

Revision History

Revision	RFA	History	Initial	Date
03	N/A	Initial Release	DJN	12.04.02
04	N/A	Accessory Detect Resistor Change	DJN	12.04.02
05	N/A	Rx, Tx Clarification	DJN	01.09.03
06	N/A	Accessory Detect Resistor Change	DJN	02.04.03
07	N/A	Car Charger Detect added	DJN	04.02.03
08	N/A	Remote protocols added	DJN	04.15.03
09	N/A	License agreement	DJN	07.07.03
10	N/A	Picture to show Remote Data lines	DJN	07.22.03
11	N/A	New serial, add bottom serial	SJB	08.04.03
12	N/A	5mA access power note	SJB	08.12.03
13	N/A	Minor clarifications	JLD	01.14.04
14	N/A	Update simple remote, doc reformat	JWA	04.20.04
15	N/A	Incorporated review feedback	JWA	05.17.04
16	N/A	Added lingo command packet examples, corrected doc properties, added minor clarifications, added new accessory detect image, updated content based on internal review	GTL	07.21.04
17	N/A	Reserved the 28k pulldown resistor	LGB	08.03.04
18	N/A	Added general lingo commands, bug fixes and pinouts for the iPod Photo release	LGB	11.12.04

iPod Interface Specification

NOTICE OF PROPRIETARY PROPERTY:

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

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

1.0 SCOPE

This document specifies the electrical and software interfaces to both the Omni (bottom) and Audio/Remote (top) connectors to the iPod. The features described in this specification are supported in versions 2.1 and 2.2 of the third generation iPod (first iPod with Omni connector), versions 1.1 and 1.2 of the iPod Mini, versions 3.0 and 3.0.1 of the fourth generation iPod (iPod with gray click wheel) and version 1.0 of the fifth generation iPod (color/photo iPod), as well as subsequent firmware versions unless otherwise noted.

3rd Generation iPod



iPod Mini



4th Generation iPod



iPod photo



2.0 APPLICABLE DOCUMENTS

- a) IEEE 1394 A (latest revision)
- b) USB 2.0 High Speed (latest revision)

3.0 GENERAL REQUIREMENTS

3.1 CONNECTOR PIN DESIGNATIONS:

3.1.1 OMNI: 30 pin connector (on bottom of iPod and iPod Mini units)

NOTE: For physical connector layout and dimensions, refer to Appendix B.

Pin	Signal name	I/O	Function
1	F/W GND	I	Firewire and charger ground
2	F/W GND	I	Firewire and charger ground
3	TPA+	I/O	Firewire signal
4	USB D+	I/O	USB signal
5	TPA-	I/O	Firewire signal
6	USB D-	I/O	USB signal
7	TPB+	I/O	Firewire signal
8	USB PWR	I	USB power in used to detect a USB host.
9	TPB-	I/O	Firewire signal
10	Accessory Identify	I	Pull down to notify iPod of specific device

11	F/W PWR+	I	Firewire and charger input power (8v to 30v dc)
12	F/W PWR+	I	Firewire and charger input power (8v to 30v dc)
13	ACCESSORY PWR(3V3)	O	3.3v is nominal output from iPod. Current limited to 100mA.
14	Reserved		
15	USB GND	GND	Digital ground in iPod
16	DGND	GND	Digital ground in iPod
17	Reserved		
18	Tx	I	Serial data to iPod
19	Rx	O	Serial data from iPod
20	Accessory Detect	I/O	(See Functional description accessory detect for details)
21	S Video Y	O	The luminance component of S Video (iPod Photo only)
22	S Video C	O	The chrominance component of S Video (iPod Photo only)
23	Composite Video	O	iPod Photo only
24	Remote Sense	O	*SEE NOTE BELOW TABLE
25	LINE-IN L	I	Line level input to the iPod for the left channel
26	LINE-IN R	I	Line level input to the iPod for the right channel
27	LINE-OUT L	O	Line level output to the iPod for the left channel
28	LINE-OUT R	O	Line level output to the iPod for the right channel
29	Audio Return	GND	Audio return – Signal, never to be grounded inside accessory
30	DGND	GND	Digital ground in iPod
31	Chassis		Chassis ground for connector shell
32	Chassis		Chassis ground for connector shell

NOTE: The remote sense function minimizes crosstalk between the analog video and analog audio outputs. The remote sense returns a low side of load voltage that is impressed on the audio to suppress crosstalk and to provide optimal analog audio and analog video. Routing and connection of remote sense is critical to achieve the best results. Haphazard application of remote sense will yield poor results. Apple Computer, Inc. must review accessories using analog video and analog audio.

3.1.2 AUDIO/REMOTE: 9 pin Apple Custom (top of iPod and iPod Mini units)

NOTE: For physical connector layout and dimensions, refer to Appendix B.

Pin 2 (Headphone Detect) is not accessible on the external connector contacts – it is only available internally to the iPod. Only 8 pins are accessible external to the iPod.

Pin	Signal name	I/O	Function
1	Audio Out Left / Mono Mic In	I/O	30mW audio out left channel, also doubles as mono mic in
2	Headphone Detect	I	Internal Switch to detect headphone plug insertion
3	Audio Return	GND	Audio return for left and right audio

4	Audio Out Right	O	30mW audio out right channel
5	Composite Video	O	iPod Photo only
6	Accessory 3.3V	O	3.3V Accessory power 100mA max
7	Tx	O	Serial protocol (Transmit Data from iPod to Device)
8	Rx	I	Serial protocol (Receive Data to iPod from Device)
9	D GND	GND	Digital ground for accessory

4.0 FUNCTIONAL DESCRIPTION

4.1 Inspection:

Devices that use this specification shall be made available to the iPod engineering prior to product release.

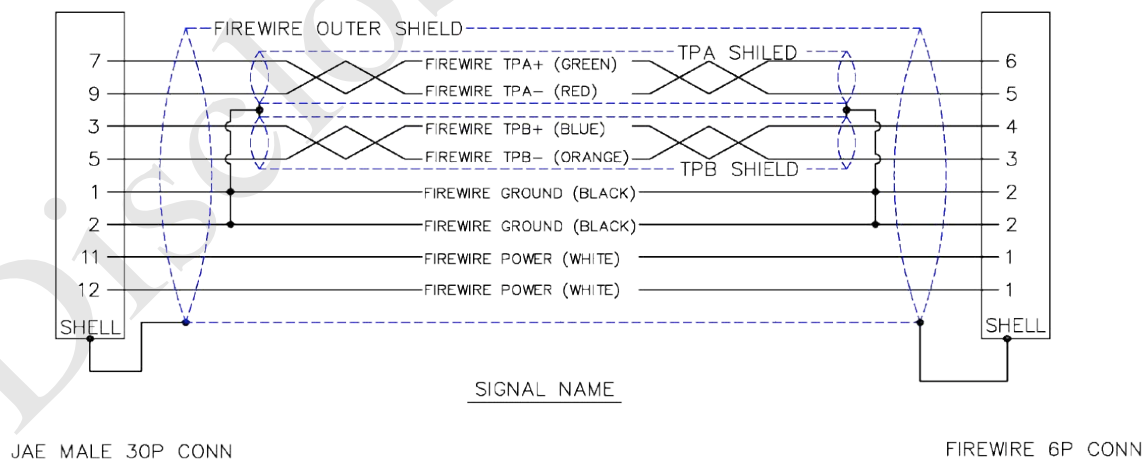
4.2 Omni Connector Functional Description:

