

EXHIBIT E



US007529870B1

(12) **United States Patent**
Schubert et al.

(10) **Patent No.:** **US 7,529,870 B1**
 (45) **Date of Patent:** **May 5, 2009**

(54) **COMMUNICATION BETWEEN AN ACCESSORY AND A MEDIA PLAYER WITH MULTIPLE LINGOES**

(75) Inventors: **Emily C. Schubert**, Los Altos, CA (US); **Wang Chun Leung**, Mountain View, CA (US); **Gregory T. Lydon**, Santa Cruz, CA (US); **Scott Krueger**, San Francisco, CA (US); **Paul Holden**, Sunnyvale, CA (US); **John Archibald**, San Francisco, CA (US); **Lawrence G. Bolton**, Fremont, CA (US); **Donald J. Novotney**, San Jose, CA (US); **John B. Filson**, San Jose, CA (US); **David Tupman**, San Francisco, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 203 days.

(21) Appl. No.: **11/479,057**

(22) Filed: **Jun. 30, 2006**

Related U.S. Application Data

(63) Continuation-in-part of application No. 10/833,689, filed on Apr. 27, 2004, now Pat. No. 7,441,062.

(60) Provisional application No. 60/784,306, filed on Mar. 20, 2006.

(51) **Int. Cl.**
G06F 13/42 (2006.01)

(52) **U.S. Cl.** **710/105**; 710/300; 710/62; 710/64; 710/315

(58) **Field of Classification Search** 710/313–315, 710/300–306, 62–74, 72–73, 110, 8–12
 See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,673,861	A	6/1987	Dubovsky et al.
4,850,899	A	7/1989	Maynard
5,041,025	A	8/1991	Haitmanek
5,055,069	A	10/1991	Townsend et al.
5,080,603	A	1/1992	Mouissie
5,104,243	A	4/1992	Harding
5,108,313	A	4/1992	Adams

(Continued)

FOREIGN PATENT DOCUMENTS

EP	1104150	5/2001
----	---------	--------

(Continued)

OTHER PUBLICATIONS

Altec Lansing, "inMotion Users Guide," Corp. Headquarters, 535 Rte.6 & 209, Milford, PA 18337.

(Continued)

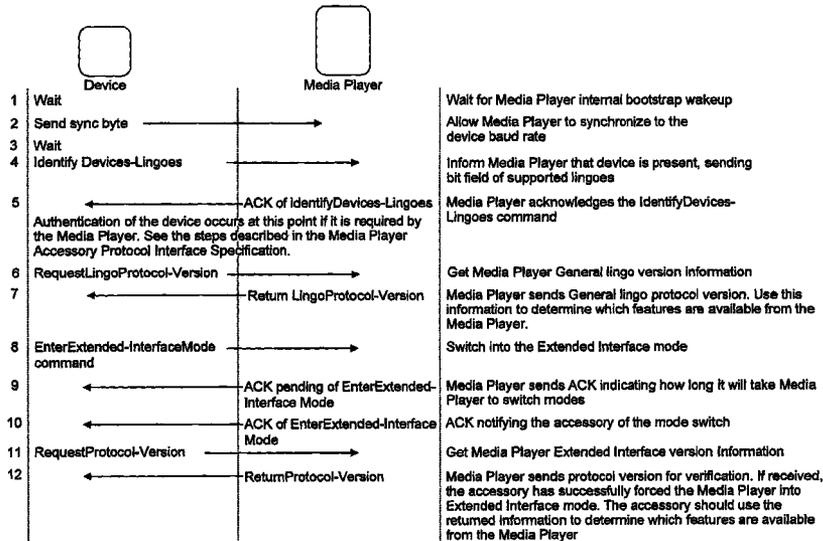
Primary Examiner—Raymond N Phan

(74) *Attorney, Agent, or Firm*—Townsend and Townsend and Crew LLP

(57) **ABSTRACT**

A connector interface system is disclosed. The connector interface system includes an interface and a protocol in communication with which allows a media player to communicate with external devices over a transport link. The protocol includes a core protocol functionality and a plurality of accessory lingoes. The accessory lingoes comprise a microphone lingo, a simple remote lingo, a display remote lingo, a RF transmitter lingo and an extended interface lingo.

