clarity, the diagram omits SOF and NAK messages.

**Figure 3-5** Transferring `IdentifyDeviceLingoes` and `ACK` commands over USB



### Apple Device Sleep Behavior When Attached to a USB Host

An Apple device will not transition to Sleep mode while it is attached to an active USB host or is itself in USB Host Mode. To ensure that a physically attached Apple device goes to sleep, a host system must pause media playback before powering down its host controller. Attaching USB power to a sleeping Apple device wakes it up.

## USB Host Mode

Certain Apple devices (see Table 2-1 (page 25)) can operate in USB Host mode, an operating mode in which the Apple device acts as a USB bus host and the accessory acts as a USB device. In this mode the accessory can control the Apple device by sending and receiving iAP commands over USB.

To put a connected Apple device into USB Host mode using hardware, the accessory must provide a 28 kΩ ±1% $R_{ID}$ resistor between pin 10 (Accessory Identify) of the 30-pin connector and ground (pins 1, 2, 15, and 16). For information about $R_{ID}$ resistors, see Table 2-10 (page 37). This resistor to ground must replace any other connection to pin 10.

Alternatively, the accessory can put an Apple device into USB Host mode using iAP commands, as described in *MFi Accessory Firmware Specification*, Release R40 or later.

**Note:** Once the Apple device is in USB Host mode it will remain in that mode regardless of whether the USB $V_{BUS}$ supply is turned on or off.

The accessory can determine that the Apple device has successfully entered USB Host mode by detecting that it has begun to poll the USB interrupt IN pipe. This means that the Apple device is ready to accept iAP commands over USB.

### General Requirements

Accessory designs must observe these general requirements for using USB Host mode with an Apple device:

- If the Apple device is an iOS device, it must be running iOS 3.2 or later.
- Only certain Apple devices can use the USB Host Mode lingo (Lingo 0x06).
- The accessory must connect to the Apple device through its 30-pin connector.
- The Apple device does not provide the accessory with +5 V nominal $V_{BUS}$ power. Instead, the 3.3 V power that the Apple device supplies on its Accessory Power line (pin 13 of the 30-pin connector) is available as specified in "Accessory Power" (page 35).
- The accessory must comply with the general USB specification, version 2.0 or later, plus any applicable device-specific USB specifications that are available at www.usb.org/developers/docs/.
- The Apple device requirements stated in "Accessory Power Policy" (page 47) override any conflicting power requirements in the USB specifications.
- All USB descriptors (particularly the Endpoint descriptors and the `bMaxPower` field of the Configuration descriptors) must accurately represent the accessory's capabilities.

- Every different accessory's device descriptors must include a unique vendor ID (VID) assigned by USB-IF and a unique product ID (PID) assigned by the vendor. The USB-IF vendor ID must be assigned to the MFi licensee responsible for the accessory.
- All USB Device, Configuration, and Interface descriptors must be accompanied by human-readable String descriptors. Among these String descriptors, the following must match `AccInfoToken` string values that the accessory passes to the Apple device during the IDPS process (see table "Accessory Info Type values" in section "Command 0x39: SetFIDTokenValues" of *MFi Accessory Firmware Specification*):
  - USB Manufacturer String Descriptor must match IDPS Accessory manufacturer
  - USB Product String Descriptor must match IDPS Accessory name
  - USB Serial Number String Descriptor must match IDPS Accessory serial number
- Upon receiving a USB Suspend command the accessory must immediately enter Low Power Mode, as defined in "Accessory Power Policy" (page 47), and remain in that mode until it receives a USB Resume command.

For further information about using USB Host mode, including packet formats, data transfers using the USB IN and OUT pipes, and exchanging iAP commands over USB, see *MFi Accessory Firmware Specification*, Release R42 or later. For USB accessory cable specifications, see "Power and Data" (page 31).

## Using iAP

Accessories must observe these requirements when exchanging iAP commands with an Apple device in USB Host mode:

- Accessories that use iAP over USB must implement USB high-speed or full-speed bulk IN/OUT pipe endpoints and the USB interrupt IN pipe endpoint. The interrupt IN pipe endpoint must specify a polling interval between 4 and 32 ms.
- The accessory cannot use the Digital Audio lingo in USB Host Mode.

When an accessory places an Apple device in USB Host mode, the Apple device enumerates USB devices with the properties shown in Table 3-3 (page 59).

**Table 3-3** USB Device Enumeration for Apple devices

| USB Descriptor | Value | Comments |
|---|---|---|
| Interface | 0x00 | |
| Interface Class | 0xFF | Vendor-specific interface |
| Interface Subclass | 0xF0 | |
| Interface Protocol | 0x00 | |
| Interface String | `"iAP Interface"` | |

## USB Audio

After authentication, an Apple device in USB Host Mode can configure USB audio OUT and audio IN stream endpoints present on the USB device. The Apple device can then interact with accessories that are compliant with either the USB Audio 1.0 or 2.0 specification. Version 2.0 is recommended.

**Note:** 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.

**Note:** Not all Apple devices that support USB Host Mode also support USB audio input/output. Before trying to use USB audio the accessory must verify that the Apple device supports it. See General lingo command 0x4B, `GetiPodOptionsForLingo`, in *MFi Accessory Firmware Specification*.

An accessory using USB audio must support the following USB features:

- 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.
- 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; see developer.apple.com/library/mac/#technotes/TN2010/tn2274.html).
- 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 using USB audio, but 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.

Developers should test their accessory designs against the latest Mac OS X Audio driver; the application Audio MIDI Setup, in the Mac OS X `Applications/Utilities` folder, can be used to verify accessory compatibility.

### USB MIDI

Starting with iOS 4.2.1, some Apple devices provide support for USB MIDI. To access this support, 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 Mac OS X MIDI driver; the application Audio MIDI Setup, in the Mac OS X `Applications/Utilities` folder, can be used to verify accessory compatibility.

> **IMPORTANT:** Every accessory that provides or receives any kind of MIDI data to or from an Apple device must do so using only USB MIDI transport in compliance with the USB MIDI specification.

Every accessory that supports USB MIDI must implement a MIDI Streaming IN endpoint if it needs to enter Intermittent High Power Mode, as defined in "Accessory Power Policy" (page 47).

### USB HID

The iOS devices marked in Table 2-4 (page 28) support USB Human Interface Device (HID) class 1.11 keyboards. The USB HID Consumer Page controls that iOS devices support are listed in the "USB Human Interface Device Reports" section of the Simple Remote lingo documentation in *MFi Accessory Firmware Specification*. Accessories that use USB HID must conform to the USB *Device Class Definition for HID* specification and also to the following MFi requirements:

- The accessory must be a keyboard; no other USB HID device types are allowed.
- If the accessory is localized, its HID descriptor must declare one of the country codes listed in the section "Command 0x0F: RegisterDescriptor" of the Simple Remote lingo documentation in *MFi Accessory Firmware Specification*.

# Bluetooth

The iOS devices listed in Table 2-1 (page 25) as supporting iAP over BT can send and receive iAP commands over their Bluetooth transports. This feature is in addition to their support of the existing UART and USB transports. Accessories may use Bluetooth to communicate wirelessly with these devices, instead of using the 30-pin hardware connectors. Starting with iOS 5.0, multiple iAP over Bluetooth accessories can also be connected simultaneously.

## General Requirements

Accessories that communicate with iOS devices using Bluetooth must meet the requirements specified in the Apple document *Bluetooth Accessory Design Guidelines for Apple Products*. That document covers the Bluetooth profiles for various iOS devices and other aspects of accessory design that are not specifically related to iAP. The information presented there is crucial for obtaining satisfactory Bluetooth communication between accessories and Bluetooth-capable iOS devices. When incorporating that document in this specification, substitute "must" for "should" throughout.

Accessories that receive iAP commands using Bluetooth must set the Device Name in their Bluetooth Extended Inquiry Response (EIR) packet to the Accessory Name string value that they previously passed to the Apple device in an `AccessoryInfoToken` during the IDPS process; see "Identification" in *MFi Accessory Firmware Specification*.

## Setting Up Bluetooth Communication for iAP

Bluetooth accessories must support the Bluetooth Service Discovery Protocol (SDP) to be discovered by an iOS device and establish a connection with it. The maximum transmission unit (MTU) for SDP data should be at least 672 bytes. The accessory must not fragment the SDP record.

> **Specification Update:** Most accessories no longer need to declare any specific Bluetooth Class of Device or Major Service to exchange iAP commands with Apple devices. The sole exception is accessories that use the Bluetooth autopairing feature, which must set the Audio/Video bit in the Class of Device field. See "Bluetooth Autopairing and Connection Status Notifications" in *MFi Accessory Firmware Specification*.

To be discovered by an Apple device, an accessory must set a Service UUID of 0x00000000DECAFADEDECADEAFDECACAFF in both its Service Discovery Protocol (SDP) record and its Extended Inquiry Response (EIR) packet.

Following are some general recommendations for a successful Bluetooth connection:

- Implement Bluetooth Sniff Mode, or Sniff Subrating if Bluetooth 2.1 is being used.
- Let the iOS 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 iAP layer.

The iOS device may refuse a Bluetooth connection under these circumstances:

- There are too many Bluetooth connections made to the iOS device. In this case, the accessory receives an error indicating that a resource is unavailable.
- There are too many RFCOMM protocol connections to the iOS device. In this case, the accessory receives a Resource Denied error when trying to connect to the Wireless iAP service.

To re-establish a connection to the iOS device, the accessory must make a Bluetooth SDP query to find the RFCOMM channel associated with the UUID 0x00000000DECAFADEDECADEAFDECACAFE, then connect to that channel. The accessory must not assume that the channel will remain the same between connections. The connection will require new authentication if no link key is present.

The accessory must not expect that the iOS device will try to re-establish a broken Bluetooth connection.

## Using Bluetooth Transport for iAP

After Bluetooth communication is established, using the RFCOMM protocol, the accessory may send iAP General lingo commands with command IDs in the ranges 0x00-0x19 and 0x23-0x2B without authentication. All other iAP commands require Authentication 2.0, as described in *MFi Accessory Firmware Specification*.

> **Note:** Authentication 1.0 is not supported for Bluetooth.

To begin the identification and authentication process, new accessory designs must use the Identify Device Preferences and Settings (IDPS) process described in the "Accessory Identification" appendix of *MFi Accessory Firmware Specification*. This will maximize their compatibility with future firmware. Existing accessory designs may continue to use the process described in General lingo command 0x13, `IdentifyDeviceLingoes`, in *MFi Accessory Firmware Specification*.

> **Note:** The following rules apply to all Bluetooth accessories:
>
> - Hibernate notifications are not sent when the Apple device enters the Hibernate power state, nor will the Apple device request accessory identification when it reenters the Power On state. The accessory may still receive identification requests for other reasons, so it must handle them. See General lingo command 0x00, `RequestIdentify`, in *MFi Accessory Firmware Specification*.
> - Accessories must ignore power state notifications from the Apple device.
> - Bluetooth transmissions to the Apple device will cause it to exit the Hibernate state.
> - Accessories should generate Bluetooth traffic only in response to a direct user action.

Once connected and authenticated, an accessory can use the lingoes listed in Table 3-4 (page 63) to communicate with the iOS device with which it is paired. For descriptions of these lingoes, see *MFi Accessory Firmware Specification*.

**Table 3-4** Lingoes accessible through Bluetooth

| Lingo | ID | Notes |
|---|---|---|
| General | 0x00 | Must be included in the IDPS tokens or the `IdentifyDeviceLingoes` command. |
| Simple Remote | 0x02 | Bluetooth accessories that support these lingoes must have their own volume controls and not use iAP volume control commands. They must also implement Extended Interface lingo playback status notifications, as specified in Release R41 of *MFi Accessory Firmware Specification*. |
| Display Remote | 0x03 | |
| Extended Interface | 0x04 | |
| Sports | 0x09 | |
| Storage | 0x0C | |
| Location | 0x0E | |

# Accessory Design Requirements

In addition to the requirements set forth elsewhere in this document, this chapter summarizes some of the hardware requirements that accessories must meet under the MFi licensing program. For details of accessory firmware requirements, see Apple's *MFi Accessory Firmware Specification*.

## General Requirements

Certain testing and design requirements apply to all accessories that are connected to, or communicate with, Apple devices.

### Audio Output From Accessories

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.

### Video Output From Accessories

Any accessory that accepts video output from an Apple device for viewing on a display must be designed to present that video for viewing at all times without requiring the user to switch viewing modes or reconfigure the accessory.

### Power Requirements

Accessories may contain internal or external power supplies for iOS devices, but their design must follow the specifications in "iOS Device Power Supply Requirements" (page 85) to avoid interfering with the sensor operation of the touch screen.

In addition, accessory power supplies must meet the electrical certification requirements described in "Electrical Testing and Certification" in *MFi Accessory Testing Specification* and must also pass the RF certification tests described in "Measuring TRP" in *MFi Accessory Testing Specification* with power off, and "Measuring EIS" in *MFi Accessory Testing Specification* with power on, using the setup specified in "Typical Test Setup for Cable-Connected Accessories" in *MFi Accessory Testing Specification*.

A speaker system designed to work with an Apple device should charge the device's battery while it plays its sound.

## Car Charger Design Requirements

Car chargers must meet the electrical certification requirements described in "Electrical Testing and Certification" in *MFi Accessory Testing Specification*, including "Car Charger Tests" in *MFi Accessory Testing Specification*. They must also pass the RF certification tests described in "Measuring TRP" in *MFi Accessory Testing Specification* with power off, and "Measuring EIS" in *MFi Accessory Testing Specification* with power on, using the setup specified in "Typical Test Setup for Cable-Connected Accessories" in *MFi Accessory Testing Specification*.

Car chargers for the iPhone must charge via USB.

## Battery Pack Design Requirements

All battery pack accessories must meet the RF certification requirements described in "RF Testing and Certification" in *MFi Accessory Testing Specification* and the relevant electrical certification requirements described in "Electrical Testing and Certification" in *MFi Accessory Testing Specification*. If the battery pack accessory includes any audio or video features, it must also pass the TDMA noise tests described in "TDMA Noise Testing and Certification" in *MFi Accessory Testing Specification*.

Battery pack accessories for the iPhone must supply power via USB.

## Magnetic Fields From Accessories

Accessories must minimize interference with the digital compass inside the Apple device and must not repeatedly trigger compass recalibration. The Apple device models with digital compasses are identified in Table I-1 (page 12).

## USB Cables

All cables that are included with an accessory, that terminate in at least one USB connector, and that do not terminate in an Apple 30-pin connector, must meet or exceed all applicable USB-IF specifications.

## Cases for Apple Devices

To pass MFi certification, cases that enclose Apple devices must must not be integrated with keyboards and must comply with the guidelines stated in the latest version of Apple's *Case Design Guidelines for Apple Devices*. When incorporating that document in this specification, substitute "must" for "should" throughout.

# iPhone-Specific Requirements

Accessories for the iPhone must meet all the general requirements of accessories for Apple devices. This section provides additional specifications for accessories designed to meet the specific requirements of Apple's iPhone licensing program.

 | 

## Certification of iPhone Accessories

Under the iPhone licensing program, all third-party accessories must pass Apple-specified certification processes, conducted as follows:

- The tests and certification procedures described in "Electrical Testing and Certification" in *MFi Accessory Testing Specification* must be performed.
- The testing and certification procedures described in "RF Testing and Certification" in *MFi Accessory Testing Specification* and "TDMA Noise Testing and Certification" in *MFi Accessory Testing Specification* must be conducted by an independent laboratory and their certification sent to Apple. The laboratory must be certified by Apple. Contact Apple Licensing for information about Apple-certified laboratories.