4.2.1 FireWire Power:

- 8V – 30V DC IN
- 8W Max

4.2.2 FireWire:

- Designed to IEEE 1394 A Spec (400 Mb/s)
- Note Digital twisted pair of wires need to be reversed per IEEE 1394 A spec, please refer to below 30P to FireWire cable diagram:

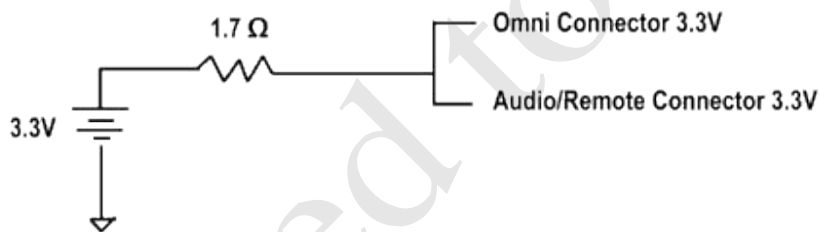


4.2.3 USB 2.0:

- a) Designed to USB 2.0 High Speed Spec

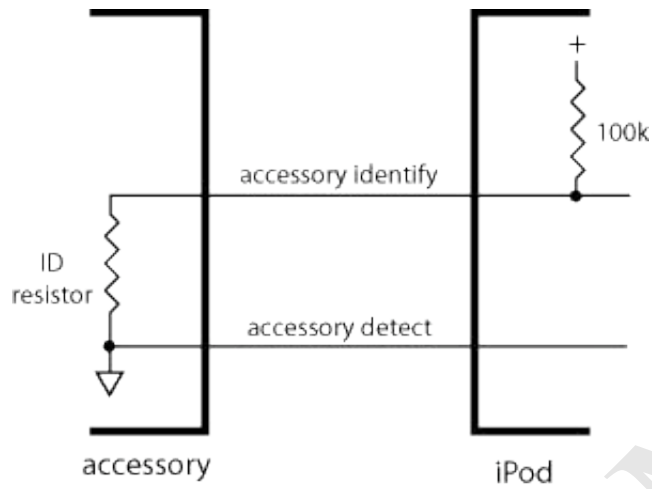
4.2.4 Accessory 3.3V Power:

- a) 3.0V – 5% (2.85V) through 3.3V + 5% (3.465V) @ 100mA max power allowed
- b) Note – 100mA max shared between Omni & Audio/Remote connectors
- c) Note – 5mA MAX current default, proper SW accessory detect will turn on supply to allow 100mA during active device usage. When devices are inactive, they must consume less than 5mA current.
- d) Note – Accessory power is switched off for a period of approximately 2 seconds during the iPod bootstrap process. This is done to ensure accessories are in a known state and can be detected properly. All accessories are responsible for re-identifying themselves after the iPod completes the bootstrap process.
- e) Grounded through F/W GND
- f) Reference Schematic Diagram below:



4.2.5 Accessory Detect/Identify:

- a) A simple resistor to ground allows the iPod to determine what has been plugged into Omni connector. There is an internal pullup on Accessory Identify.
- b) Two pins required (Accessory Identify & Accessory Detect)
- c) Reference Schematic Diagram below:



NOTICE OF PROPRIETARY PROPERTY:

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

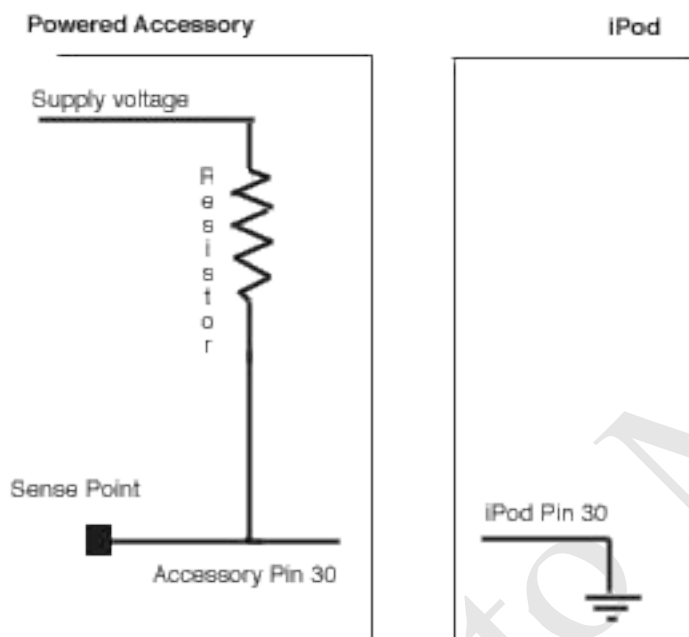
ACCESS TO THIS DOCUMENT AND THE INFORMATION CONTAINED THEREIN IS GOVERNED BY THE TERMS OF THE *IPOD CONNECTOR USE LICENSE AGREEMENT* AND/OR THE *IPOD TECHNOLOGY EVALUATION LICENSE AGREEMENT*. ALL OTHER USE SHALL BE AT APPLE'S SOLE DISCRETION.

d) Below are the Specified Resistor values for known accessories:

Accessory Identify Resistors calculations					
Internal pullup (k):	Internal pullup (k):				
100	100			A/D RANGE	
nominal value for	choose a 1%	FUNCTION:	MIN	TYP	MAX
Pulldown resistor (K):	pulldown resistor (K):				
3.03	3.01	Simple Dock	0	7.5	15
10.15	10.2	Reserved	16	23.6	31
18.33	18.2	Reserved	32	39.3	47
27.82	28.0	Reserved	48	55.8	63
38.96	39.2		64	71.8	79
52.24	52.3		80	87.6	95
68.32	68.1		96	103.3	111
88.19	88.7	Diag Dock	112	119.9	127
113.39	113		128	135.3	143
146.38	147		144	151.8	159
191.43	191		160	167.4	175
256.64	255	Battery Pack	176	183.2	191
359.46	360		192	199.6	207
545.57	549	UART Dock	208	215.7	223
985.11	1000	Car Charger	224	231.8	239
3300.00	3010	unusable	240	246.8	255

4.2.6 iPod Detect for Powered Accessories

Powered accessories can be manufactured to detect that an iPod has been plugged in, even when the iPod has no battery power. To detect an iPod a powered accessory 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 iPod is attached, and it will be low when an iPod is attached because pin 30 on the iPod is tied to ground.