40 Claims, 14 Drawing Sheets



US 7,529,870 B1

Page 2

U.S. PATENT DOCUMENTS					
5,150,031 A	9/1992	James et al.	6,614,232 B1	9/2003	Mukai
5,186,646 A	2/1993	Pederson	6,616,473 B2	9/2003	Kamata et al.
5,277,624 A	1/1994	Champion	6,629,197 B1	9/2003	Bhogal et al.
5,471,128 A	11/1995	Patino et al.	6,642,629 B2	11/2003	DeLeeuw
5,586,893 A	12/1996	Mosquera	6,653,813 B2	11/2003	Khatri
5,592,588 A	1/1997	Reekes et al.	6,663,420 B1	12/2003	Xiao
5,618,045 A	4/1997	Kagan et al.	6,665,803 B2 *	12/2003	Lunsford et al. 713/320
5,648,712 A	7/1997	Hahn	6,674,995 B1	1/2004	Meyers et al.
5,660,558 A	8/1997	Osanai et al.	6,725,061 B1	4/2004	Hutchison, IV et al.
5,727,866 A	3/1998	Kraines et al.	6,728,546 B1	4/2004	Peterson et al.
5,754,027 A	5/1998	Oglesbee et al.	6,728,729 B1	4/2004	Jawa et al.
5,830,001 A	11/1998	Kinoshita	6,747,859 B2	6/2004	Walbeck et al.
5,845,217 A	12/1998	Lindell et al.	6,754,468 B1	6/2004	Sieben et al.
5,901,049 A	5/1999	Schmidt et al.	6,774,939 B1	8/2004	Peng
5,964,847 A	10/1999	Booth et al.	6,776,626 B2	8/2004	Huang et al.
5,975,957 A	11/1999	Noda et al.	6,776,660 B1	8/2004	Kubota et al.
6,007,372 A	12/1999	Wood	6,776,665 B2	8/2004	Huang
6,012,105 A	1/2000	Rubbmark et al.	6,799,226 B1	9/2004	Robbin et al.
6,031,797 A	2/2000	Van Ryzint et al.	6,813,528 B1	11/2004	Yang
6,053,773 A	4/2000	Wu	6,816,376 B2	11/2004	Bright et al.
6,078,402 A *	6/2000	Fischer et al. 358/1.16	6,830,160 B2	12/2004	Risolio
6,125,455 A	9/2000	Yeo	6,859,538 B1	2/2005	Voltz
6,130,518 A	10/2000	Gabehart et al.	6,859,854 B2	2/2005	Kwong
6,154,773 A	11/2000	Roberts et al.	6,931,456 B2	8/2005	Payne et al.
6,154,798 A	11/2000	Lin et al.	6,939,177 B2	9/2005	Kato et al.
6,161,027 A *	12/2000	Poirel 455/572	6,991,483 B1	1/2006	Milan et al.
6,169,387 B1	1/2001	Kaib	7,004,787 B2	2/2006	Milan
6,178,514 B1	1/2001	Wood	7,050,783 B2	5/2006	Curtiss et al.
6,184,652 B1	2/2001	Yang	7,054,888 B2	5/2006	La Chapelle et al.
6,184,655 B1	2/2001	Malackowski	7,062,261 B2 *	6/2006	Goldstein et al. 455/419
6,203,345 B1	3/2001	Roque et al.	7,108,560 B1	9/2006	Chou et al.
6,204,637 B1	3/2001	Rengan	7,127,678 B2	10/2006	Bhesania et al.
6,206,480 B1	3/2001	Thompson	7,127,879 B2	10/2006	Zhu et al.
6,211,581 B1	4/2001	Farrant	7,167,112 B2	1/2007	Andersen et al.
6,211,649 B1	4/2001	Matsuda	7,167,935 B2 *	1/2007	Hellberg 710/58
6,224,420 B1	5/2001	Nishio et al.	7,187,947 B1	3/2007	White et al.