## RF Transmission and Reception Factors

Accessories for the iPhone need to be designed to avoid specific radio interference problems. Even case developers need to take into consideration the iPhone's antenna and sensor locations.

Accessories for the iPhone are evaluated on two general criteria to determine their RF compatibility with it:

- Reduction of the iPhone's RF/antenna efficiency. Accessories should minimize decreases in the iPhone's total radiated power (TRP). This can be quantified by measuring TRP across all of the iPhone's operating bands and some frequencies. For accessory testing and certification requirements, see "Measuring TRP" in *MFi Accessory Testing Specification*.
- Desense of the iPhone's RF reception. Accessories should minimize decreases in the iPhone's effective isotropic sensitivity (EIS). This can be quantified by measuring EIS across all of the iPhone's operating bands. For accessory testing and certification requirements, see "Measuring EIS" in *MFi Accessory Testing Specification*.

## TDMA Noise Factors

Accessories must minimize coupling of audible interference from the iPhone (commonly known as 'TDMA noise' or 'chopper noise') into an accessory's electronics. The impact of TDMA noise on an existing accessory can be qualitatively assessed by plugging an iPhone into the accessory while an incoming call is received on the handset. For TDMA testing and certification requirements, see "TDMA Noise Testing and Certification" in *MFi Accessory Testing Specification*.

## Speaker System Design Requirements

GSM phones emit radiated and conducted RF noise, which can produce time division multiple access (TDMA) sounds from audio outputs. Speaker systems that work with the iPhone must be designed to reduce or eliminate these unwanted sounds, and they must be tested to ensure that TDMA noise does not affect the user's experience. TDMA acoustic noise from the speaker system while the iPhone is docked in it must be tested and certified as described in "TDMA Noise Testing and Certification" in *MFi Accessory Testing Specification* with the test setup specified in "Typical Speaker System Test Setup" in *MFi Accessory Testing Specification*.

To obtain Apple certification, a speaker accessory for the iPhone must also pass the following tests with the variations noted:

- The RF testing configuration shall be freestanding, as shown in "RF Certification Setup" in *MFi Accessory Testing Specification*.
- In addition to the other iPhone configuration requirements for RF testing, described in "RF Certification Setup" in *MFi Accessory Testing Specification* , the iPhone display must be switched off.
- Total radiated power (TRP) of the iPhone while connected to the accessory must be tested and certified, as described in "Measuring TRP" in *MFi Accessory Testing Specification* , with no AC power applied to the accessory.
- The antenna sensitivity (EIS) of the iPhone while connected to the accessory must be tested and certified, as described in "Measuring EIS" in *MFi Accessory Testing Specification*, with AC power to the accessory tuned on.

## iPod Out Accessory Requirements

Accessories that use the iPod Out feature of Apple devices must have only captive video displays; they must not be able to pass the video output to an external display. In addition, every display used for iPod Out must render the entire iPod Out signal, without underscanning, overscanning, or other modifications. If absolutely necessary, iPodOut can be used in an action-safe mode that allows accessories to drive displays that would not otherwise render the entire iPod Out signal. This action-safe mode can be invoked through the iAP `SetUIMode` command, as described in *MFi Accessory Firmware Specification*. Accessories must meet the foregoing requirements by default, without requiring configuration by the user.

Accessories designed to work with multiple displays, whether used sequentially or in parallel, must meet the foregoing requirements for each display. Samples of all displays must be submitted to Apple as part of each accessory's certification.

Examples of iPod Out display configurations that are not acceptable include (but are not limited to) the following:

- No video display at all.
- Exposed video output terminals to which the user might connect an external display.
- Video displays that the user might disconnect and replace.
- Video displays that do not make visible the entire iPod Out signal with its default settings.

For information about setting and using iPod Out mode, see the General lingo command 0x37 (`SetUIMode`) and the Accessory lingo 0x0D (iPod Out Lingo) in *MFi Accessory Firmware Specification*.

## AssistiveTouch Accessory Requirements

An accessory designed to use the Apple AssistiveTouch feature must declare that fact, using an `AccessoryCapsToken`, during the IDPS process described in *MFi Accessory Firmware Specification*. It must also confirm that the attached Apple device supports the AssistiveTouch feature by sending a `GetiPodOptionsForLingo` command.

Once the Apple device confirms that it supports AssistiveTouch, the accessory must register with it as a USB HID (Human Interface Device) mouse. If the accessory uses Bluetooth as an iAP transport, it must send HID reports to the Apple device using iAP, as specified in "USB Human Interface Device Reports" in *MFi Accessory Firmware Specification*. If the accessory uses USB Host mode as an iAP transport, it may implement a native USB HID mouse device component. See "Appendix E.10: Report Descriptor (Mouse)" in *Device Class Definition for Human Interface Devices 1.11*, (available from USB-IF) for a sample HID descriptor.

The following requirements apply to accessories that use AssistiveTouch:

- 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 have two user-accessible 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.

# Wi-Fi Network Login Sharing Requirements for Accessories

An accessory designed to use the Wi-Fi Network Login Sharing feature documented in Apple's *MFi Accessory Firmware Specification* must implement the following hardware specifications:

- It must let the user select Wi-Fi Network Login Sharing, through either a physical button or an onscreen option.
- It must not be able to initiate Wi-Fi Network Login Sharing without an explicit user selection, as 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.

# Headphone Remote and Mic System

The technology defined in this appendix supports sending button press information from a headset accessory to an Apple device through the headphone connector (see "Headphone/Microphone Jack" (page 21)).

Remote button detection requires a transmitter chip, provided by Apple, that communicates over the accessory's microphone bias line. When implemented with a MEMS microphone as specified in this document, the transmitter chip currently supports three remote buttons in the accessory: volume up, volume down, and push-to-talk.

**Note:** Headset accessories that use Apple's headphone remote and mic system must be tested and certified. The required process is specified in "About Accessory Certification" in *MFi Accessory Testing and Certification Specification*.

This specification covers the transmitter chip that must be included in the headphone accessory and its required external circuitry. The corresponding receiver functionality is implemented by a controller in the iPod nano (4th generation), iPod classic (120 and 160 GB), iPod touch (2nd generation), and the iPod 3G shuffle. It is implemented in the iPhone 3GS, but not in the original iPhone or iPhone 3G.

## Transmitter Chip

This section describes the transmitter chip that must be included in the headset accessory. The external circuitry required to implement the transmitter is described in "Button Detection Circuitry" (page 80).

### Overview

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 decodes the button information from the transmitter chip.

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.

## Part Numbers

Table 5-1 (page 72) shows the part numbers of different versions of the transmitter chip that are available.

**Table 5-1** Transmitter chip part numbers

| Part number | Usage |
|---|---|
| MFI353S2429 | MFi noise-occluding headphones. These are headphones that block or cancel outside sound. |
| MFI353S2430 | MFi standard (nonoccluding) headphones |

**Note:** The accessory designer should perform subjective listening tests to determine which chip produces the best user experience with the accessory configuration. These tests must be performed while the accessory is connected to an iPhone 4.

## Pin Assignments

Table 5-2 (page 72) lists the transmitter chip's pin assignments. Figure 5-1 (page 73) illustrates the transmitter chip package, showing physical pin locations, with physical dimensions shown in Table 5-3 (page 73).

**Table 5-2** 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 |

**Figure 5-1** Transmitter chip package

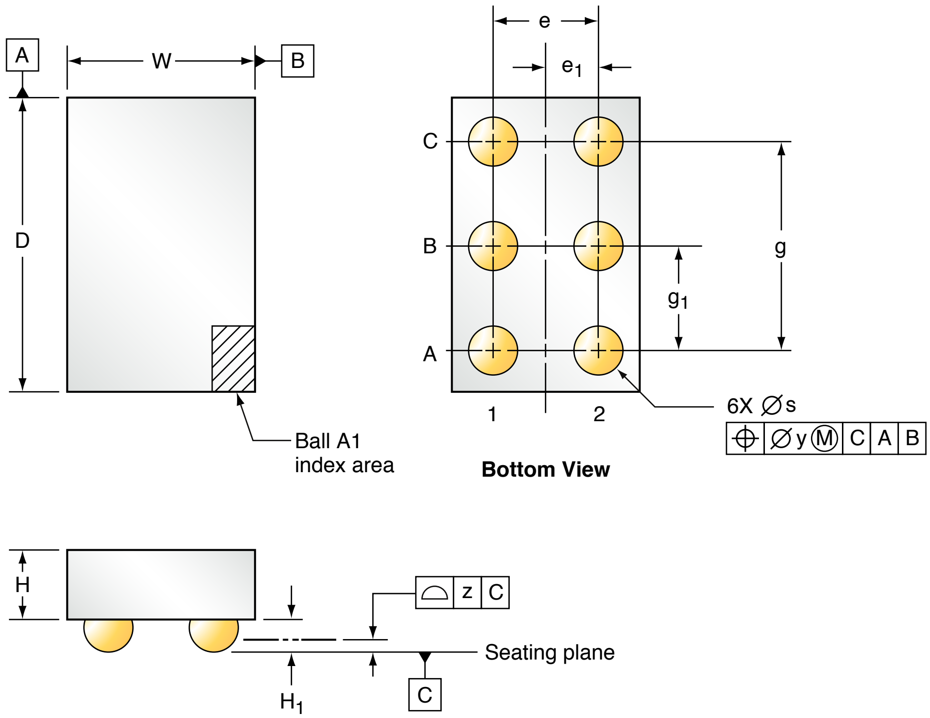



**Table 5-3** Transmitter chip package dimensions

| Dimension | Value in mm |
|---|---|
| W | 0.95/0.85 |
| D | 1.45/1.35 |
| H | 0.50 max |
| $H_1$ | 0.19/0.15 |
| e | 0.50 |
| $e_1$ | 0.25 |
| g | 1.00 |
| $g_1$ | 0.50 |
| s | 0.25/0.21 |
| y | 0.015 |
| z | 0.05 |

## Maximum Voltage and Current Ratings

Table 5-4 (page 74) lists the transmitter chip's maximum voltage and current ratings while operating over a free-air temperature range ($T_A$) of –40 to +85° C. The chip's maximum storage temperature ($T_{stg}$) 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.

**Table 5-4** Maximum voltage and current ratings

| Symbol | Description | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{SUPPLY}$ | Supply voltage, VSHUNT, MIC | –0.5 | 4.6 | V |
| $V_I$ | Input voltage, REM | –0.5 | 4.6 | |
| $V_0$ | Output voltage, MICPWR, TONE | –0.5 | 4.6 | |
| $I_{IK}$ | Input clamp current, REM ($V_I$ < 0) | –20 | | mA |
| $I_{0K}$ | Output clamp current, MICPWR, TONE ($V_O$ < 0) | –20 | | |
| $I_{SUPPLY}$, $I_{GND}$ | Continuous current through VSHUNT, MIC, or GND | –50 | 50 | |

**Note:** All input and output clamp-current ratings must be observed.

## Thermal Impedance

The transmitter chip's package thermal impedance, calculated in accordance with Specification JESD51-7, is 123° C/W.

## Moisture Sensitivity

The transmitter chip has a Moisture Sensitivity Level (MSL) of 1, as defined by industry standard JEDEC specifications.

## Electrical Characteristics

Table 5-5 (page 75) lists the transmitter chip's electrical and timing characteristics under the following conditions:

- Operating temperature = –40 to +85° C.
- Button mode, $V_{MICBIAS}$ = 1.8 to 2.1 V; MIC is connected to $V_{MICBIAS}$ through a 2.21 kΩ ±1% resistor.

- Tone mode, $V_{MICBIAS}$ = 2.56 to 2.84 V; MIC is connected to $V_{MICBIAS}$ through a 2.21 kΩ ±1% resistor.

The values in the "Typical" column of the table are measured at 25° C.

**Table 5-5** Electrical characteristics

| Symbol | Parameter | Test conditions | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|---|
| GENERAL | | | | | | |
| $I_{MICBIAS-B}$ | Quiescent Current into MIC+VSHUNT | Button Mode, $V_{MICBIAS}$ = 2.1 V | | 3 | 6 | µA |
| | | Button Mode, $V_{MICBIAS}$ = 1.5 V | | | | |
| $I_{MIC-T}$ | Quiescent Current into MIC | Tone Mode | | 34 | 46 | |
| $I_{VSHUNT-T}$ | Quiescent Current into VSHUNT | Tone Mode (see Note 1, below) | | 60 | 70 | |
| $I_{MIC-TA}$ | Active Current into MIC | Tone Mode | | 35 | 45 | |
| $I_{VSHUNT-TA}$ | Active Current into VSHUNT | Tone Mode (see Note 1, below) | | 104 | 118 | |
| $V_{TR}$ | Tone Mode Threshold Voltage | MIC Rising (Microphone enable), $V_{MICPWR}$ = 1.0 V | 2.20 | 2.35 | 2.50 | V |
| $V_{TF}$ | | MIC Falling (Microphone disable), $V_{MICPWR}$ = 400 mV | 0.55 | 0.8 | 1 | |
| $V_{MICPWR}$ | MICPWR Output Voltage | $I_{MICPWR}$ = 120–150 µA | 1.51 | 1.56 | 1.61 | |
| $R_{SO}$ | Shunt Regulator Output Impedance | Freq = 100 Hz | 5 | 18 | 25 | Ω |
| | | Freq = 20 kHz | 12 | 21 | 35 | |
| $R_{ONA}$ | Switch A, $R_{DSON}$ | Tone Mode, $I_{MICPWR}$ = 1 mA, $V_{MICBIAS}$ = 2.56V | | 40 | 55 | |
| $R_{ONB}$ | Switch B, $R_{DSON}$ | $V_{MIC}$ = 1.2V, $I_{REM}$ = 1 mA | | 22 | 30.5 | |
| TONE MODE | | | | | | |
| $e_{n-mic100}$ | MIC Integrated Noise | 100 Hz to 20 kHz | | 1.5 | 2 | µVrms |

| Symbol | Parameter | Test conditions | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|---|
| $e_{n\text{-}mic1k}$ | MIC Integrated Noise | 1 kHz to 20 kHz | | 0.39 | 1 | |
| $f_{TONE1}$ | Button 1 Frequency | $R_{REM}$ = 6.81 kΩ | 109 | 130 | 159 | kHz |
| $f_{TONE2}$ | Button 2 Frequency | $R_{REM}$ = 9.42 kΩ | 138 | 165 | 200 | |
| $f_{REL}$ | Button Release Frequency | | 81 | 97 | 117 | |
| $R_{BT1}$ | Button 1 Boundary | | 6.61 | 6.81 | 7.01 | kΩ |
| $R_{BT2}$ | Button 2 Boundary | | 9.33 | 9.42 | 9.51 | |
| $V_{TA}$ | Tone Amplitude | $R_{TONE}$ = 1 MΩ | 350 | 550 | 720 | mV |
| | | $R_{TONE}$ = 100 kΩ | 300 | 515 | 710 | |
| BUTTON MODE | | | | | | |
| $t_{ONA}$ | Switch A Enable Time | Time to turn on Switch A | 0.8 | 1.2 | 2 | ms |
| $t_{OFFB}$ | Switch B Disable Time | Time to turn off Switch B | 0.7 | 1 | 2 | |
| $t_{REG}$ | Shunt regulator enable time | Time From MIC = 2.3 V to MICPWR = 1.56 V | 1 | 2.5 | 3.5 | |

**Note 1:** 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.

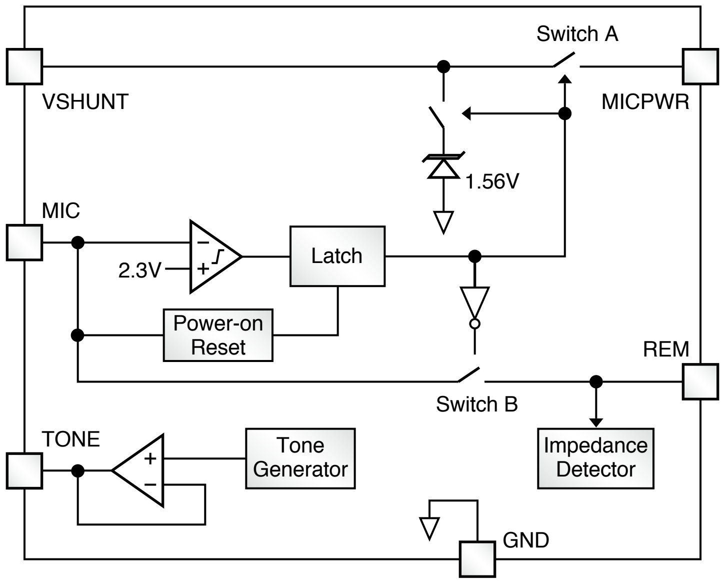
## Theory of Operation

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 5-2 (page 77) 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.