4.2.7 Serial Protocol Communication:

- Two pins used to communicate to and from iPod (Rx & Tx)
- Input & Output (0V=Low; 2.85V - 3.465V=High)
- A device with a 549k Ω identity resistor is a serial dock accessory. These devices use the standard serial protocol but only through the bottom/Omni connector. Attaching a serial dock accessory to a 3G iPod makes any top-attached (Audio/Remote) accessories inactive. The 4G, 5G and iPod Mini models have two serial ports so plugging in a serial dock accessory will not inactivate the top port.

4.2.8 Line Level Input (Left & Right):

- Stereo audio input
- Input Level 0.909 V_{rms} (2.570 V_{pp}) maximum
- Audio Return: NOT a ground – treat it as a signal. Left and right audio are referenced to Audio Return.

4.2.9 Line Level Output (Left & Right):

- Stereo audio output does not have volume control and is not adjusted by device

- b) Output level 1V RMS (max)
- c) Audio Return: NOT a ground – treat it as a signal. Left and right audio are referenced to Audio Return.

4.2.10 Overall Grounding Requirements:

- a) Chassis ground is tied to specified pins
- b) Digital ground should not be tied to Audio Return

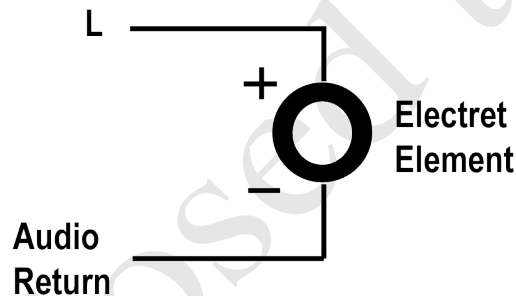
4.3 Audio/Remote Functional Description:

4.3.1 Audio Out:

- a) Stereo Output ~30mW per channel when driven into a 32 Ω load
- b) Output volume controlled by iPod

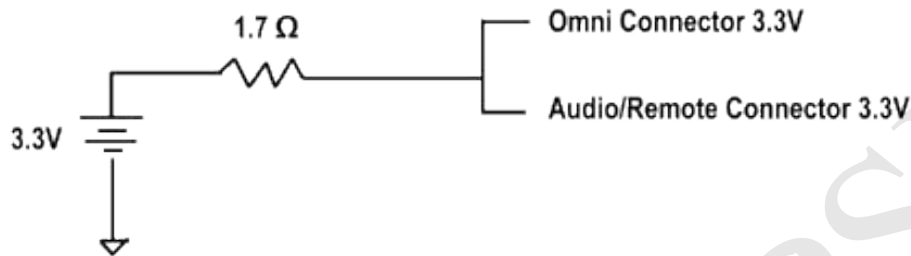
4.3.2 Mono Mic In (depending on the mic element a preamp may be required):

- a) Mono mic in through Left channel
- b) Filtered electret power supplied by internal device
- c) Reference schematic below:



4.3.3 Accessory 3.3V Power (same as Omni connector):

- a) 3.0V – 5% (2.85V) through 3.3V + 5% (3.465V) @ 100mA max power allowed
- b) Note – 100mA max shared between Omni & Audio/Remote connectors
- c) Note – 5mA, max current is the default; proper software accessory detect will turn on supply to allow 100mA.
- d) Grounded through D GND
- e) Reference Schematic Diagram below:



SERIAL PROTOCOL

4.4 Overview

The Serial Protocol allows the iPod to communicate with a functional range of external devices. This specification breaks the protocol into two logical components: the Protocol Core and the Function Specific. The Protocol Core is common to the communication with all external devices and encompasses the basic packet definition, initial identification and the iPod information commands. The Function Specific component consists of different Lingo or dialects where each one corresponds to a functional class of external device. The iPod will only communicate in one Lingo or device type command set at a time.

Every external device will support the Protocol Core and a single Lingo. The iPod standard in-line remote control is an example of a device that uses the Simple Remote lingo.

The iPod will receive command packets via the serial port from the external device and then react by changing the iPod internal state and/or sending a response packet to the requesting device. This means that an external device must initiate communications with an iPod by sending one or more command packets until it is recognized by the iPod.

From a hardware perspective, the protocol builds upon the RS-232 serial specification. However, the signaling levels are non-standard. In true RS-232, a mark is $-7V$ and a space is $+7V$. In this protocol, a mark is 2.85V through 3.465V and a space is 0V.

Apple must have the opportunity to evaluate every accessory before release, in part to verify correct operation.

4.5 Protocol history and compatibility

This specification is valid for System Software versions 2.1 and 2.2 on the third generation iPod, versions 1.0 and 1.1 on the iPod Mini, versions 3.0 and 3.0.1 of the fourth generation iPod (gray click wheel) and version 1.0 of the fifth generation iPod (iPod Photo). In pre-2.1 software, the RF Transmitters lingo (lingo 0x05) is not supported.

Version 1.0 of the iPod Photo software fixes a problem where iPod Accessory Protocol commands on the bottom connector did not wake up unpowered, sleeping iPods.

Both iPod software version 2.1 and iPod Mini software version 1.0 (and later versions) support a small packet format payload maximum length of 255 bytes. Earlier software versions of both iPods were limited to a 127 byte small packet format payload maximum length. The older iPod versions did not support lingos that required a bigger payload length so this should not be a major issue.

iPod Software Version Release Dates:

iPod Features/Model	3G Firmware	Mini Firmware	4G Firmware	5G (Photo) Firmware
Firmware Release Dates				
2003, April	2.0	N/A	N/A	N/A
2003, October	2.1	N/A	N/A	N/A
2004, February	2.2	1.0	N/A	N/A
2004, March	-	1.1	N/A	N/A
2004, July	-	-	3.0	N/A
2004, August	-	-	3.0.1	N/A
2004, October	-	-	-	1.0

iPod Supported Features Table:

iPod Features/Model	3G Firmware	Mini Firmware	4G Firmware	5G (Photo) Firmware
Lingo 0x00: General				
Protocol version 1.0	-	-	-	1.0
identify cmds: 0x00-0x01	2.0	1.0	3.0	1.0
Advanced cmds: 0x02-0x10	-	-	-	1.0
Lingo 0x01: Microphone				
Protocol version 1.0	-	N/A	-	1.0
All cmds, top connector	2.1	N/A	3.0	1.0
All cmds, bottom connector	-	N/A	-	1.0
Lingo 0x02: Simple Remote				
Protocol version 1.0	-	-	-	1.0
Buttons 00-04, top connector	2.0	1.0	3.0	1.0
Buttons 05-25, top connector	-	-	3.0	1.0
Buttons 00-04, bottom connector	2.0	1.0	3.0	1.0
Buttons 05-25, bottom connector	-	-	3.0	1.0
Button wakeup, bottom connector	-	-	-	1.0
Lingo 0x03: Reserved				
Lingo 0x04: Reserved				

Lingo 0x05: RF Transmitter				
Protocol version 1.0	-	-	-	1.0
All cmds, top connector	2.1	1.0	3.0	1.0
All cmds, bottom connector	-	-	-	1.0
Other HW/SW Features				
Serial baud rates: 38400/57600	-	-	3.0.1	1.0

NOTE: This table shows the first version of iPod Software that supports each feature for each iPod model. Support for these features is carried forward into all subsequent releases of software for each model.