6,230,205 B1	5/2001	Garrity et al.	7,215,042 B2 *	5/2007	Yan 307/116
6,234,827 B1	5/2001	Nishio et al.	7,281,214 B2	10/2007	Fadell
6,252,380 B1	6/2001	Koenck	7,293,122 B1	11/2007	Schubert et al.
6,261,109 B1	7/2001	Liu et al.	7,293,227 B2	11/2007	Plastina et al.
6,267,623 B1	7/2001	Hisamatsu	7,299,304 B2	11/2007	Saint-Hilaire et al.
6,271,605 B1	8/2001	Carkner et al.	7,303,282 B2 *	12/2007	Dwyer et al. 353/15
6,280,251 B1	8/2001	Nishio et al.	7,305,506 B1	12/2007	Lydon et al.
6,283,789 B1	9/2001	Tsai	2001/0003205 A1	6/2001	Gilbert
6,304,764 B1	10/2001	Pan	2001/0005641 A1	6/2001	Matsumoto et al.
6,314,479 B1	11/2001	Frederick et al.	2001/0006884 A1	7/2001	Matsumoto
6,316,916 B2	11/2001	Bohne	2002/0002035 A1	1/2002	Sim et al.
6,322,396 B1	11/2001	Kwan	2002/0010759 A1	1/2002	Hitson et al.
6,344,727 B1	2/2002	Desai et al.	2002/0029303 A1	3/2002	Nguyen
6,353,894 B1	3/2002	Pione	2002/0065074 A1	5/2002	Cohn et al.
6,354,713 B1	3/2002	Leifer et al.	2002/0068610 A1	6/2002	Anvekar et al.
6,358,089 B1	3/2002	Kuroda et al.	2002/0072390 A1	6/2002	Uchiyama
6,385,596 B1	5/2002	Wiser et al.	2002/0103008 A1	8/2002	Rahn et al.
6,394,905 B1	5/2002	Takeda et al.	2002/0105861 A1	8/2002	Leapman
6,431,915 B1	8/2002	Ko	2002/0115480 A1	8/2002	Huang
6,453,371 B1	9/2002	Hampson et al.	2002/0132651 A1	9/2002	Jinnouchi
6,454,592 B2	9/2002	Takagi	2002/0151327 A1	10/2002	Levitt
6,461,173 B1	10/2002	Mizuno et al.	2002/0152874 A1	10/2002	Vilcauskas et al.
6,468,110 B2	10/2002	Fujino et al.	2002/0156546 A1	10/2002	Ramaswamy
6,478,603 B1	11/2002	Wu	2002/0156949 A1	10/2002	Kubo et al.
6,483,428 B1	11/2002	Fish et al.	2002/0173273 A1	11/2002	Spurgat et al.
6,485,328 B1	11/2002	Wu	2002/0174269 A1	11/2002	Spurgat et al.
6,489,751 B2	12/2002	Small et al.	2003/0004934 A1	1/2003	Qian
6,505,160 B1	1/2003	Levy et al.	2003/0011608 A1	1/2003	Wada
6,524,119 B2	2/2003	Kato et al.	2003/0028664 A1	2/2003	Tan et al.
6,558,201 B1	5/2003	Begley et al.	2003/0041206 A1	2/2003	Dickie
6,577,877 B1	6/2003	Charlier et al.	2003/0059022 A1	3/2003	Nebiker et al.
6,589,076 B1	7/2003	Davis et al.	2003/0067741 A1	4/2003	Alfonso et al.
6,591,085 B1	7/2003	Grady	2003/0073432 A1	4/2003	Meade
6,608,264 B1	8/2003	Fouladpour	2003/0079038 A1	4/2003	Robbin et al.
6,608,399 B2	8/2003	McConnell et al.	2003/0090998 A1	5/2003	Lee et al.
			2003/0097379 A1	5/2003	Ireton
			2003/0110403 A1	6/2003	Crutchfield et al.