**Figure 5-2** Transmitter 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.

## 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 5-6 (page 77) shows the DETECT pin voltages with $V_{MICBIAS}$ = 2.0 V.

**Table 5-6** DETECT Pin Voltage

| Switch Closed | Voltage |
|---|---|
| $S_0$ | 0.000 V ±1% |
| $S_1$ | 1.510 V ±1% |
| $S_2$ | 1.603 V ±1% |

When the transmitter chip is in button mode ($V_{MIC}$ 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 5-6 (page 77) shows the DC voltage corresponding to a given button press when using the R1 and R4 resistor values listed in Table 5-7 (page 82). This DC level is then detected the microphone bias line by the controller in the Apple device. Switch $S_0$ 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.

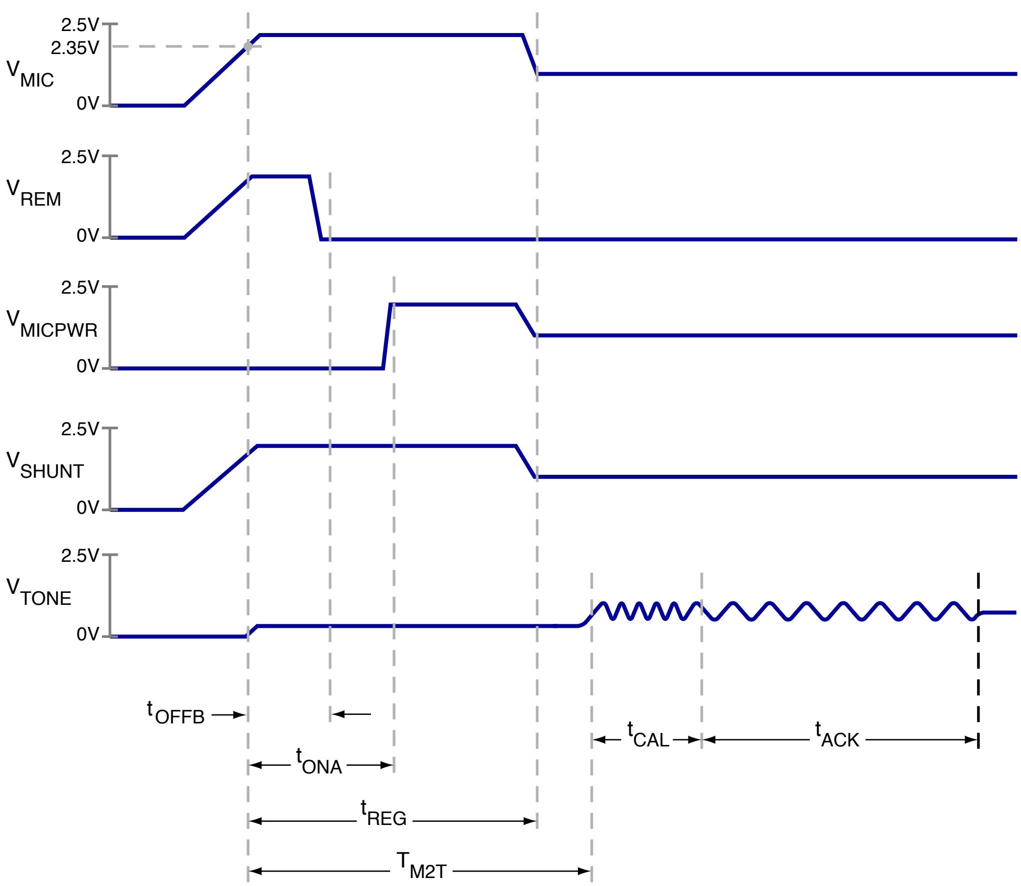
## Tone Mode

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 5-2 (page 77).

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 5-3 (page 78). Values for the timing parameters are given in Table 5-5 (page 75).

**Figure 5-3** Startup timing



The tone mode startup sequence is as follows:

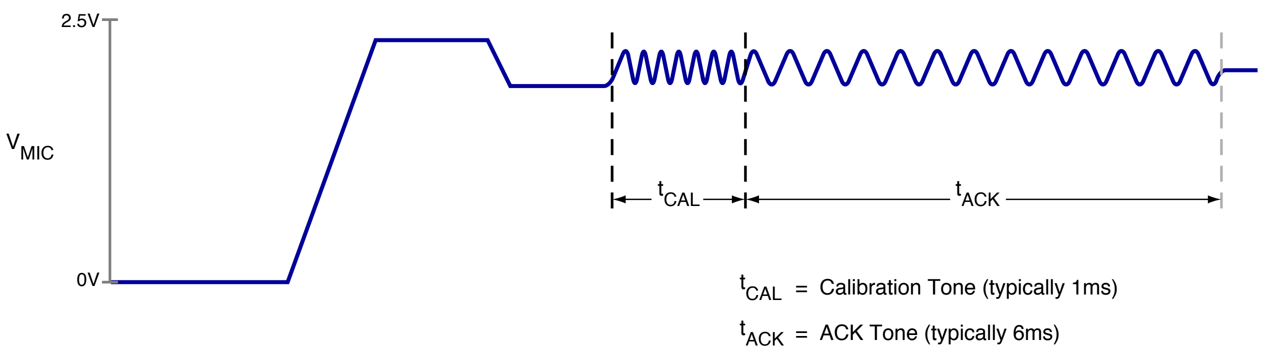
1. Upon detecting $V_{MIC}$ > 2.35 V, the switch connecting the MIC and REM pins together is opened after time $t_{OFFB}$ (see Figure 5-3 (page 78) and Table 5-5 (page 75)).

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 5-4 (page 79)) and authenticates the presence of the transmitter chip.

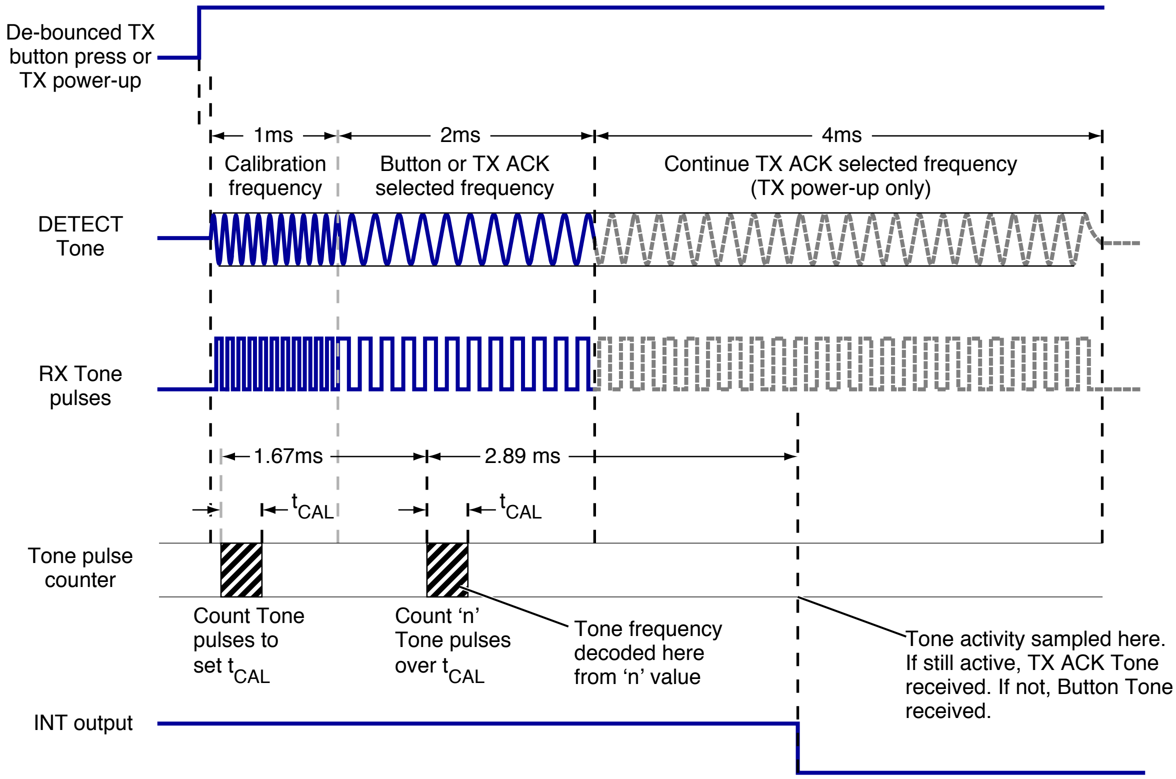
**Figure 5-4** Tone mode ACK sequence




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 5-5 (page 80). 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 5-5** Tone transmit/decode method




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.

# Button Detection Circuitry

To implement remote button detection, the accessory manufacturer must install the following specific components in the Apple device-compatible headphone:

- The Apple-provided transmitter chip described in this specification.
- A Knowles SPQ2409HE5H-PB MEMS digital microphone.

> **Note:** Button detection must not be implemented in accessories without microphones, except for 3G shuffle remote controls.

The circuits in the accessory that support these components must be those shown in Figure 5-6 (page 81) and Figure 5-7 (page 82). The nominal values of the components shown in these schematics are given in Table 5-7 (page 82).

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 in Figure 1-6 must be adjusted to achieve this result. Figure 5-7 (page 82) 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 "Circuit Adjustments" (page 83).

Figure 5-6 (page 81) and Figure 5-7 (page 82) 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 earbud speakers 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 earbud channels and the microphone.

> **Note:** With the exception of the MEMS digital microphone listed in Table 5-7 (page 82), symbol U2, components of equal or better specifications may be substituted for the components called out below.

**Figure 5-6** Transmitter circuit

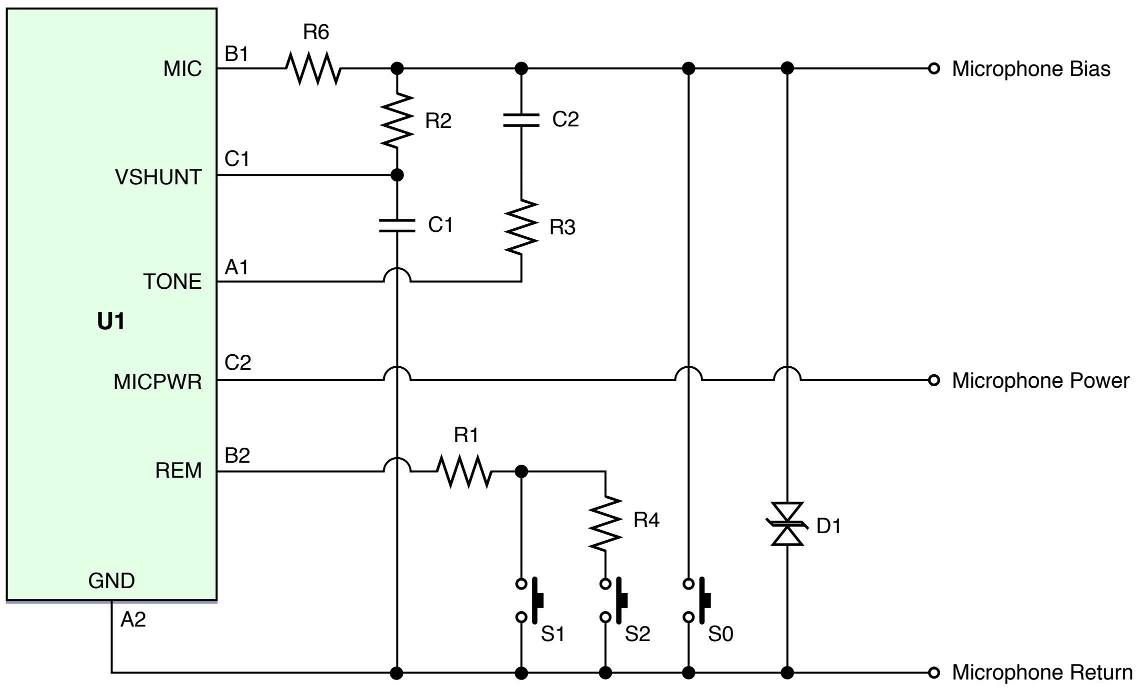


**Figure 5-7** Microphone circuit

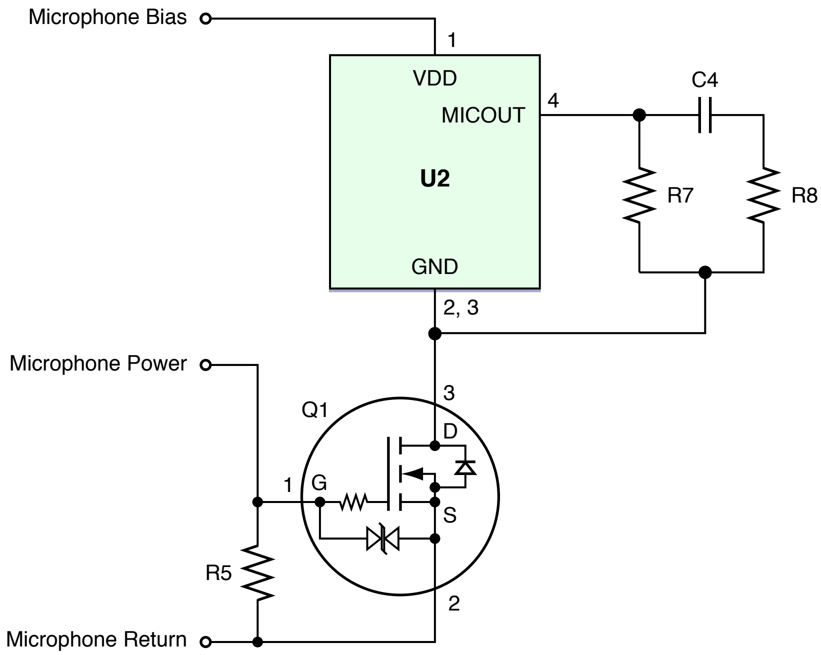



**Table 5-7** Circuit components

| 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 | |

| Symbol | Description | Notes |
|---|---|---|
| R8 | Resistor, 1.2 kΩ ±1%, 1/20 W | |
| S0 | Dome switch | Push-to-talk; must not exceed 20 Ω when closed. |
| S1 | Dome switch | Volume down; must not exceed 20 Ω when closed. |
| S2 | Dome switch | Volume up; must not exceed 20 Ω when closed. |
| U1 | Headset interface transmitter chip | Provided by Apple |
| U2 | Knowles SPQ2409HE5H-PB | MEMS digital microphone (no substitutes) |

## Circuit Adjustments

The values of some of the components listed in Table 5-7 (page 82) 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.