4.6 Protocol terms used

Checksum – The byte sum of packet bytes from the payload length through the last packet byte 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) should be the 2's complement (negative) of the sum of the payload length byte up to, but not including, the packet checksum byte.

Device – An external electronic component connected to the iPod using the bottom/omni connector or the top/audio/remote connector.

Lingo – The command category type used by a device. There is a general lingo that must be supported by all devices and other lingos designed for use by specific devices such as simple remote controls, microphones, etc.

Link – The logical connection between external device and iPod via serial port or other physical connection.

Packet – The logical set of bytes that compose a valid command sequence (includes sync byte, packet start byte, packet payload length, payload, and payload checksum). 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.

4.7 Serial Port Hardware Protocol

The serial baud rate for this protocol is one of the following standard rates: 9,600bps, 19,200bps, 38400bps or 57600bps. The iPod will automatically detect which of the supported baud rates the external device is using and adjust its own baud rate accordingly. All serial communications use 8 data bits, no parity and one stop bit (8-N-1). Serial hardware flow control (i.e., RTS/CTS and DTR/DSR) and XON/XOFF protocols are not used and will be ignored.

4.8 Remote command protocol core

4.8.1 Link specifics

This protocol is to be used in both directions of a link. Every device is encouraged to implement both sending and receiving capabilities. It should be possible to determine the direction (device to iPod or iPod to device) of a packet from its packet contents only. This means that no packet is valid for sending from both the iPod and the device.

All devices must be able to handle variable-length packets. For example, even though most identify packets currently have no defined data, a device must be able to understand an identify packet with extra data and should respond to the best of its ability. It must at least not lose sync to the packet signaling.

4.8.2 Extra-packet signaling and initialization

Upon power up a device should wait 80ms and then send a sync byte (see below). It should then wait 20ms more and then identify itself by sending the Identify packet.

A device must re-identify itself if a request identify command is received from the iPod.

Once packet transmission has begun, the maximum time between transmitted data bytes must be less than 20 milliseconds. If the delay between characters exceeds 20ms, the packet characters already received and any remaining characters received as part of the same packet will be discarded until the start of the next valid packet.

One known limitation exists in the iPod Mini, 4th Generation and Photo iPods concerning waking these devices from sleep. The UART of iPod is not available for the first 1.6 mS after waking from sleep. If an external device sends a packet to the iPod while it is asleep, the first packet will be lost. Please follow the steps below when an accessory device needs to wake up the iPod and wants to ensure the first packet will wake the device:

1. Send a sync byte. (This should wake up iPod)
2. Wait for 10 mS.
3. Send the command packet with sync byte.

4.8.3 Command Packet Formats

4.8.3.1 Small packet format (up to 255 byte payloads)

For command packets whose payloads are 255 bytes or less, the small packet format should be used. The small packet format is shown below:

Byte number	Value	Meaning
0x00	0xFF	Sync byte
0x01 (SOP)	0x55 ('U')	Packet start byte
0x02	0xNN	Packet payload length
0x03	0xNN	Lingo ID
0x04	0xNN	Command ID
0x05...0xNN	0xNN	Command data
(last byte)	0xNN	Packet payload checksum

Note that the Command ID and command data format for packets with as-yet unspecified Lingo IDs may not follow the packet payload format (1 byte Command ID, 0xN bytes Command data) indicated here. Also note that a packet payload length of 0x00 is not valid for the small packet format (it is reserved as a marker for the large packet format).

4.8.3.2 Large packet format (256 – 65535 byte payloads)

For command packets whose payloads are between 256 bytes and 65535 bytes in length, the large packet format should be used. The large packet format is shown below:

Byte number	Value	Meaning
0x00	0xFF	Sync byte
0x01 (SOP)	0x55 ('U')	Packet start byte
0x02	0x00	Packet payload length marker
0x03	0xNN	Packet payload length (bits 15:8)
0x04	0xNN	Packet payload length (bits 7:0)
0x05	0xNN	Lingo ID
0x06	0xNN	Command ID
0x07...0xNN	0xNN	Command data
(last byte)	0xNN	Packet payload checksum

4.8.3.3 Packet details

The sync byte (0xFF) is not considered part of the packet. It is sent merely to facilitate automatic baud rate detection/correction and, in some cases, to wake up the iPod. The packet payload length is the number of bytes in the packet not including the sync byte, packet start byte, packet payload length byte, or packet payload checksum byte. That is, it is the length of the Command ID plus Lingo plus the Command data. Thus the packet payload data length for a request identify command would be 0x02. The Lingo ID specifies the broad category this communication falls under. The Command ID is a more specific indication of the significance of the packet and is interpreted differently based on the Lingo ID.

NOTE: The 16 bit large packet format payload length is sent in big endian order – the high byte of the length is sent first, followed by the low byte of the length.

The sum of all the bytes from the packet payload length (marker, if applicable) through the packet payload checksum is 0x00. The checksum should be calculated appropriately (add the bytes together as signed 8 bit values, discard any signed 8 bit overflow, and then negate the sum to create the signed 8 bit checksum byte). **NOTE: All received packets with a non-zero checksum are presumed to be corrupted and will be discarded.**

4.8.4 Lingo types

The iPod interface lingo types are shown in the table below:

Lingo	ID
General (all devices)	0x00
Microphone	0x01
Simple Remote	0x02
Reserved (do not use)	0x03 – 0x04
RF transmitter	0x05
Reserved for future use	0x06 – 0xFF

The General lingo is intended for housekeeping commands and must be supported by all devices. In addition to the general lingo, an external device will implement a function-specific lingo. A microphone device attached to the mono Audio/Remote connector on the iPod uses the Microphone lingo (0x01). The Simple Remote lingo (0x02) is used by Apple's simple in-line remote control. An external RF transmitter device uses the RF transmitter lingo (0x05).

4.8.5 General lingo commands

Command	ID	Data length	Protocol Ver.
Request identify	0x00	0x00	All
Identify	0x01	0x01	All
ACK	0x02	0x01 or 0x05	1.0
Reserved (do not use)	0x03-0x06	N/A	N/A
Request iPod Name	0x07	0x00	1.0
Return iPod Name	0x08	0xNN	1.0
Request iPod Software Version	0x09	0x00	1.0
Return iPod Software Version	0x0A	0xNN	1.0
Request iPod Serial Number	0x0B	0x00	1.0
Return iPod Serial Number	0x0C	0xNN	1.0

Request iPod Model Number	0x0D	0x00	1.0
Return iPod Model Number	0x0E	0x04 + 0xNN	1.0
Request Lingo Protocol Version	0x0F	0x01	1.0
Return Lingo Protocol Version	0x10	0x03	1.0
Reserved for future use	0x11 – 0xFF	N/A	N/A

The iPod may send a request identify to the device to ask the device to re-identify itself. There is no currently defined request identify data.