US 7,529,870 B1

Page 3

2003/0151621	A1	8/2003	McEivilly et al.	JP	2002 245719	8/2002
2003/0172209	A1	9/2003	Liu et al.	JP	2002-252566	9/2002
2003/0185395	A1	10/2003	Lee et al.	JP	3090747 U	10/2002
2003/0198015	A1	10/2003	Vogt	JP	2002-374447	12/2002
2003/0220988	A1*	11/2003	Hymel 709/220	JP	2003-17165	1/2003
2003/0236075	A1	12/2003	Johnson et al.	JP	2003-032351	1/2003
2003/0237043	A1	12/2003	Novak et al.	JP	2003-274386	9/2003
2004/0003300	A1	1/2004	Malueg et al.	JP	2002-342659	11/2003
2004/0019497	A1	1/2004	Volk et al.	WO	WO 99/26330	5/1999
2004/0039860	A1	2/2004	Mills et al.	WO	WO 00/39907	7/2000
2004/0048569	A1	3/2004	Kawamura	WO	WO 00/60450	10/2000
2004/0090998	A1	5/2004	Chen	WO	WO 02/49314	6/2002
2004/0103223	A1*	5/2004	Gabehart et al. 710/2	WO	WO 03/036541 A1	5/2003
2004/0116005	A1	6/2004	Choi	WO	WO 03/036957 A1	5/2003
2004/0162029	A1	8/2004	Grady	WO	WO 03/073688	9/2003
2004/0186935	A1	9/2004	Bel et al.	WO	WO 2004/084413	9/2004
2004/0194154	A1	9/2004	Meadors et al.	WO	WO 2004/112311 A	12/2004
2004/0198436	A1	10/2004	Alden	WO	WO 2005/119463 A	12/2005
2004/0224638	A1	11/2004	Fadell et al.	WO	WO 2006/080957 A1	8/2006
2004/0235339	A1	11/2004	Sato et al.			
2004/0249994	A1	12/2004	Shapiro et al.			
2004/0252966	A1	12/2004	Holloway et al.			
2004/0267812	A1	12/2004	Harris et al.			
2004/0267825	A1	12/2004	Novak et al.			
2005/0014119	A1	1/2005	Rudakov			
2005/0014531	A1	1/2005	Findikli			
2005/0014536	A1	1/2005	Grady			
2005/0015355	A1	1/2005	Heller et al.			
2005/0022212	A1	1/2005	Bowen			
2005/0149213	A1	7/2005	Guzak et al.			
2005/0181756	A1	8/2005	Lin			
2005/0207726	A1	9/2005	Chen			
2005/0239333	A1	10/2005	Watanabe et al.			
2005/0240705	A1	10/2005	Novotney et al.			
2005/0281185	A1	12/2005	Kawasaki			
2006/0031545	A1	2/2006	Manders et al.			
2006/0088228	A1	4/2006	Marriott et al.			
2006/0156415	A1	7/2006	Rubinstein et al.			
2006/0184456	A1	8/2006	de Janasz			
2006/0258289	A1	11/2006	Dua			
2006/0294209	A1	12/2006	Rosenbloom et al.			
2007/0056012	A1	3/2007	Kwon et al.			
2007/0056013	A1	3/2007	Duncan			
2007/0070856	A1	3/2007	Tebele			
2007/0080823	A1	4/2007	Fu et al.			
2007/0083814	A1	4/2007	Wilbrink et al.			
2007/0086724	A1	4/2007	Grady et al.			
2007/0173197	A1	7/2007	Hsiung			
2007/0173294	A1	7/2007	Hsiung			
2007/0206827	A1	9/2007	Tupman et al.			
2007/0226384	A1	9/2007	Robbin et al.			

OTHER PUBLICATIONS

Anonymous: "Introduction to Digital Audio," Microsoft Windows Hardware Developer Central Archive, [Online] Dec. 4, 2001 URL:<http://www.microsoft.com/whdc/archive/digitaudio.mspx> [retrieved on Jan. 15, 2008].

Anonymous: "Future of Digital Music in Windows," Microsoft Windows Hardware Developer Central Archive, [Online] Dec. 4, 2001 URL:<http://www.microsoft.com/whdc/archive/digitaudio.mspx> [retrieved on Jan. 15, 2008].

"A Serial Bus on Speed Diagram: Getting Connected with FireWire," downloaded Oct. 16, 2001, PC Magazine: PC Tech (A Serial Bus on Speed) wysiwyg://51<http://www.zdnet.com/pctech/content/18/10/tu1810.007.html>.

Belkin iPod Voice Recorder, Product Specification Sheet, printed Jun. 16, 2004.

Bindra, "Standard Turns Monitor into I/O Hub," Electronic Engineering Times, vol. 918, Sep. 6, 1996, p. 14.

Brentrup, "Introduction to Public Key Cryptography Demystified," Campus Technology, printed from <http://www.campus-technology.com/article.asp?id=7626> on Oct. 6, 2004.

"Cables to Go," download Oct. 16, 2001 <http://www.cablestogo.com/product.asp?cat%5Fid=601&sku=27028>.

Crawford et al., "Sample rate conversion and bit rate reduction in the studio," IEE Colloquium on Digital Audio Signal Processing, May 22, 1991, pp. 8/1-8/3.

Derman, "Monitors Make Net Connections," Electronic Engineering Times, vol. 933, 1996, pp. 60 and 69.

"ExpressBus™ F5U010," User Guide Packing Checklist, Belkin Components Product Warranty.

"FireWire," downloaded Oct. 16, 2001, [si_wyg://_4_2/http://developer.apple.com/hardware/Fire_Wire](http://developer.apple.com/hardware/Fire_Wire).

"Fire Wire Connector", downloaded Oct. 16, 2001, [wysiwyg://176/http://developer.apple.com/...es/Macintosh_CPUsG3/ibook/ibook-27.html](http://developer.apple.com/...es/Macintosh_CPUsG3/ibook/ibook-27.html).

Fried, "FireWire poised to become ubiquitous," downloaded Oct. 16, 2001, CNET News.com, 1394 Trade Association: Press, [wysiwyg://132/http://_113_94ta.org/Press/200_1_Press/august/8.2_7_.b.html](http://www.cnet.com/news/0-I006-200-6021210.html).