**WARNING:** 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.

# iOS Device Power Supply Requirements

Accessory developers may design external power sources for iOS devices, but such designs **must** follow the guidelines in this appendix to avoid interfering with the touch-sensing operation of the keyboard.

## Converter Switching Frequency Guidelines

To be compatible with the frequency-hopping touch sensors in iOS devices, every AC or DC power adapter design must conform to the following guidelines 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.

## 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 iOS device. It is strongly suggested that any AC adapter design for an iOS 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.

## Impedance Stability of the Diode Bridge

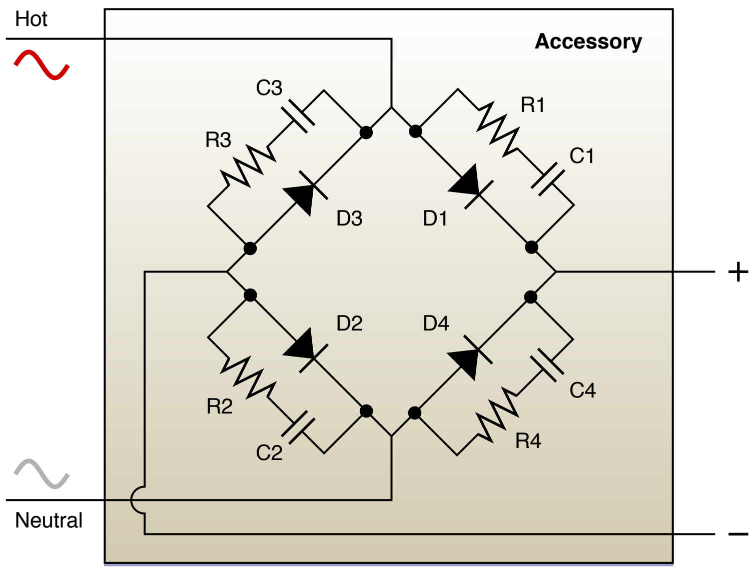
The diodes used in its full-wave bridge rectifier can be a major source of abrupt changes in an AC adapter's series impedance. To reduce unwanted touch sensor output oscillations, the AC adapter circuit should be designed such that its series impedance does not change abruptly.

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 A-1 (page 86). 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, R4 shown were chosen to have trivial levels of impedance relative to the impedances of C1, C2, C3, and C4 at power line frequencies.

**Figure A-1** Typical diode bridge circuit for an AC adapter




| Component | Value |
|---|---|
| C1, C2, C3, C4 | 47 pF typical |
| R1, R2, R3, R4 | 2 kΩ typical |

# Sample Accessory Circuits

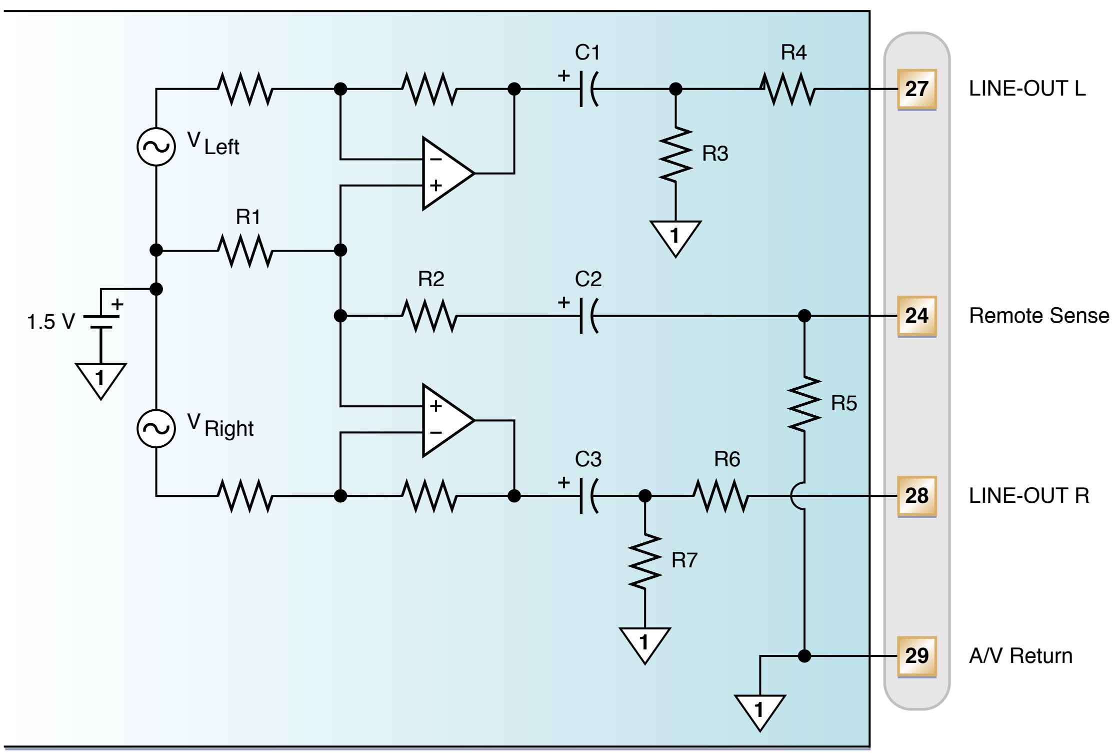
To assist developers with their accessory designs, this appendix includes sample schematics for handling audio and video in three kinds of accessories, as well as partial schematics of the internal audio and video circuitry of Apple devices. The schematics should help developers avoid crosstalk and extraneous noise that may affect the quality of audio and video playing from the devices.

## Internal Audio Circuits in Apple Devices

Figure B-1 (page 87) shows schematically some of the audio circuitry inside a typical Apple device.

**Note:** The audio circuitry shown in Figure B-1 (page 87) inverts the polarity of the signals in both the left and right analog audio channels. This normally has no audible consequences. If an accessory needs to recover the exact analog polarity represented by the digital source, it must invert the LINE-OUT signals.

**Figure B-1** Apple device audio circuitry



| Component | Value |
|---|---|
| C1, C3 | 10 µF |
| C2 | 4.7 µF |
| R1, R2 | 10 kΩ |
| R3, R7 | 100 kΩ |
| R4, R6 | 100 Ω |
| R5 | 47 Ω |
| $\overset{1}{\bigtriangledown}$ | Apple device internal ground |

**Note:** Unless otherwise specified, all component tolerances in this appendix are ±5%.

| 30-pin connector pin | Signal name | I/O | Description |
|---|---|---|---|
| 24 | Remote Sense | I | See "Minimizing Crosstalk and Noise" (page 42). |
| 27 | LINE-OUT L | O | Line level output to the Apple device for the left channel. |
| 28 | LINE-OUT R | O | Line level output to the Apple device for the right channel. |
| 29 | Audio Return | — | Audio return. This is a signal and must not be grounded inside the accessory. |

The Apple device's analog audio circuits, as well as the video subsystems in devices with video output capability, share a common star point as their internal ground reference. This point is shown as $\overset{1}{\bigtriangledown}$ in Figure B-1 (page 87). It is very important to maintain the integrity of the dock connector analog signals with respect to this ground; otherwise, the analog signal handling of the entire device/accessory system may exhibit unwanted behavior. This behavior causes no harm to the Apple device, but it can easily ruin the quality of its audio and video output.

The analog output signal paths are best thought of as loops. The video signals S Video Y (pin 21), S Video C (pin 22), and Composite Video (pin 23), leave the dock connector and must be terminated in a 75.0 Ω ±1% load if they are in use. The load return current flows to Audio Return (pin 29). These input/output connections are listed in Table 1-1 (page 19).

Similarly, analog audio flows from LINE-OUT L (pin 27) and LINE-OUT R (pin 28) to an external load. The load return current flows to Audio Return (pin 29). The external load may be any value in the range 1 kΩ to 100 kΩ.

To make the audio signal as free of video crosstalk as possible requires careful routing of the low side differential sense signal, which is connected to Remote Sense (pin 24). The Remote Sense line should be routed directly from the Apple device's 30-pin connector to the point in the accessory where audio and video share a ground connection (for example, at the ground sleeve of an A/V connector). This trace should be routed to minimize coupling with other signals, because any outside signals coupled into the Remote Sense circuit will appear as extraneous noise in the system's audio outputs. Accessories that do not use the Apple device's video output should terminate Remote Sense to Audio Return at the device's 30-pin connector.

# Verifying an Accessory's Audio/Video Output Design

An accessory's audio/video connections to the Apple device's dock are most easily verified by making measurements with an audio spectrum analyzer, as well as by careful listening and viewing. In an device/accessory system, video-to-audio crosstalk appears in the audio band spectrum as a cluster of noise lines between 25 Hz and 1 kHz and another cluster of lines around 15.6 kHz.

The most bothersome audible artifact is a buzzing sound that may occur when the vertical video raster component appears in the audio. This component will duplicate the field rate (50 or 59.94 Hz) with an amplitude around –106 dBV. The frequency component at the horizontal rate (15625 or 15734 Hz) will be around –90 dBV. When measuring broadband noise density using 20 kHz bandwidth and 512 bins, the broadband density noise will be –122 dBV/bin ±3 dBV.

Careful listening is suggested as an adjunct to careful measurement to verify an device/accessory audio/video design. Careful viewing is helpful to determine whether audio-to-video crosstalk is a problem. A full-scale 1 kHz sine wave audio signal makes a good test; the audio-to-video crosstalk appears as horizontal bars in the video image when the audio is playing.

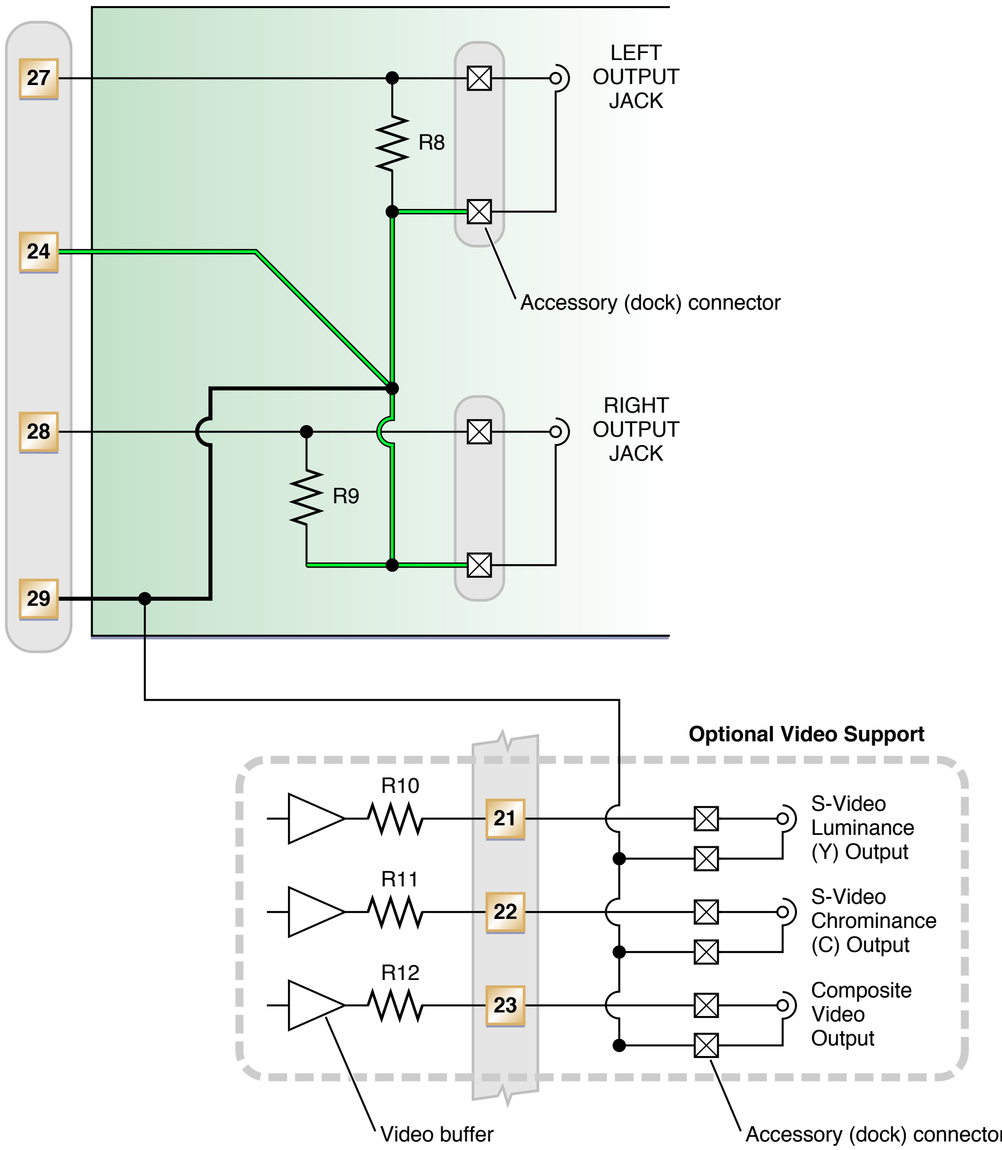
Correct accessory system designs should exhibit insignificant video-to-audio crosstalk and audio-to-video crosstalk when playing Apple device outputs.

**WARNING:** The circuits illustrated in the rest of this appendix are samples, *not* reference designs. They are shown here only for general guidance; using them as-is will not guarantee the successful operation of any specific accessory.

# Sample 1: A Passive Dock Accessory

Figure B-2 (page 90) shows sample circuitry of an accessory that serves as a passive dock for Apple devices.

**Figure B-2** A passive Apple device dock




| Component | Value |
|---|---|
| R8, R9 | 47 kΩ |
| R10, R11, R12 | 75.0 Ω ±1% |

| 30-pin connector pin | Signal name | I/O | Description |
|---|---|---|---|
| 21 | S Video Y | O | The luminance component of S Video. |

| 30-pin connector pin | Signal name | I/O | Description |
|---|---|---|---|
| 22 | S Video C | O | The chrominance component of S Video. |
| 23 | Composite Video | O | Composite Video output. |
| 24 | Remote Sense | I | See "Minimizing Crosstalk and Noise" (page 42). |
| 27 | LINE-OUT L | O | Line level output to the Apple device for the left channel. |
| 28 | LINE-OUT R | O | Line level output to the Apple device for the right channel. |
| 29 | Audio Return | — | Audio return. This is a signal and must not be grounded inside the accessory. |

The following notes apply to the sample design in Figure B-2 (page 90) if the accessory supports audio:

- Connect the left and right connector shields together, using a short trace, to make sure they are at the same potential.
- Connect the midpoint of the foregoing connection to pin 29. This completes the audio return circuit back to the Apple device.
- Connect pin 24 to the mid-point of the first connection.
- Any external noise coupled to pin 24 appears at the output; therefore, make sure pin 24 is not routed near noisy traces.
- Resistors R8 and R9 prevent the left and right audio signals from floating when the Apple device is unplugged.