The device sends an identify packet to identify itself. At this time multifunction (combo) devices are not supported. The identify data payload is the command ID 0x01 followed by a single byte of the same value as the lingo specification of the functionality the device implements unless specified otherwise. The identify packet returned in response to a request identify packet does not need to have the extra sync bytes and delays used during the device startup process.

The remaining general lingo commands can be used to extract general information from the iPod. These commands allow the device to request the iPod's name, serial number, model number and software version number. The request lingo protocol version command allows a device to query the lingo protocol versions for all supported lingoes on the iPod. The ACK command is used by the iPod to report command error conditions and has an ACK pending feature to notify the requesting device how long to wait for responses to certain commands.

To determine whether the iPod connected supports the full general lingo command set a device should send a 'Request Lingo Protocol Version' command, passing lingo 0x00 as the parameter. If the command returned is a 'Return Lingo Protocol Version' packet containing the lingo and lingo protocol version information then the connected iPod supports all of the general lingo commands. If there is no response from the iPod to the request lingo protocol version command then the device should retry once and if no response is received a second time it should be assumed that the connected iPod only supports the Request Identify and Identify general lingo commands. Here is the sequence of the events that should be used to test for the full set of general lingo commands:

Device → iPod	iPod → Device	Comments
Send sync byte (0xFF)		Send sync byte to wake up iPod
Wait 10 ms		Allow iPod to wake from sleep
Request Lingo Protocol command using lingo 0x00 as the parameter		
	Either no response or a Return Lingo Protocol command is returned	If the iPod returns the return lingo protocol command then the entire general lingo command set is

		supported by the connected iPod.
Wait 10 ms		Wait for return packet...
Request Lingo Protocol command with lingo 0x00 as the parameter.		
	Either no response or a Return Lingo Protocol command is returned	If the return lingo protocol command is not received during the retry then the device should assume that only the Request Identify and Identify general lingo commands are supported by the connected iPod.

NOTE: The RF Transmitter lingo defines an extended Identify response packet format that includes option information. Refer to the RF Transmitter lingo section for more details.

4.8.6 General lingo examples

4.8.6.1 Request Identify (iPod -> Device, Cmd: 0x00)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x00	Lingo = 0x00 (General lingo, support required by all devices)
0x00	Command = 0x00 (request device to identify itself)
0xFE	Checksum (length + lingo + cmd + checksum = 0x00)

4.8.6.2 Device Identify (Device -> iPod, Cmd: 0x01)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x03	Length of packet payload (Lingo + Command + LingoSupported)
0x00	Lingo = 0x00 (General lingo)
0x01	Command = 0x01 (device response to identify request)

0x02	LingoSupported = Simple Remote (in this example)
0xFA	Checksum (length + lingo + cmd + lingoSupp + checksumByte = 0x00)

4.8.6.3 ACK (iPod -> Device, Cmd: 0x02)

The ACK command is used to notify the device of command completion status and errors. The ACK command may come in one of two forms depending on the status being returned. If the status is anything except for command pending, then the normal ACK packet structure below is used:

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x04	Length of packet payload (Lingo + Command + Cmd status + Cmd ID ACK'd)
0x00	Lingo = 0x00 (General lingo)
0x02	Command = 0x02
0xNN	Command result status: 0x00 = Success (OK) 0x01 = ERROR: Unknown database category 0x02 = ERROR: Command failed 0x03 = ERROR: Out of resources 0x04 = ERROR: Bad parameter 0x05 = ERROR: Unknown ID 0x06 = Command Pending – see alternate packet format below
0xNN	Command ID being ACK'd
0xNN	Checksum (length + lingo + cmd + cmd status + cmd id ACK'd + checksumByte = 0x00)

If the status returned by the ACK is command pending, then an additional field is added to the ACK packet that represents the amount of time (in milliseconds) that a device should wait to receive the final packet notifying the device that the current command completed or returned an error status.

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x08	Length of packet payload (Lingo + Command + Cmd status + Cmd ID ACK'd + Max pending ms)

0x00	Lingo = 0x00 (General lingo)
0x02	Command = 0x02
0x06	Command result status: 0x06 = Command Pending
0xNN	Command ID being ACK'd
0xNN	Max pending milliseconds (bits 31:24)
0xNN	Max pending milliseconds (bits 23:16)
0xNN	Max pending milliseconds (bits 15:8)
0xNN	Max pending milliseconds (bits 7:0)
0xNN	Checksum (length + lingo + cmd + cmd status + cmd id ACK'd + max pending ms + checksumByte = 0x00)

The device should wait for up to the number of milliseconds specified for a final ACK response when receiving a command pending ACK packet. If no final ACK packet is received the command should be retried.

4.8.6.4 Request iPod Name (Device -> iPod, Cmd: 0x07)

The request iPod name command is sent by the device to retrieve the iPod's name string. The iPod will respond with a 'Return iPod Name' command containing the iPod's name as a null terminated UTF-8 character array.

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x00	Lingo = 0x00 (General lingo)
0x07	Command = 0x07
0xF7	Checksum (length + lingo + cmd + checksumByte = 0x00)

4.8.6.5 Return iPod Name (iPod -> Device, Cmd: 0x08)

This command is sent by the iPod in response to the 'Request iPod Name' message received from the device. The iPod name is encoded as a null terminated UTF-8 character array. If the iPod name has not been modified by the user, it will be returned as "iPod".

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)

0xNN	Length of packet payload (Lingo + Command + iPod name string)
0x00	Lingo = 0x00 (General lingo)
0x08	Command = 0x08
0xNN...	iPod name as null terminated UTF-8 character array
0xNN	Checksum (length + lingo + cmd + iPod name + checksumByte = 0x00)

4.8.6.6 Request iPod Software Version (Device -> iPod, Cmd: 0x09)

The request iPod software version command is sent by the device to retrieve the iPod's software version information. The iPod will respond with a 'Return iPod Software Version' command containing the iPod's major, minor and revision version numbers.

Note: This is the iPod software version and not the version of a particular lingo protocol.

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x00	Lingo = 0x00 (General lingo)
0x09	Command = 0x09
0xF5	Checksum (length + lingo + cmd + checksumByte = 0x00)

4.8.6.7 Return iPod Software Version (iPod -> Device, Cmd: 0x0A)

This command is sent by the iPod in response to the 'Request iPod Software Version' message received from the device. The iPod returns each version number as an individual byte with the major version number sent first.