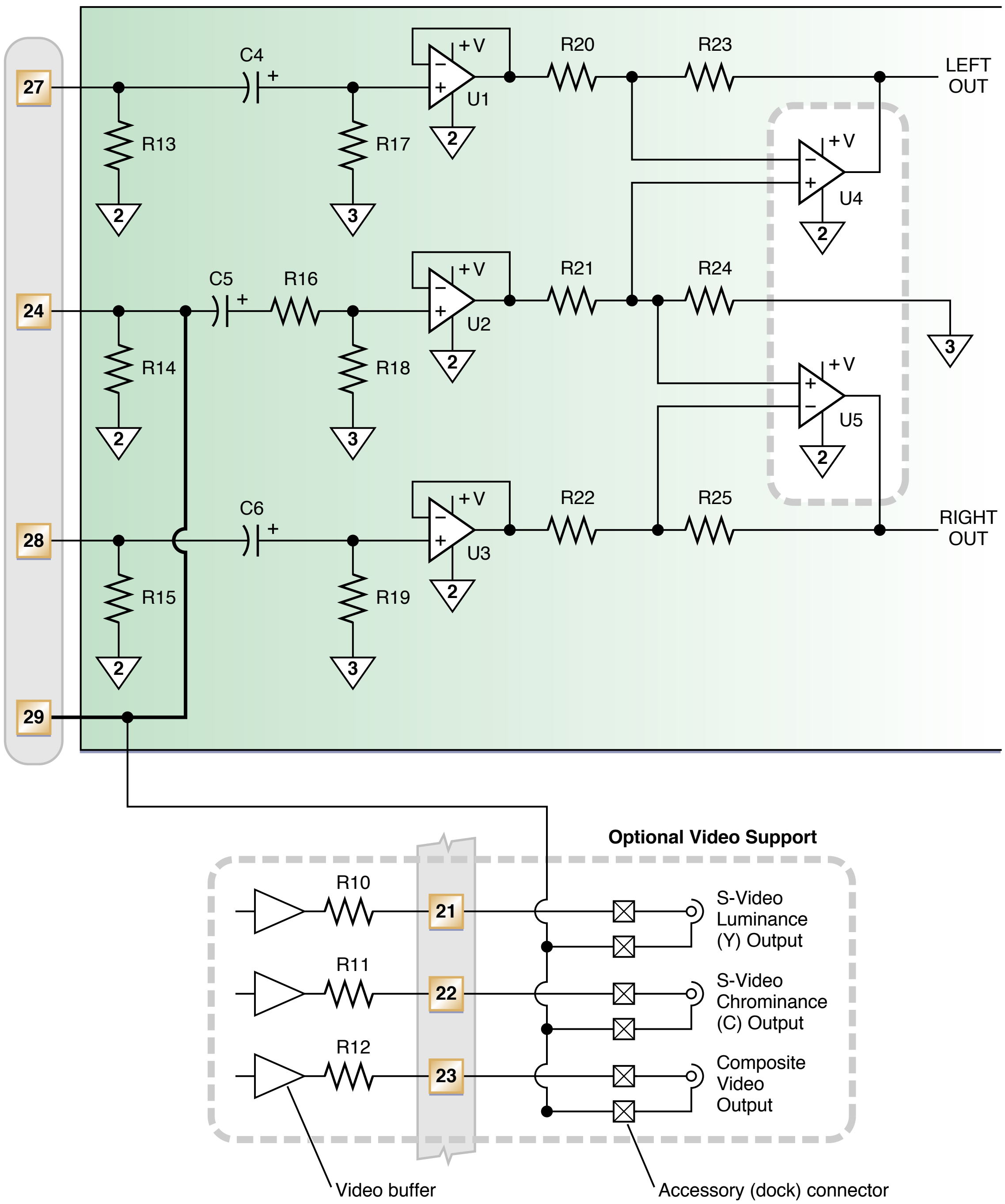
If the accessory also supports video, these additional notes apply:

- Connect the S video Y, S video C, and composite output connector shields together, using a short trace.
- Connect the audio return and the video return together at pin 29.

## Sample 2: An Apple Device-Powered Accessory

Figure B-3 (page 92) shows an example of an accessory that draws its power from an attached Apple device. Circuitry of the type shown is useful when connecting the Apple device's ground to a different ground in a high common-mode noise environment. This example shows an arrangement that uses only one power supply, that in the Apple device.

**Figure B-3** An Apple device-powered accessory




| Component | Value |
|---|---|
| C4, C5, C6 | 10 µF |
| R10, R11, R12 | 75.0 Ω ±1% |

| Component | Value |
|---|---|
| R13, R15, R17, R19 | 100 kΩ |
| R14 | 1 kΩ |
| R16 | 100 Ω |
| R18, R20, R21, R22, R23, R24, R25 | 49.9 kΩ ±1% |
| U1, U2, U3, U4, U5 | Differential amplifiers |
| 2 | Apple device digital ground (pins 1, 2, 15, 16, and 30) |
| 3 | Accessory device reference voltage |

| 30-pin connector pin | Signal name | I/O | Description |
|---|---|---|---|
| 21 | S Video Y | O | The luminance component of S Video. |
| 22 | S Video C | O | The chrominance component of S Video. |
| 23 | Composite Video | O | Composite Video output. |
| 24 | Remote Sense | I | See "Minimizing Crosstalk and Noise" (page 42). |
| 27 | LINE-OUT L | O | Line level output to the Apple device for the left channel. |
| 28 | LINE-OUT R | O | Line level output to the Apple device for the right channel. |
| 29 | Audio Return | — | Audio return. This is a signal and must not be grounded inside the accessory. |

The following notes apply to the sample design in Figure B-3 (page 92) if the accessory supports audio:

- Resistors R20 through R25 set common mode rejection.
- R16 is needed for maximum common mode rejection.
- LEFT OUT and RIGHT OUT lead to the accessory's audio circuits. To minimize popping and clicking sounds that occur at power-up and power-down, add circuitry to the accessory.

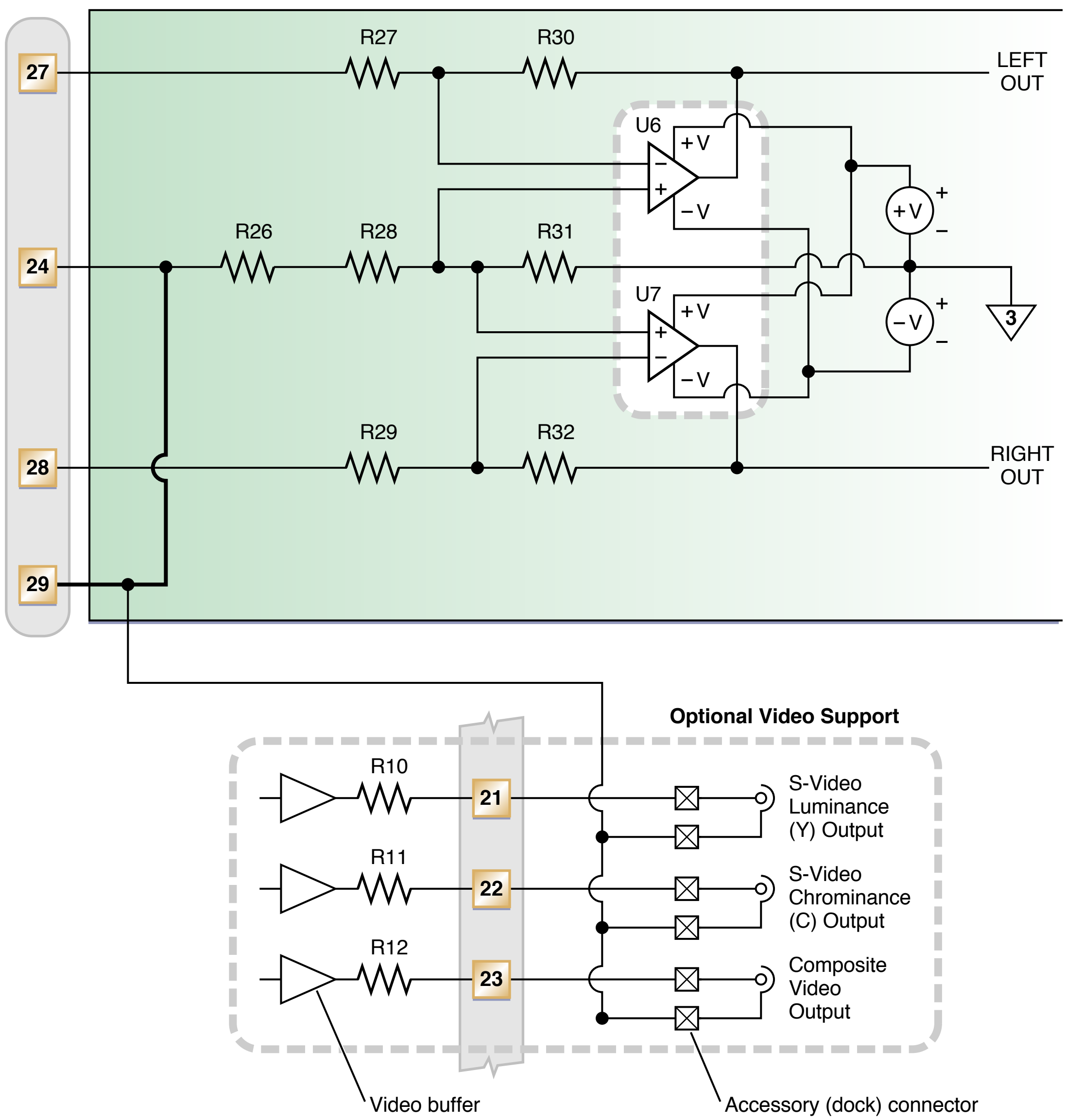
If the accessory also supports video, these additional notes apply:

- Connect the S Video Y, S Video C, and Composite output connector shields together, using a short trace.
- Connect the audio return and the video return together at pin 29.

# Sample 3: A Self-Powered Accessory

Figure B-4 (page 94) shows an example of an accessory that provides its own power. Circuitry of the type shown is useful when connecting the Apple device's ground to a different ground in a high common-mode noise environment.

**Figure B-4** A self-powered accessory




| Component | Value |
|---|---|
| R10, R11, R12 | 75.0 Ω ±1% |

| Component | Value |
|---|---|
| R26 | 100 Ω |
| R27, R28, R29, R30, R31, R32 | 22.1 kΩ ±1% |
| U6, U7 | Differential amplifiers |
| 3 | Accessory device reference voltage |

| 30-pin connector pin | Signal name | I/O | Description |
|---|---|---|---|
| 21 | S Video Y | O | The luminance component of S Video. |
| 22 | S Video C | O | The chrominance component of S Video. |
| 23 | Composite Video | O | Composite Video output. |
| 24 | Remote Sense | I | See "Minimizing Crosstalk and Noise" (page 42). |
| 27 | LINE-OUT L | O | Line level output to the Apple device for the left channel. |
| 28 | LINE-OUT R | O | Line level output to the Apple device for the right channel. |
| 29 | Audio Return | — | Audio return. This is a signal and must not be grounded inside the accessory. |

The following notes apply to the sample design in Figure B-4 (page 94) if the accessory supports audio:

- Common mode rejection is set by resistors R30 through R35, which must have tolerances of ±1% or better.
- R29 is needed for maximum common mode rejection.
- Using a split power supply for U6 and U7, as shown in the diagram, is the easiest way to avoid poor common mode rejection at low frequencies.
- The accessory reference circuit must have a low source impedance.
- The connection to pin 24 must be direct and must not run near external noise sources.
- LEFT OUT and RIGHT OUT lead to the accessory's audio circuits. To minimize popping and clicking sounds that occur at power-up and power-down, add circuitry to the accessory.

If the accessory also supports video, these additional notes apply:

- Connect the S Video Y, S Video C, and Composite output connector shields together, using a short trace.
- Connect the audio return and the video return together at pin 29.

## Avoiding a Warning When a Self-Powered Accessory Is Off

The two circuits shown in Figure B-5 (page 96) can be used to let a self-powered iPhone accessory selectively drive Accessory Identify (pin 10) and Accessory Detect (pin 20) of the iPhone's 30-pin connector. The accessory must not ground pin 20 or present an ID resistor on pin 10 until it is in a state in which it can generate and handle iAP traffic.

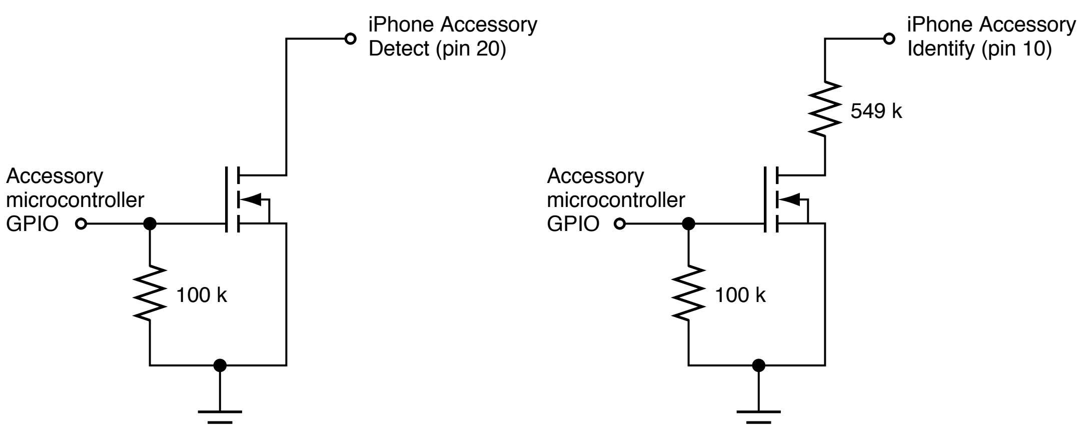
If a self-powered accessory does not use these or similar circuits, the iPhone will display an incompatibility message whenever it is connected and the accessory is unpowered.

> **Note:** After powering the accessory, observe the 3-second time limit for completing the iPhone's Identify Device Preferences and Settings process. See "Using IDPS" in *MFi Accessory Firmware Specification*.

Connect the gates of both FETs shown in Figure B-5 (page 96) to a GPIO pin of the accessory microcontroller. Ensure that the FETs transition from fully off to fully on (or vice versa) in no more than 10 ms. Do not connect the FET gates directly to accessory Vcc unless this timing requirement can be met at all times, including immediately after external power to the accessory is switched off or otherwise removed.

> **Note:** These circuits are not required for iPhone-powered accessories, or for self-powered accessories that do not claim iPhone capability.

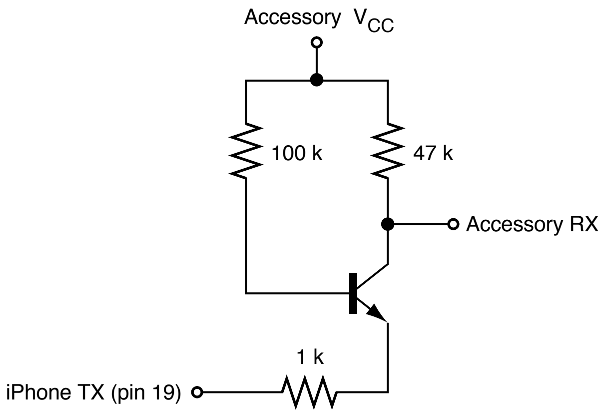
**Figure B-5** Selectively controlling accessory detect and identify




# Avoiding TX Back-Powering

The circuit shown in Figure B-6 (page 97) can be used to prevent logic-high states on the Apple device's serial TX line from inadvertently back-powering the accessory. The resistor values in this example are for illustration only, and the resulting waveform at the accessory's serial RX input in normal operation should be verified with an oscilloscope.

**Figure B-6** TX line isolation circuit



# FireWire to USB Reference Design

This appendix provides a reference design for FireWire to USB DC-to-DC power converters to be used with Apple devices. Charging an Apple device through the FireWire pins on the 30-pin connector is deprecated; instead, accessory designers must provide 5 VDC power to the USB pins.

> **Note:** FireWire to USB DC-to-DC power converters for use with Apple devices must be tested and certified. The required process is specified in "About Accessory Certification" in *MFi Accessory Testing Specification*.

If an accessory has only FireWire power available (or its equivalent), a DC-to-DC converter and resistor divider network must be added to furnish USB power and conform to the D+/D– configuration defined in "USB 2.0" (page 29).

> **Note:** The design presented in this appendix is for general reference only. Actual converters must be designed to meet the specific needs of the accessory.

## Converter Requirements

The FireWire power input to the converter may vary from 8 to 30 VDC. The USB power output from the converter should furnish 5 VDC ±5% at current draws up to 1 A. Other requirements include:

- If the converter is built into the cable to the Apple device's 30-pin connector, it must pass through all the other connections, including the proper handling of pins 10 and 20 as described in "Accessory Detect and Identify" (page 36).
- To conserve power, the converter should turn itself off when no Apple device is connected.
- If USB power is present, the converter should turn itself off and pass all USB connections through. It must pass USB signals without endangering their compliance with requirements such as eye diagram tests.
- If the converter is working to convert FireWire power to USB power, it must furnish the proper resistor divider network, as described in "USB 2.0" (page 29).

## Switching Frequency Issues

The RF and touchscreen operations of iOS devices are sensitive to noise in several frequency bands. (See "iOS Device Power Supply Requirements" (page 85) for details.) Some accessories, such as automotive accessories, also tend to be sensitive to noise in areas such as the AM and FM radio bands. Hence DC-to-DC converter designs must take care to avoid switching frequencies that fall into these bands (or whose harmonics fall into these bands). The frequencies that most often cause trouble are listed in Table F-1.

**Table C-1** Unwanted switching frequencies

| Band | Frequencies |
|---|---|
| AM radio | 520–1710 kHz |
| FM radio | 76–108 MHz |
| GSM | 824–849, 869–960, 1710–1785, 1805–1990 MHz |

Unwanted frequencies can be avoided by selecting a suitable inductor in the USB power out circuit (such as L1 in Figure C-1 (page 101)), and by using a bleeder resistor such as R5 in Figure F-1. (A bleeder resistor can be used only where its power dissipation does not cause efficiency or thermal problems.)

The converter design must also minimize radiated emissions which may cause desense of RF circuits in iOS devices. This can be accomplished through careful PCB layout in combination with proper circuit design. Here are some tips:

- Place a solid ground plane directly beneath the converter circuit.
- Keep the switching node of the converter as close to the converter IC as possible.
- Use snubber branches (such as R2 and C3 in Figure C-1 (page 101)) to minimize ringing on the switch node. Snubbers can also help with efficiency.

## Other Design Issues

The design of a FireWire-to-USB converter should take into account these other issues:

- Efficiency is important in converter designs, and is often a trade-off with switching frequency control. Modern DC-to-DC buck regulators often have efficiencies in the mid-90 percent range.
- Small converter enclosures need to be tested adequately to ensure that they can dissipate the heat created by the converter.
- Hot-plugging the converter to a live supply via highly inductive cabling can produce significant voltage overshoot, which may damage the converter. In addition to bulk capacitance on the BUCK_PWR_IN line, such as C1 in Figure C-1 (page 101), placing a high energy rated zener diode (such as Vishay SMF33A) between BUCK_PWR_IN and ground can help protect the converter from dangerous voltage peaks.

## Typical Design

A typical design for a FireWire to USB converter uses a buck DC-to-DC converter to step the FireWire voltage down to the USB voltage with minimum power loss. Figure C-1 (page 101) shows a Linear Technology LT3493 1.2 A 750 kHz switching regulator in this role. In Figure F-1, R2 and C3 form a snubber to suppress voltage transients.

If USB power is present from another source, it is switched in by the circuit shown in Figure C-2 (page 101). This circuit also supplies a 2 V turn-on signal (USB_PWR_SIG) to the switch shown in Figure C-4 (page 102).

If either the Apple device is disconnected or USB power is present, the disabling circuit shown in Figure C-3 (page 102) shuts down the converter. The resistor divider network shown in Figure C-4 (page 102) delivers power to the Apple device.

Values of the components show in the Figures F-1 through F-4 are given in Table C-2 (page 102).

**Figure C-1** Main converter circuit

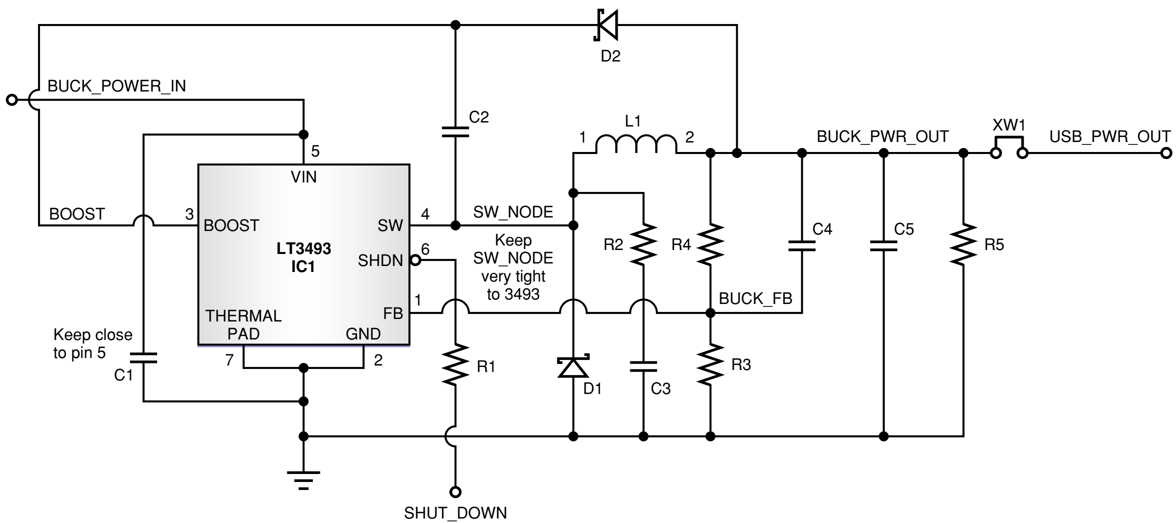



**Figure C-2** USB power switching circuit

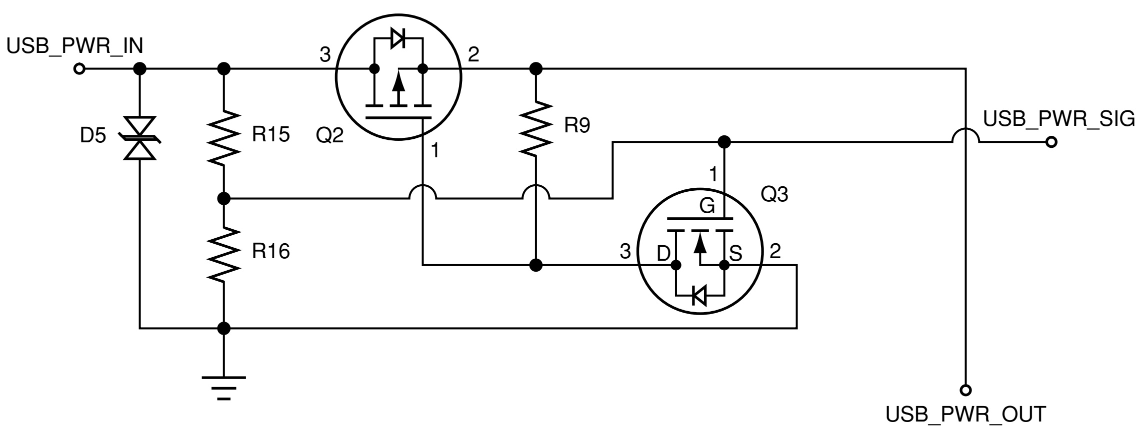


**Figure C-3** Converter shut-down circuit

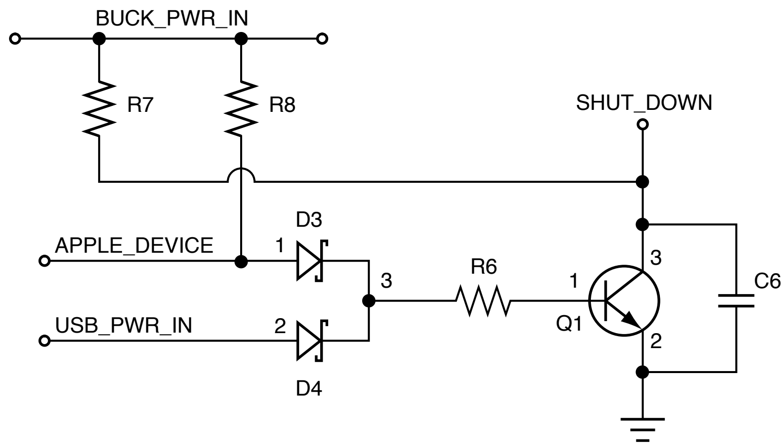



**Figure C-4** Resistor divider network

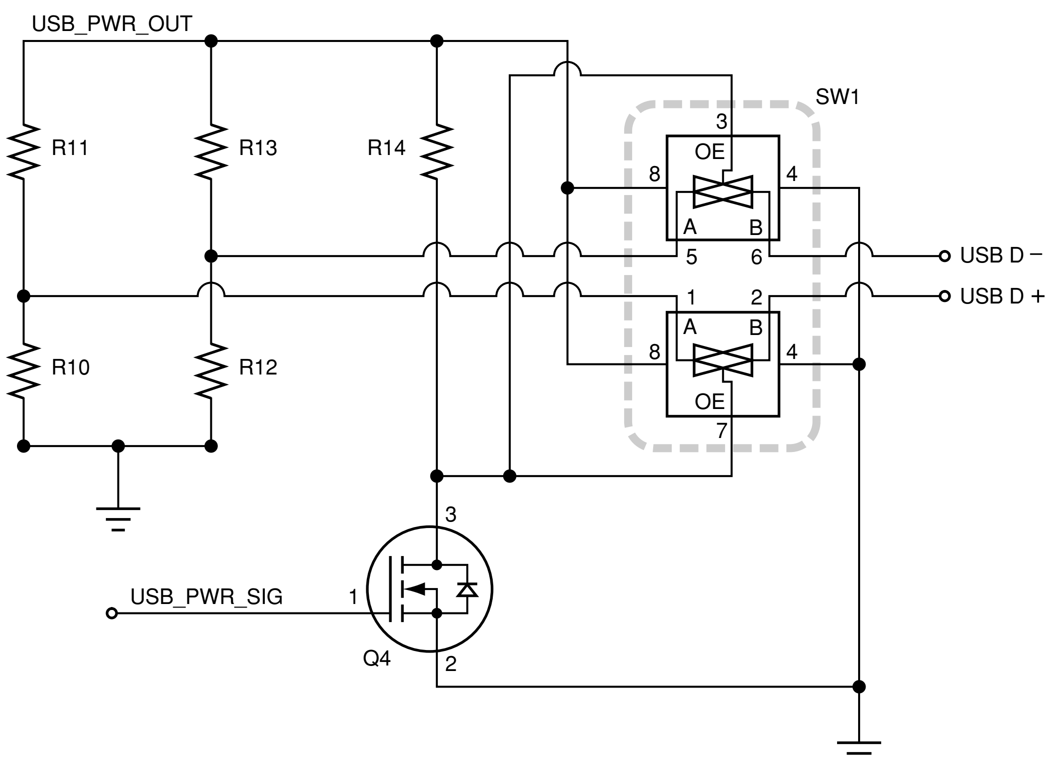



**Table C-2** Typical converter circuit components

| Symbol | Description | Notes |
|---|---|---|
| C1 | Capacitor, 2.2 µF ±10%, 50 V | |
| C2 | Capacitor, 0.1 µF ±10%, 25 V | |

| Symbol | Description | Notes |
|---|---|---|
| C3 | Capacitor, design-specific value | Snubber capacitor |
| C4 | Capacitor, 22 pF ±5%, 50 V | Ceramic |
| C5 | Capacitor, 10 μF ±20%, 6.3 V | |
| C6 | Capacitor, 0.1 μF ±10%, 25 V | |
| D1 | Schottky diode | Rohm RB160VA-40 |
| D2 | Schottky diode, 40 V clamp | ON Semiconductor NSR0140 |
| D3 | Schottky diode | Fairchild BAT54 |
| D4 | Schottky diode | Fairchild BAT54 |
| D5 | 8 V, 100 pF varistor | ESD transient suppressor |
| IC1 | Step-down switching regulator, 1.2 A, 750 kHz | Linear Technology LT3493 |
| L1 | Power inductor, 68 μH, 1.6 A | TDK VLF10045T-680M1R6 |
| Q1 | Transistor amplifier, NPN | Fairchild MMBT3904 |
| Q2 | P-channel MOSFET, 40 V | Vishay Siliconix Si2319DS |
| Q3 | N-channel MOSFET, 40 V | Vishay Siliconix Si2318DS |
| Q4 | N-channel MOSFET, 40 V | Vishay Siliconix Si2318DS |
| R1 | Resistor, 15 kΩ ±5%, 1/16 W | |
| R2 | Resistor, design-specific value | Snubber resistor |
| R3 | Resistor, 10.0 kΩ ±1%, 1/16 W | |
| R4 | Resistor, 54.9 kΩ ±1%, 1/16 W | |
| R5 | Resistor, design-specific value | Bleeder resistor |
| R6 | Resistor, 3 kΩ ±5%, 1/16 W | |
| R7 | Resistor, 2.9 kΩ ±5%, 1/16 W | |
| R8 | Resistor, 4 kΩ ±5%, 1/16 W | |
| R9 | Resistor, 1 MΩ ±5%, 1/16 W | |
| R10 | Resistor, 49.9 kΩ ±1%, 1/16 W | |
| R11 | Resistor, 75 kΩ ±1%, 1/16 W | |

| Symbol | Description | Notes |
|---|---|---|
| R12 | Resistor, 49.9 kΩ ±1%, 1/16 W | |
| R13 | Resistor, 43.2 kΩ ±1%, 1/16 W | |
| R14 | Resistor, 1 MΩ ±5%, 1/16 W | |
| R15 | Resistor, 1.5 MΩ ±5%, 1/16 W | |
| R16 | Resistor, 1 MΩ ±5%, 1/16 W | |
| SW1 | Dual SPST switch | Fairchild NC7WB66 |
| XW1 | Short | Soldermask-covered |

# Interfacing With the 3G iPod

The 3G iPod is the oldest model of Apple device with a 30-pin connector. Because later models also have 30-pin connectors, users may assume that an accessory designed for later Apple devices will work in the same way when a 3G iPod is plugged into it. However, there are functional differences. This appendix summarizes some of the model-specific design issues that must be addressed if an accessory for current Apple device models is to provide 3G iPod support.