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x05	Length of packet payload (Lingo + Command + major byte + minor byte + revision byte)
0x00	Lingo = 0x00 (General lingo)
0x0A	Command = 0x0A
0xNN	iPod major version number byte
0xNN	iPod minor version number byte

0xNN	iPod revision version number byte
0xNN	Checksum (length + lingo + cmd + major byte + minor byte + revision byte + checksumByte = 0x00)

4.8.6.8 Request iPod Serial Number (Device -> iPod, Cmd: 0x0B)

The request iPod serial number command is sent by the device to retrieve the iPod's serial number string. The iPod will respond with a 'Return iPod Serial Number' command containing the iPod's serial number as a null terminated UTF-8 character array.

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x00	Lingo = 0x00 (General lingo)
0x0B	Command = 0x0B
0xF3	Checksum (length + lingo + cmd + checksumByte = 0x00)

4.8.6.9 Return iPod Serial Number (iPod -> Device, Cmd: 0x0C)

This command is sent by the iPod in response to the 'Request iPod Serial Number' message received from the device. The iPod serial number is encoded as a null terminated UTF-8 character array.

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0xNN	Length of packet payload (Lingo + Command + iPod name string)
0x00	Lingo = 0x00 (General lingo)
0x0C	Command = 0x0C
0xNN...	iPod serial number as null terminated UTF-8 character array
0xNN	Checksum (length + lingo + cmd + iPod serial number + checksumByte = 0x00)

4.8.6.10 Request iPod Model Number (Device -> iPod, Cmd: 0x0D)

The request iPod model number command is sent by the device to retrieve the iPod's model information. The iPod will respond with a 'Return iPod Model Number'

command containing the iPod's model number as a 32 bit integer (model id) and as a null terminated UTF-8 character array. If an internal memory error occurs while the iPod is processing this command an ACK command containing the error status 'command failed' will be returned. The model id number returned can be used to determine what iPod hardware has been connected. See the description of 'Return iPod Model Number' for details.

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x00	Lingo = 0x00 (General lingo)
0x0D	Command = 0x0D
0xF1	Checksum (length + lingo + cmd + checksumByte = 0x00)

4.8.6.11 Return iPod Model Number (iPod -> Device, Cmd: 0x0E)

This command is sent by the iPod in response to the 'Request iPod Model Number' message received from the device. The iPod model number information is encoded as 32 bit integer (model id) and a null terminated UTF-8 character array. A device should use the model information to determine what type of iPod has been connected, and coupled with the version returned from RequestLingoProtocolVersion, can determine what features of IAP are available.

iPod Model ID	iPod Hardware
0x0003NNNN	3rd Generation iPod (white iPod with 4 buttons above white click wheel)
0x0004NNNN	iPod Mini
0x0005NNNN	4th Generation iPod (white iPod with gray click wheel)
0x0006NNNN	5 th Generation iPod (color/photo iPod)

iPod Model ID String	iPod Hardware
M8976LL	3G - 10GB
M9460LL	3G - 15GB
M9244LL	3G - 20GB
M8948LL	3G - 30GB
M9245LL	3G - 40GB
M9160LL	iPod Mini - 4GB silver
M9437LL	iPod Mini - 4GB gold

M9436LL	iPod Mini - 4GB blue
M9435LL	iPod Mini - 4GB pink
M9434LL	iPod Mini - 4GB green
M9282LL	4G - 20GB
M9268LL	4G - 40GB
M9787LL	4G - 20GB (black U2 special edition)
M9585LL	iPod photo - 40GB
M9586LL	iPod photo - 60GB

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0xNN	Length of packet payload (Lingo + Command + iPod name string)
0x00	Lingo = 0x00 (General lingo)
0x0E	Command = 0x0E
0xNN	iPod Model (bits 24:31)
0xNN	iPod Model (bits 16:23)
0xNN	iPod Model (bits 15:8)
0xNN	iPod Model (bits 7:0)
0xNN...	iPod model number as null terminated UTF-8 character array
0xNN	Checksum (length + lingo + cmd + iPod model number bytes (4 total) + iPod model number array + checksumByte = 0x00)

4.8.6.12 Request Lingo Protocol Version (Device -> iPod, Cmd: 0x0F)

The request lingo protocol version command is sent by the device to retrieve any of the iPod's supported lingo's version information. The iPod will respond with a 'Return Lingo Protocol Version' command containing the major and minor version information of the requested iPod lingo. This command has one parameter, the lingo for which version information should be returned. An ACK command containing a bad parameter status will be returned if an accessory calls this command with an invalid lingo ID or when an iPod Mini requests the version of the microphone lingo (the iPod Mini does not support lingo 0x01). The 'Request Lingo Protocol Version' command should be used by the device to determine what IAP features are available for each lingo used.

Byte	Comment
0xFF	Sync byte (required)

0x55	Start of packet (SOP)
0x03	Length of packet payload (Lingo + Command)
0x00	Lingo = 0x00 (General lingo)
0x0F	Command = 0x0F
0xNN	Lingo of requested version information
0xNN	Checksum (length + lingo + cmd + requested lingo + checksumByte = 0x00)

4.8.6.13 Return Lingo Protocol Version (iPod -> Device, Cmd: 0x10)

This command is sent by the iPod in response to the 'Request Lingo Protocol Version' message received from the device. The major and minor version information for the passed lingo will be returned.

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0xNN	Length of packet payload (Lingo + Command + Lingo for version info + protocol major version + protocol minor version)
0x00	Lingo = 0x00 (General lingo)
0x10	Command = 0x10
0xNN	Lingo for the version information being returned
0xNN	Protocol major version for passed lingo
0xNN	Protocol minor version for passed lingo
0xNN	Checksum (length + lingo + cmd + passed lingo + major version byte + minor version byte + checksumByte = 0x00)

4.9 Function Specific

4.9.1 Microphone lingo specification

This Lingo (0x01) is intended for devices that contain a microphone for voice/sound recording and optionally an external speaker with amplifier for voice/sound playback. Command telegrams are sent to the device based on the iPod recording/playback user interface.

4.9.1.1 Commands

Command	ID	Data length	Protocol Ver.
Begin recording session	0x00	0x00	All
End recording session	0x01	0x00	All
Begin playback session	0x02	0x00	All
End playback session	0x03	0x00	All
Reserved for future use	0x04 – 0xFF	N/A	N/A

Note: The microphone lingo is not supported by the iPod Mini.

iPod sends a begin recording command when recording is about to begin. The device microphone bias (if applicable) should already be present. iPod sends an end recording command when recording is completed. The device may remove microphone bias (if applicable) after the end recording command is received. iPod sends a begin playback command when playback is about to begin. The device may then turn on its speaker amplifier (if present). Upon receipt of an end playback command the device must turn off its speaker amplifier. For all microphone lingo commands, no device response is expected.