## Accessory Detection

Serial accessories designed to work with the 3G iPod must use a 549 kΩ $R_{ID}$ resistor for accessory detection (see "Accessory Detect and Identify" (page 36)). With any other resistor value, the 3G iPod will not correctly detect when the accessory has been detached. If the accessory uses Extended Interface Mode, this can cause the 3G iPod to remain locked in Extended mode until a new serial accessory is attached to it.

## Powering the 3G iPod

Note the following cautions about accessories that charge or provide power to the 3G iPod:

- The 3G iPod may be charge only through FireWire power; the presence of the D+ and D– resistors required for USB power can disable it.
- An accessory that can successfully charge or power a 3G iPod will not pass self-certification testing.

## Connector Usage

The 3G iPod serial port is shared between the 30-pin connector and the 9-pin Audio/Remote connector. Attaching a serial accessory to the 30-pin connector of a 3G iPod makes any accessories attached to the 9-pin Audio/Remote connector inactive, because the 3G iPod shares the serial port between these two connectors. The iPod mini, and 4G iPod models have two serial ports, so plugging in a 30-pin connector port serial accessory does not deactivate the 9-pin Audio/Remote connector.

Connecting both FireWire power and USB power simultaneously will put the 3G iPod into an invalid state.

The 3G iPod's serial communication rate is limited to 19200 baud, and it does not support autobaud on framing errors (see "UART" (page 51)).

The 3G iPod does not automatically awake from Sleep state when it receives a packet over UART transport (see "Apple Device Power States" (page 43)).

# DisplayPort Digital Video Connectivity

Future Apple products may support two-lane DisplayPort output connectivity to digital video accessories through their 30-pin connectors. The connection to an external video accessory is typically done through an accessory dock or cable, which must perform the IDPS and Authentication 2.0 processes specified in *MFi Accessory Firmware Specification*.

## Connection to the Apple Device

Some Apple devices may use seven signal pins and four ground pins of their 30-pin connector to provide DisplayPort connectivity to digital video accessories. They repurpose the signal pins in all operating modes from their previous usage, as shown in Table E-1. For other signals on all Apple device 30-pin connectors, see Table 1-1 (page 19).

**Table E-1** Repurposing of 30-pin connector pins for DisplayPort

| Pin | Proposed new usage | | Old usage | |
|---|---|---|---|---|
| | **Signal name** | **Function** | **Signal name** | **Function** |
| 1 | DGND | Digital ground in the Apple device and in the accessory | Same | |
| 2 | GND | Main link lane 0 return | DGND | Digital ground |
| 3 | DP_ML_LANE0_P | Main link lane 0 positive | TPA+ | Deprecated FireWire signal |
| 4 | USB D+ | USB signal | Same | |
| 5 | DP_ML_LANE0_N | Main link lane 0 negative | TPA– | Deprecated FireWire signal |
| 6 | USB D- | USB signal | Same | |
| 7 | DP_ML_LANE1_P | Main link lane 1 positive | TPB+ | Deprecated FireWire signal |
| 8 | USB Vbus | USB power in. | Same | |
| 9 | DP_ML_LANE1_N | Main link lane 1 negative | TPB– | Deprecated FireWire signal |
| 10 | Accessory Identify | See "Accessory Detect and Identify" (page 36) for details. | Same | |

| Pin | Proposed new usage | | Old usage | |
|---|---|---|---|---|
| | Signal name | Function | Signal name | Function |
| 11 | DP_HPD | Hot plug detect | F/W PWR+ | Deprecated FireWire signal |
| 12 | F/W PWR+ | FireWire and charger input power (8 V to 15 V DC) **Deprecated** | Same | |
| 13 | Accessory Power | 3.3 V is the nominal output from the Apple device. Nominal current is 5 mA or less (low power mode), with current limited to 100 mA in high power mode. | Same | |
| 14 | Reserved | Pin must be left disconnected (floating). | Same | |
| 15 | GND | Auxiliary channel return | DGND | Digital ground |
| 16 | GND | Main link lane 1 return | DGND | Digital ground |
| 17 | Reserved | Pin must be left disconnected (floating). | Same | |
| 18 | Apple Device RX | Apple device accessory protocol (Data to the Apple device from the accessory) | Same | |
| 19 | Apple Device TX | Apple device accessory protocol (Data from the Apple device to the accessory) | Same | |
| 20 | Accessory Detect | See "Accessory Detect and Identify" (page 36) for details. | Same | |
| 21 | S Video Y / Component Video Pr | Either the luminance signal of S Video or the (Pr) signal of component video. | Same | |
| 22 | S Video C / Component Video Y | Either the chrominance signal of S Video or the luma signal of component video. | Same | |
| 23 | Composite Video / Component Video Pb | Either the Composite Video signal or the (Pb) signal of Component Video. | Same | |
| 24 | Remote Sense | See "Minimizing Crosstalk and Noise" (page 42). | Same | |
| 25 | DP_AUX_CH_P | Auxiliary channel positive | LINE-IN L | Line level input, left |
| 26 | DP_AUX_CH_N | Auxiliary channel negative | LINE-IN R | Line level input, right |
| 27 | LINE-OUT L | Line level output to the Apple device for the left channel. | Same | |

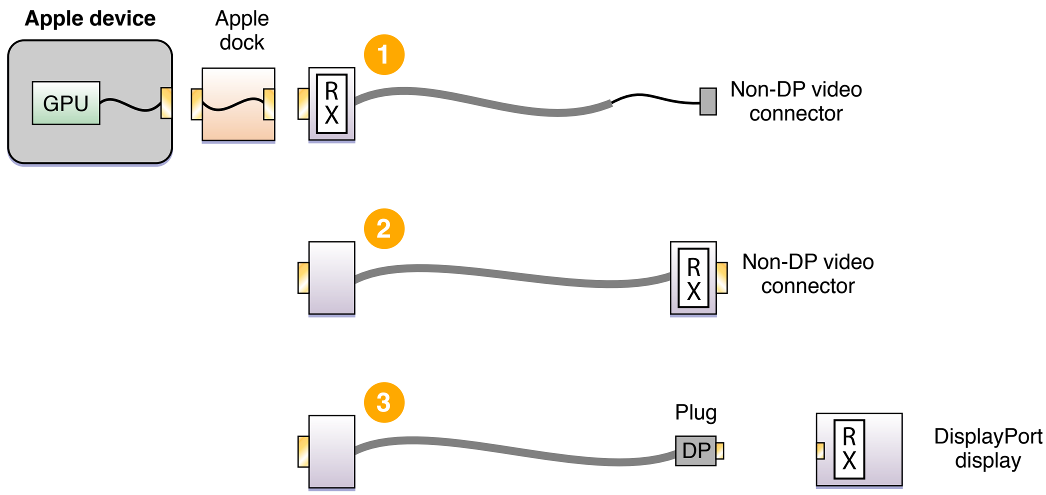
| Pin | Proposed new usage | | Old usage | |
|---|---|---|---|---|
| | **Signal name** | **Function** | **Signal name** | **Function** |
| 28 | LINE-OUT R | Line level output to the Apple device for the right channel. | Same | |
| 29 | A/V Return | Audio/Video return. This is a signal and must not be grounded inside the accessory. | Same | |
| 30 | Apple Device Detect | Ground signal for Apple device detection. The accessory must ground this pin if it is not monitoring it to detect Apple device attachment. | Same | |
| | Chassis | Mounting tabs: chassis ground for connector shell. | Same | |

# DisplayPort Output

The Apple device's DisplayPort output generally conforms to the VESA *DisplayPort Specification,* Version 1.2, with the following differences and added specifications:

- The supported DisplayPort connection configurations are those shown in Figure E-1 (page 110).
- The main link comprises two signaling lanes, as shown in Table E-1 (page 107).
- Only Manchester Mode transactions are supported on the AUX channel.
- The VESA HBR2 signaling rate and FAUX signaling specifications are not supported.
- The VESA *DisplayPort Interoperability Specification* (for connecting to DVI and HDMI displays using level-shifting adaptors) is not supported.
- The Apple device provides power on the Accessory Power line of the 30-pin connector; it does not support the VESA DisplayPort power standard.

**Figure E-1** Apple device-to-accessory DisplayPort configurations




> **Note:** The Apple dock shown in Figure E-1 (page 110) is optional; it may be included or not in any of the three cable configurations.

Figure E-1 (page 110) shows the following devices and cables that may be used to connect an Apple device to a digital video accessory:

- The Apple dock passes the DisplayPort signals from a 30-pin plug that accepts an Apple device to a 30-pin receptacle.
- A DisplayPort receiver implemented within a 30-pin connector that plugs into the Apple dock is labeled (1) in Figure E-1 (page 110). This receiver typically converts the DisplayPort signaling standard and protocol to a different audio/video signaling standard and protocol.
- A DisplayPort receiver implemented at the far end of a cable assembly is labeled (2) in Figure E-1 (page 110). This receiver was to convert the DisplayPort signaling standard and protocol to a different audio/video signaling standard and protocol. The receiver was to compensate for signal losses in the cable.
- A cable assembly that allows connection to a DisplayPort receiver is labeled (3) in Figure E-1 (page 110). This cable was to be be implemented as a passive connection with a worst case loss and with no active components. Alternatively, it could have contained an active repeater.

# Cable Connections

Two tables in this section specify the wiring of the cables shown in Figure E-1 (page 110).

Table E-2 specifies the wiring of the cable labeled (3) when the far end has a DisplayPort connector. Table E-3 (page 111) specifies the wiring of the cable labeled (3) when the far end has a Mini DisplayPort connector.

**Table E-2** Apple device to accessory DisplayPort cable

| 30-pin connector end | | | | Cable | | | | Accessory end | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Signal** | **Name** | **Pin** | | **Pin** | | **Pin** | | **Pin** | **Name** | **Signal** |
| GND | GND | 2 | | 2 | <—> | 11 | | 11 | GND | GND |
| Out | ML_Lane 0 (p) | 3 | | 3 | <—> | 12 | | 12 | ML_Lane 0 (p) | In |
| Out | ML_Lane 0 (n) | 5 | | 5 | <—> | 10 | | 10 | ML_Lane 0 (n) | In |
| GND | GND | 16 | | 16 | <—> | 8 | | 8 | GND | GND |
| Out | ML_Lane 1 (p) | 7 | | 7 | <—> | 9 | | 9 | ML_Lane 1 (p) | In |
| Out | ML_Lane 1 (n) | 9 | | 9 | <—> | 7 | | 7 | ML_Lane 1 (n) | In |
| | | | | | N/C | 5 | | 5 | GND | GND |
| | | | | | N/C | 6 | | 6 | ML_Lane 2 (p) | In |
| | | | | | N/C | 4 | | 4 | ML_Lane 2 (n) | In |
| | | | | | N/C | 19 | | 19 | GND | GND |
| In | Hot Plug Detect | 11 | | 11 | <—> | 18 | | 18 | Hot Plug Detect | Out |
| | | | | | N/C | 13 | | 13 | CONFIG1 | CFG |
| | | | | | N/C | 14 | | 14 | CONFIG2 | CFG |
| | | | | | N/C | 2 | | 2 | GND | GND |
| | | | | | N/C | 3 | | 3 | ML_Lane 3 (p) | In |
| | | | | | N/C | 1 | | 1 | ML_Lane 3 (n) | In |
| GND | GND | 15 | | 15 | <—> | 16 | | 16 | GND | GND |
| I/O | AUX_CH (p) | 25 | | 25 | <—> | 15 | | 15 | AUX_CH (p) | I/O |
| I/O | AUX_CH (n) | 26 | | 26 | <—> | 17 | | 17 | AUX_CH (n) | I/O |
| | | | | | N/C | 20 | | 20 | DP_PWR | PWR Out |

**Table E-3** Apple device to accessory Mini DisplayPort cable

| 30-pin connector end | | | | Cable | | | | Accessory end | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Signal** | **Name** | **Pin** | | **Pin** | | **Pin** | | **Pin** | **Name** | **Signal** |
| GND | GND | 2 | | 2 | <—> | 8 | | 8 | GND | GND |
| Out | ML_Lane 0 (p) | 3 | | 3 | <—> | 12 | | 12 | ML_Lane 0 (p) | In |

| 30-pin connector end | | | | Cable | | | | Accessory end | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Signal** | **Name** | **Pin** | | **Pin** | | **Pin** | | **Pin** | **Name** | **Signal** |
| Out | ML_Lane 0 (n) | 5 | | 5 | <—> | 10 | | 10 | ML_Lane 0 (n) | In |
| GND | GND | 16 | | 16 | <—> | 13 | | 13 | GND | GND |
| Out | ML_Lane 1 (p) | 7 | | 7 | <—> | 17 | | 17 | ML_Lane 1 (p) | In |
| Out | ML_Lane 1 (n) | 9 | | 9 | <—> | 15 | | 15 | ML_Lane 1 (n) | In |
| | | | | | N/C | 7 | | 7 | GND | GND |
| | | | | | N/C | 11 | | 11 | ML_Lane 2 (p) | In |
| | | | | | N/C | 9 | | 9 | ML_Lane 2 (n) | In |
| | | | | | N/C | 19 | | 19 | GND | GND |
| In | Hot Plug Detect | 11 | | 11 | <—> | 2 | | 2 | Hot Plug Detect | Out |
| | | | | | N/C | 4 | | 4 | CONFIG1 | CFG |
| | | | | | N/C | 6 | | 6 | CONFIG2 | CFG |
| | | | | | N/C | 1 | | 1 | GND | GND |
| | | | | | N/C | 5 | | 5 | ML_Lane 3 (p) | In |
| | | | | | N/C | 3 | | 3 | ML_Lane 3 (n) | In |
| GND | GND | 15 | | 15 | <—> | 14 | | 14 | GND | GND |
| I/O | AUX_CH (p) | 25 | | 25 | <—> | 16 | | 16 | AUX_CH (p) | I/O |
| I/O | AUX_CH (n) | 26 | | 26 | <—> | 18 | | 18 | AUX_CH (n) | I/O |
| | | | | | N/C | 20 | | 20 | DP_PWR | PWR Out |

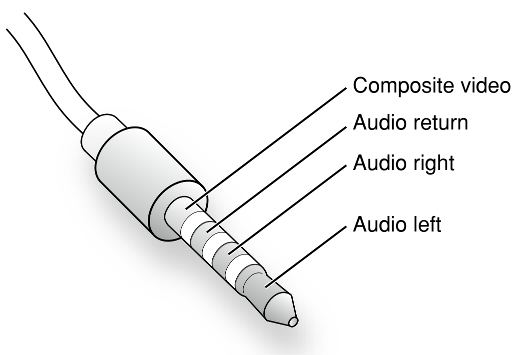
# Historical Information

This appendix memorializes the hardware specifications of past Apple devices. It is included in this specification to provide guidance for developers who need to design accessories compatible with these past technologies.

## Headphone Jack on the 4G iPod color and 5G iPod

**For legacy purposes only,** the 4G iPod (color display) and 5G iPod can provide composite video signals through their headphone jack, as shown in Figure F-1 (page 113). These signals have the same electrical characteristics as the composite video signals available from the 30-pin connector; see Table 2-11 (page 39). **This use of the composite video signal on the headphone jack is deprecated.**

**Figure F-1** Pinout for the headphone jack on the 4G iPod color display and 5G iPod




## 38400/57600 BPS Serial Baud Rates

Table F-1 (page 113) shows the Apple device firmware versions in which 38400 and 57600 bps serial baud rates were introduced. The rate of 38400 bps has since been deprecated. The current baud rates for all Apple devices are described in "UART" (page 51).