4.9.2 Microphone lingo examples

4.9.2.1 Begin Recording Session (iPod -> Device, Cmd: 0x00)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x01	Lingo = 0x01 (Microphone lingo)
0x00	Command = 0x00 (begin recording session)
0xFD	Checksum (length + lingo + cmd + checksum = 0x00)

4.9.2.2 End Recording Session (iPod -> Device, Cmd: 0x01)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)

0x01	Lingo = 0x01 (Microphone lingo)
0x01	Command = 0x01 (end recording session)
0xFC	Checksum (length + lingo + cmd + checksum = 0x00)

4.9.2.3 Begin Playback Session (iPod -> Device, Cmd: 0x02)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x01	Lingo = 0x01 (Microphone lingo)
0x02	Command = 0x02 (begin playback session)
0xFB	Checksum (length + lingo + cmd + checksum = 0x00)

4.9.2.4 End Playback Session (iPod -> Device, Cmd: 0x03)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x01	Lingo = 0x01 (Microphone lingo)
0x03	Command = 0x03 (end playback session)
0xFA	Checksum (length + lingo + cmd + checksum = 0x00)

4.9.3 Simple Remote lingo specification

The simple remote lingo commands are shown in the table below:

Command	ID	Data length	Protocol Ver.
Send Button Status	0x00	0x01 – 0x04	All
Reserved for future use	0x01 – 0xFF	N/A	N/A

This lingo (0x02) is intended for a remote device that retains no state information about the iPod. Simple commands are sent to the iPod and no acknowledgement or state information is sent back to the device. This lingo is used by the iPod standard in-line remote control.

4.9.3.1 History and applicability

NOTE: System Software versions 2.0 through 2.2 on the 3G iPod and versions 1.0 and 1.1 of the iPod Mini only support the first 5 button responses (0-4) in the Button Status command.

4.9.3.2 Button status details

A simple remote device sends a button status command to indicate an updated status of which buttons are held down or released. The data of the packet is a number of bytes indicating which buttons are currently held down. The bytes are made up by ORing the masks of the buttons together. The device will send a full data payload consisting of 0x00 (or no data) to indicate all buttons are released. While any buttons are held down the device should send an update Button Status packet on a 30 - 100ms interval. If no packet of this sort is received by the iPod for 200ms the iPod may assume a packet has been lost and go to “all buttons up” mode. The Simple Remote button map is shown in the table below:

Button Name	Number	Byte Index	Button Bitmask
Play/Pause	0	0x0	0x01
Volume Up	1	0x0	0x02
Volume Down	2	0x0	0x04
Next Track	3	0x0	0x08
Previous Track	4	0x0	0x10
Next Album	5	0x0	0x20
Previous Album	6	0x0	0x40
Stop	7	0x0	0x80
Play/Resume	8	0x1	0x01
Pause	9	0x1	0x02
Mute toggle	10	0x1	0x04
Next Chapter	11	0x1	0x08
Previous Chapter	12	0x1	0x10
Next Playlist	13	0x1	0x20
Previous Playlist	14	0x1	0x40
Shuffle setting advance	15	0x1	0x80
Repeat setting advance	16	0x2	0x01
Power On	17	0x2	0x02
Power Off	18	0x2	0x04
Backlight for 30 seconds	19	0x2	0x08
Begin FF	20	0x2	0x10
Begin REW	21	0x2	0x20
Menu	22	0x2	0x40
Select	23	0x2	0x80
Up Arrow	24	0x3	0x01
Down Arrow	25	0x3	0x02
Reserved for future use	26-31	0x3	0xFC

It is not necessary to transmit any trailing bytes in which no bits are set. If this option is exercised, the length of the packet in the header should be adjusted accordingly i.e., the packet payload length decreased to exclude the trailing zero byte(s) that will not be transmitted.

4.9.3.2.1 Button navigation/programming notes

- A simple remote device should generate the button down status packet immediately, and repeat it every 30 to 100ms as long as the button is pressed, when the user presses and holds down a button. If a second button is pressed while the first button is down, the

button down status packet sent should include status for both buttons, and this packet should be repeated every 30 to 100ms for as long as both buttons are held down.

- Some iPod button down states are interpreted differently by the iPod when pressed and held down for 2 seconds or more:
 - The Next Track button becomes a [Scan Forward] button when pressed and held when a song is playing.
 - The Previous Track button becomes a [Scan Backward] button when pressed and held when a song is playing.
 - The Play/Pause button becomes a [Power Off] button when pressed and held.
 - The Menu button becomes a [Display Backlight On/Off] button when pressed and held.
 - The Select button becomes an [Add Track to Playlist] button when pressed and held, if the iPod is in Browse mode.
- Repeated Next/Prev Track commands (3,4) without intervening “all buttons up” will be interpreted as FF/REW. For a “locking” FF/REW button, use the Begin FF and Begin REW to start and a “Play/Resume” to return to play.
- The Next/Previous Album commands (5,6) have no effect if there are no next/previous album to go to in the list.
- Use the following steps to wake up an iPod when a simple remote button is pressed:
 - 1) Send 0xFF sync byte
 - 2) Wait 10ms
 - 3) Send button status packet
 - 4) Wait 30-100ms
 - 5) Repeat Steps 3 & 4 for as long as any button is pressed
- Multiple button status packets cannot be sent back to back, else the repeated button status packets may be misinterpreted as part of a corrupted packet. Repeated packets must always be separated by a gap of more than 25ms so the iPod knows that the new packet is not part of the previous packet (which may happen if the SYNC and SOP bytes in the first packet have been lost).

4.9.4 Simple remote lingo examples

4.9.4.1 Send [Play/Pause] Button Status (Device -> iPod, Cmd: 0x00)

Byte	Comment
------	---------

0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x03	Length of packet payload (Lingo + Command + ButtonStatus byte 0)
0x02	Lingo = 0x02 (Simple remote lingo)
0x00	Command = 0x00 (Send device button status)
0x01	ButtonStatus byte 0, bit 0 = Play/Pause button pressed
0xFA	Checksum (length + lingo + cmd + buttonStatus + checksum = 0x00)

4.9.4.2 Send [Select] Button Status (Device -> iPod, Cmd: 0x00)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x05	Length of packet payload (Lingo + Command + ButtonStatus bytes 0/1/2)
0x02	Lingo = 0x02 (Simple remote lingo)
0x00	Command = 0x00 (Send device button status)
0x00	ButtonStatus byte 0, no buttons pressed
0x00	ButtonStatus byte 1, no buttons pressed
0x80	ButtonStatus byte 2, bit 7 = Select button pressed
0x79	Checksum (length + lingo + cmd + button status0/1/2 + checksum = 0x00)

4.9.4.3 Send [All Buttons Up] Button Status (Device -> iPod, Cmd: 0x00)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x03	Length of packet payload (Lingo + Command + ButtonStatus byte 0)
0x02	Lingo = 0x02 (Simple remote lingo)
0x00	Command = 0x00 (Send device button status)
0x00	ButtonStatus byte 0, no buttons pressed
0xFB	Checksum (length + lingo + cmd + buttonStatus + checksum = 0x00)

4.9.5 RF Transmitter lingo specification

This lingo is used for devices that transmit the iPod analog audio over radio frequencies (typically an unused frequency in the FM band). The begin transmission command packet notifies the external RF transmitter device that the iPod is entering playback mode. The end transmission command packet notifies the RF transmitter that the iPod is exiting playback mode (e.g., stopped, entering sleep mode, etc.).

This specification has facilities for devices that draw more than 5mA power from the iPod. The conditions under which the device can draw more than 5mA are listed in the Identify packet options field below.

4.9.5.1 Commands

Command	ID	Data length	Protocol Ver.
Reserved (do not use)	0x00 – 0x01	N/A	N/A
Begin transmission	0x02	0x00	All
End transmission	0x03	0x00	All
Reserved for future use	0x04 – 0xFF	N/A	N/A

The General lingo Identify command packet sent by an RF transmitter device contains more than just the device ID. It also includes option bits defining the requirements of the device. The option bit flag byte bit positions are defined as follows:

Option Bit	Description
0	RF transmitter iPod power consumption requirements. 0 = Device requires 5mA or less power from iPod 1 = Device requires more than 5mA power from iPod (up to 100mA max)
1	Reserved for Apple use, must be set to 0 by external devices
2 – 7	Reserved for future use

The option flag byte consists of bitfields. At this time, the most significant 6 bits of the option flags are reserved and should be sent as zeroes. Bit 1 is defined for Apple use only and must be sent as a zero. Bit 0 should be sent as a 1 if the FM transmitter requires more than 5mA during active transmission. This will be the case if the transmitter is powered by the iPod.

The RF Transmitter identify packet format is shown below:

Byte	Comment
0xFF	Sync byte

0x55	Start of packet (SOP)
0x06	Packet payload length (Lingo + Cmd + LingoSupp + Rsvd + valBits + optBits)
0x00	Lingo = 0x00 (General lingo)
0x01	Command = 0x01 (device identify response)
0x05	LingoSupported = RF Transmitter lingo
0x00	Reserved (do not use)
0x02	Number of valid bits in RF transmitter option flag
0x0N	Option flag bits (see option bit flag byte description above)
0xNN	Checksum (length + lingo + cmd + lingoSupp + checksum = 0x00)

Upon receipt of the begin transmitting command, the device may begin drawing more than 5mA (up to 100mA) if it has specified that it wants it in the Identify packet options response. The device must return to less than 5mA power usage within 1 second of receipt of an end transmission packet. **NOTE: Failure to reduce device power consumption to below 5mA within 1 second after receiving the end transmission notification could damage the iPod.**

4.9.6 RF Transmitter lingo examples

4.9.6.1 Begin Transmission (iPod -> Device, Cmd: 0x02)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x05	Lingo = 0x05 (RF Transmitter lingo)
0x02	Command = 0x02 (begin transmission)
0xF7	Checksum (length + lingo + cmd + checksum = 0x00)

4.9.6.2 End Transmission (iPod -> Device, Cmd: 0x03)

Byte	Comment
0xFF	Sync byte (required)
0x55	Start of packet (SOP)
0x02	Length of packet payload (Lingo + Command)
0x05	Lingo = 0x05 (RF Transmitter lingo)
0x03	Command = 0x03 (end transmission)
0xF6	Checksum (length + lingo + cmd + checksum = 0x00)

Appendix A

Device power usage

The power supply to serial devices is a single 2.85V - 3.465V supply. The iPod will always provide accessory power when it is awake or lightly sleeping (not in a deep sleep/power off mode). It will provide 5mA of current by default, but some devices (microphone/speaker or RF transmitter) will be provided high power (100mA) during high consumption periods such as recording or playback. The accessory power will be switched off for two seconds during the iPod bootstrap process and when the iPod can no longer supply it due to low battery conditions.

Simple remotes must use less than 5mA at all times. Simple remotes are very likely to be left on during sleep in order to allow waking from the remote. During sleep, power usage is **very** critical and thus power usage by an inactive remote should be less than 10 μ A.

Microphones must use less than 5mA except when in recording or playback mode.

No device may use more than 100mA at any time.

Appendix B

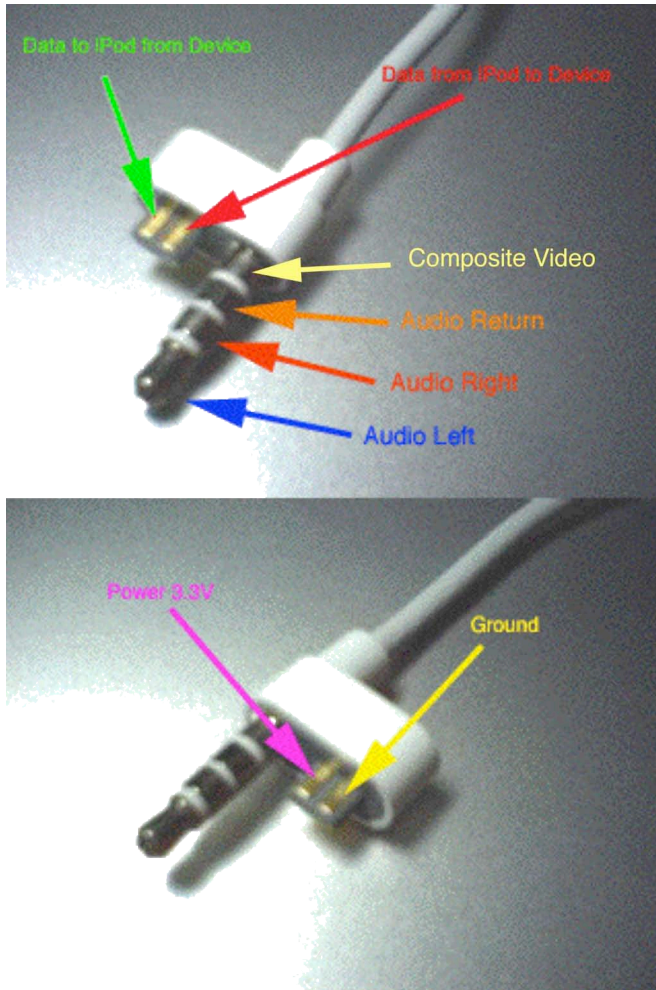
Physical Dimensions and Connector Specifications

30pin “Omni” Connectors

Contact connector vendor for detailed dimensional drawings and CAD/CAM files.

Audio/Remote Connector

Contact connector vendor for detailed dimensional drawings and CAD/CAM files.



Signal Descriptions	
L	Left Channel
R	Right Channel
Ret	Return
GND	Ground
Vcc	Accessory Power
Tx	Transmit
Rx	Receive

NOTICE OF PROPRIETARY PROPERTY:

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

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