**Table F-1** Firmware versions introducing 38400/57600 bps serial baud rates

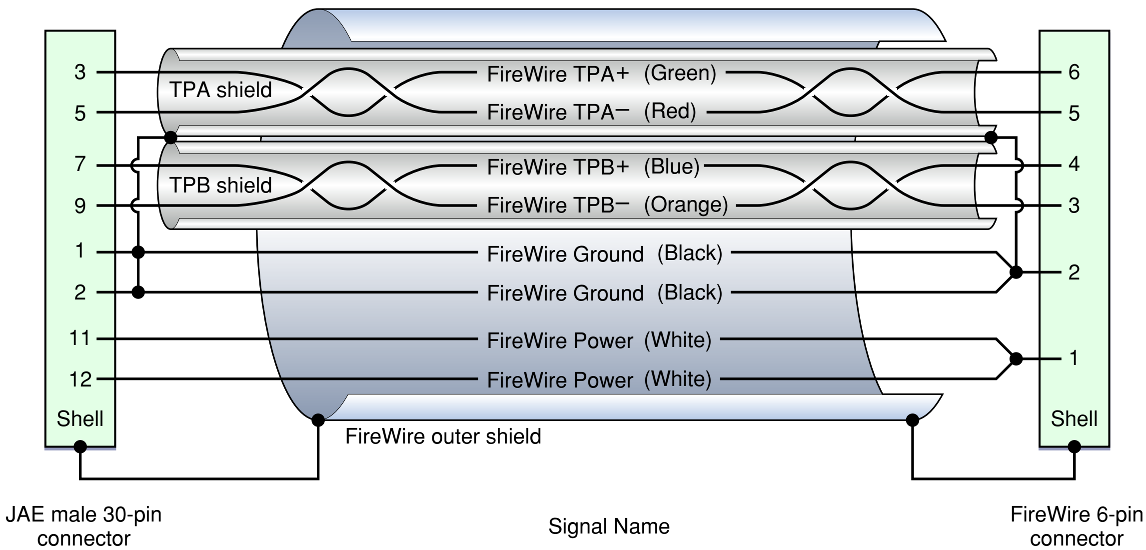
| Features | Software versions | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 3G | mini | 4G | nano | 5G | 2G nano | classic | 3G nano | touch/ iPhone |
| Serial baud rates: 38400/57600 bps | — | 1.2.0 | 3.0.2 | 1.0.0 | 1.0.0 | 1.0.0 | 1.0.0 | 1.0.0 | 1.0 |

# FireWire

**Note: The 30-pin connector FireWire interface is deprecated.**

**For legacy purposes,** the 30-pin connector FireWire interface is designed to the IEEE standard 1394a, supporting transfer rates up to 400 Mbps. Per the IEEE 1394a specification, digital twisted pairs of wires need to be reversed as shown in Figure F-2 (page 114).

**Figure F-2** 30-pin to FireWire cable




The FireWire power pins on the 30-pin connector support 8-volt to 15-volt DC power input (8-volt to 30-volt power is allowed for legacy accessories). FireWire pins require an 8 watt power supply. **Developers of new accessories that charge Apple devices are required to design accessories that use the USB power pins for charging instead of the FireWire pins. Existing products that supply FireWire power must be updated to supply USB power instead.** See "FireWire to USB Reference Design" (page 99).

# Deprecated ID Resistors

The $R_{ID}$ resistor values shown in Table F-2 (page 115) are deprecated. Acceptable $R_{ID}$ values are listed in "Accessory Detect and Identify" (page 36).

**Table F-2** Deprecated RID values

| iAP transport | Resistor value | Functional behavior and requirements |
|---|---|---|
| iAP over UART; power only | 255 kΩ | The Apple device draws power from the accessory but does not charge its internal battery. Its user interface displays the battery level as full, instead of charging. The accessory must use the D+ and D- resistors specified in Figure 2-1 (page 30). The iPhone requires firmware version 1.1.3 or later. The accessory must ground Accessory Detect (pin 20). **This function is deprecated;** use the iAP General lingo `SetInternalBatteryChargingState` command instead. |
| iAP over UART; pause on power removal | 1 MΩ | The Apple device pauses music playback when USB power is removed. With iOS devices, the iOS application enters Pause mode. The iPhone requires firmware version 1.0.1 or later. The accessory must ground Accessory Detect (pin 20). **This function is deprecated;** use the iAP General lingo `SetiPodPreferences` command instead. |
| iAP over USB; alternate USB Device mode | 191 kΩ | A 191 kΩ ID resistor may be used only in systems where there is no doubt about which USB device is connected. In particular, it must never be used in an accessory or cable that might be connected directly to a standard USB port. The accessory must ground Accessory Detect (pin 20). **This function is deprecated; do not use.** |
| N/A | 3.01 kΩ | The 3G, 4G, and mini iPods will beep when connected. **This function is deprecated; do not use.** |
| No iAP transport | 360 kΩ | The Apple device charges its own battery from the accessory's power supply. The LINE-OUT signals are disabled. The iAP interface must not be used. **This function is deprecated; do not use.** |

# Document Revision History

This table describes the changes to *MFi Accessory Hardware Specification*.

| Date | Notes |
|---|---|
| 2012-02-21 | *Revision R9:* Updated for iPad (3rd generation). |
| | Added a specification for "USB Cables" (page 66). |
| | Updated Bluetooth autopairing information in "Specification Update" (page 62). |
| | Changed last cell of Table 2-10 (page 37) per *Update to MFi Accessory Hardware Specification R8* dated 11-10-21. |
| | Corrected reversed descriptions of switches S1 and S2 in Table 5-7 (page 82). |
| | Clarified Important note under "Choosing an Apple Device USB Configuration" (page 53). |
| 2011-10-14 | *Revision R8:* Updated for iPhone 4S. |
| | Reorganized "Protocol Transports" chapter and renamed it "Command and Data Transports" (page 51). |
| | Added Serial Copy Management System requirements to "USB Audio" (page 60) and new section "Audio Output From Accessories" (page 65). |
| | Modified section "Accessory Power Policy" (page 47) to remove IDPS registration of power modes and eliminate Constant High Power mode. |
| | Added new section "AssistiveTouch Accessory Requirements" (page 68). |
| | Added sensor information to Apple device model descriptions in Table I-1 (page 12). |
| | Deprecated $R_{ID}$ resistors previously listed in Table 2-10 (page 37) and listed them in Table F-2 (page 115) for historical reference. |
| | Removed references to deprecated Accessory Power lingo from Table 2-16 (page 48) and Table 2-18 (page 49). |
| | Removed sections "Protocol Compatibility," "Communication and Commands," and "User Interface Restrictions" from Appendix D, "Interfacing With the 3G iPod." This content is duplicated in *MFi Accessory Firmware Specification*. |
| | Required the use of USB for all MIDI data transport. |

| Date | Notes |
|---|---|
| | Updated section "Bluetooth" (page 61) to permit the connection of multiple Bluetooth accessories. |
| | Changed communication terminology to refer to "transports" instead of "transport links." |
| 2011-04-04 | *Revision R7:* Updated for iPad 2, iPhone 4 (CDMA model), and iOS 4.3.1. |
| | Updated "Notice of Proprietary Property" (page 11). |
| | Incorporated contents of *Update to MFi Accessory Hardware Specification* (dated 2010-11-30), which updated sections "General Requirements" (page 58), "USB Audio" (page 60), "USB MIDI" (page 61), and "Accessory Power Policy" (page 47). |
| | In "Cases for Apple Devices" (page 66), referenced compliance with Apple's *Case Design Guidelines for Apple Devices* as a requirement for MFi certification. |
| | Deprecated analog audio line level inputs; see Table 1-1 (page 19). |
| | Revised section "Accessory Power Policy" (page 47). |
| 2010-11-29 | *Revision R6:* |
| | Corrected generic references to Apple devices, iOS devices, and iPods throughout the document; see "IMPORTANT" (page 11). |
| | Consolidated appendix "Apple Device Power States and Accessory Power" into Chapter 2; see "Accessory Power" (page 35), "Apple Device Power States" (page 43), and "Accessory Power Policy" (page 47). |
| | Renamed chapter "iPhone Accessory Design Guidelines" to "Accessory Design Requirements" (page 65). |
| | Added new section "iPod Out Accessory Requirements" (page 68). |
| | Added new section "Cases for Apple Devices" (page 66). |
| | Added new section "USB HID" (page 61). |
| | Added new section "Informing Apple Devices of Available Power" (page 33). |
| | Revised the Apple device feature tables; see Table 2-1 (page 25), Table 2-2 (page 26), Table 2-3 (page 27), and Table 2-4 (page 28). |
| | Renamed pin 30 of the 30-pin connector from "iPod Detect" to "Apple Device Detect"; see Table 1-1 (page 19). |
| | Restructured section "USB Host Mode" (page 58). |
| | Referenced Apple document *Bluetooth Accessory Design Guidelines for Apple Products* for some Bluetooth connectivity details and noted that its guidelines become mandatory in the context of this specification. |

| Date | Notes |
| --- | --- |
| 2010-09-08 | *Revision R5:* Updated for 6G nano and 4G touch. |
| | Added new section "General Requirements" (page 61). |
| | Updated Table I-1 (page 12). |
| | Updated "USB 2.0" (page 29). |
| | Clarified resistor usage and removed 549 kΩ resistor in Table 2-10 (page 37). |
| | Clarified digital audio volume control in "USB Audio" (page 60). |
| | Clarified generic references to iPods and iOS devices; see "IMPORTANT" (page 11). |
| 2010-06-24 | *Revision R4:* Update for iPhone 4. |
| | Changed document name from "iPod/iPhone/iPad Hardware Specifications" to "MFi Accessory Hardware Specification." |
| | Changed name of "iPhone OS" to "iOS." |
| | Revised section "Accessory Detect and Identify" (page 36). |
| | Revised section "USB" (page 52). |
| | Revised "iPod power states." |
| | Revised section "Avoiding RX Back-Powering" (page 35). |
| | Added appendix "DisplayPort Digital Video Connectivity" for future products. |
| 2010-04-09 | *Revision R3:* Added information for the iPad. |
| | Added new appendix A, "Hardware Interfaces for the iPad." |
| | Added new section "USB Power" (page 30). |
| | Added new section "USB Synchronization Cables." |
| | Clarified uses of pins 8 and 30 in Table 1-1 (page 19). |
| | Updated Figure 5-6 (page 81) and Figure 5-7 (page 82). |
| | Updated "Wireless Standards" (page 23). |
| | Made other minor updates and clarifications. |
| 2009-10-22 | *Revision R2:* Added information for the 5G nano and iPod classic 160 GB. |
| | Updated line level output values in Table 2-12 (page 41). |
| | Expanded documentation of UART signal levels in Table 3-1 (page 51). |

| Date | Notes |
| --- | --- |
| | Updated section "Bluetooth" (page 61) throughout. |
| | Clarified requirements for accessories to support Accessory Detect and Accessory Identify. |
| | Updated and clarified section "Resistor-Based Accessories." |
| | Removed documentation of obsolete 9-pin Audio/Remote connector. |
| 2009-09-09 | *Revision R1:* First release. |
| | Imported most of the content from *iPod Accessory Protocol Interface Specification*, Release R36. |
| | Imported some content from discontinued book *iPhone Accessory Interface Specification*, Release R9. |

# Glossary

**accessory** A third-party device licensed under the Made for iPod program.

**authentication** A mechanism used by an Apple device to verify whether an attached accessory is an authorized accessory and by an accessory to authenticate the Apple device, if desired.

**checksum** The byte sum of packet bytes from the payload length through the last packet byte. This is used to validate the contents of a command packet. For a valid packet, the sum of the bytes, including the checksum byte, must be `0x00`. The packet checksum byte—the last byte in a packet—must be the 2's complement (the negative) of the sum of the payload length byte up to, but not including, the packet checksum byte.

**deprecated** Used to describe a technology or feature that is supported but whose use is discouraged and not recommended. Such a technology or feature has typically been replaced by a newer one and is likely to become unsupported in the future.

**HID (Human Interface Device)** HID is a standard USB class. A USB host such as a PC or Macintosh will recognize any attached USB device that supports a HID interface and makes it available to the application layers of the operating system via a set of programming interfaces. A common application of a HID interface is a USB mouse or joystick.

**HID report** A single unit of data that is used to send information to the HID interface of an Apple device or from an Apple device to the host. iAP packets are broken into HID reports before being sent across the USB transport and are reassembled on the receiving side.

**iPod photo** Alternative product name for the 4G iPod with color display.

**iUI (iPod USB Interface)** A configuration of an Apple device when attached as a device over USB. This configuration allows the Apple device to be controlled using iAP, using a USB HID class interface as a transport mechanism.

**LCB (Link Control Byte)** A byte used by the iUI to indicate report sets and manage data flow.

**lingo** The command category used by an accessory. There is a General lingo that must be supported by all accessories. Other lingoes are designed for use by specific accessories, such as simple remote controls and microphones.

**low power mode** An operating mode of an accessory in which it draws 5 mA or less from an attached Apple device.

**packet** The logical set of bytes that compose a valid command sequence. This set includes the packet start byte, packet payload length, payload, and payload checksum. Note that a sync byte is appended to the beginning of the packet when using UART transport. There are two different packet types: small format and large format.

**payload** The sequence of bytes consisting of the lingo, command, and data that are contained within a packet.

**podcasting** A way to publish multimedia files on the Internet that lets users receive new files automatically by subscription. Podcast files are typically downloaded to Apple devices through Apple's iTunes application.

**RDS/RBDS (Radio [Broadcast] Display System)** A technology for broadcasting and displaying artist, album, track titles, and similar information on FM radio receivers.

**resistor-based accessory** An accessory that uses an Accessory Identify resistor to access only limited functions in an Apple device. Compare Serial accessory.

**RSSI (Receive Signal Strength Indicator)** A measure of the strength of an RF signal coming into a radio frequency tuner.

**serial accessory** An accessory that uses the iPod Accessory Protocol Interface to access a range of Apple device functions. Compare Resistor-based accessory.

**UART (Universal Asynchronous Receiver/Transmitter)** A piece of computer hardware that translates between parallel and serial bits of data. A UART is usually an integrated circuit used for serial communications over computer or peripheral device serial transport.

**USB (Universal Serial Bus)** An interface standard for communication between a computer and external peripherals over a cable using biserial transmission.

**USB descriptor** A standard USB data structure that is passed from a USB device to the host upon request. Descriptors are used by the USB device to communicate its characteristics and resource requirements to the host.

**USB endpoint** A logical connection point that is used to set up a data transfer pipe between a USB host and the interface on a USB device. For instance, the HID interface on Apple devices uses an interrupt-type endpoint to enable a pipe for transferring data to the USB Host.

**USB host** A single computer connected to one or more USB devices or functions. The host is responsible for recognizing that a USB device has been attached to it and for driving the communications with the device. For the purposes of this document, an Apple device is a USB device that provides a function, and the accessory is the USB host. However, the Apple device can also be made a USB host with the accessory acting as its USB device.

**X.509 certificate** A standard defined by the International Telecommunication Union (ITU) that governs the format of certificates used for authentication and sender identity verification in public-key cryptography. In the iAP, X.509 certificates contain the public keys used in the authentication process.