Fried, "New Fire Wire to blaze faster trail", downloaded Oct. 16, 2001, CNET News.com, <http://news.cnet.com/news/0-I006-200-6021210.html>.

"How to Connect Your Computer, PC Hardware", downloaded Oct. 16, 2001, <http://www.scar.utoronto.ca/~ccweb/faculty/connect-howto.html>.

"IEEE 1394/USB Comparison", downloaded Oct. 16, 2001, www.genitech.com.au/LIBRARY/TechSupport/infobits/firewirevsusb.htm.

"Introduction to Public Key Cryptography," Oct. 9, 1998, printed from <http://developer.netscape.com/docs/manuals/security/pkin/contents.html> on Oct. 6, 2004.

iPod Classic User's Guide, acquired from apple.com, 2002; 44 pages. iPod nano Features Guide, acquired from apple.com, 2008; 72 pages.

FOREIGN PATENT DOCUMENTS

EP	1498899	A	1/2005
EP	1594319	A	11/2005
EP	1672613	A	6/2006
GB	2405718	A	3/2005
JP	07-176351		7/1995
JP	10-321302		4/1998
JP	10-334993		12/1998
JP	11-288420		10/1999
JP	2000-214953		8/2000
JP	2000-223215		8/2000
JP	2000-223216		8/2000
JP	2000-223218		8/2000
JP	2001-035603		2/2001
JP	2001-069165	A	3/2001
JP	2001-196133		7/2001
JP	2001-230021		8/2001
JP	2001-332350		11/2001
JP	2002-025720		1/2002
JP	2002-203641		7/2002

US 7,529,870 B1

Page 4

- iPod touch User's Guide, acquired from apple.com, 2008, 120 pages.
- "iPodDock/iPod Cradle," www.bookendzdocks.com/bookendz/dock_cradle.html, downloaded Feb. 27, 2003.
- Lambert, "Digital Audio Interfaces," *Journal of the Audio Engineering Society, Audio Engineering Society*, New York, NY, vol. 38, No. 9, (Sep. 1, 1990), pp. 681-684, 686, 68 XP000175146 ISSN: 1549-4950 figures 9, 10.
- Lewis, "On Technology." *Fortune Magazine*, Dec. 9, 2002.
- "Making UBS Work," downloaded Oct. 16, 2001, *PC Magazine: PC Tech* [wysiwyg:/155/http://www.zdnet.com/pcmag/ptech/content!18/04/tu1804.001.html](http://www.zdnet.com/pcmag/ptech/content!18/04/tu1804.001.html).
- Menezes et al., "Handbook of Applied Cryptography," Identification and Entity Authentication, pp. 385-424.
- Microsoft, "Media Transport Protocol Implementation Details," 2005, 18 pages.
- "MPV™ Music Profile Specification Revision 1.00" Internet Citation [online] (Jan. 7, 2004) URL:<http://www.osta.org/mpv/public/specs/MPVMusic-Prof-Spec-1.00.pdf> [retrieved Jun. 20, 2006] the whole document.
- Networking Tech Note, "1394 Standards and Specifications," 3 pgs.
- "Neuros MP3 Digital Audio Computer," www.neurosaudio.com, downloaded Apr. 9, 2003.
- "PMC FW2 IEEE 1394 FireWire Controller," downloaded Oct. 16, 2001, <http://www.bvmltd.co.uk/PMCFw2ds.html>.
- Severance, "FireWire Finally Comes Home", Michigan State University, Standards, Nov. 1998, pp. 117-118.
- Sinitsyn, "Synchronization Framework For Personal Mobile Servers," *Pervasive Computing and Communications Workshops (PERCOMW'04)*, Proceedings of the Second IEEE Annual Conference, Piscataway, NJ, USA, IEEE, Mar. 14, 2004, pp. 208-212.
- Teener, "Understanding Fire Wire: The IEEE 1394 Standards and Specifications", downloaded Oct. 16, 2001, [wysiwyg:119/http://www.chipcenter.com/networking/ieee_1394/main.html](http://www.chipcenter.com/networking/ieee_1394/main.html).
- "The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition", Published by Standards Information Network, IEEE Press.
- "Universal serial bus specification—rev 2.0," XP002474828, Chapter 9, USB Device Framework, pp. 239-274.
- Vitaliano, "Why FireWire is Hot!Hot!Hot!" downloaded Oct. 16, 2001, "Impact.FireWire.SideBar" <http://www.vxm.com/21R.35.html>.
- Whittle, "Public Key Authentication Framework: Tutorial," First Principles Consulting, Jun. 2, 1996.
- Anonymous; "Windows and Red Book Audio," Microsoft Windows Hardware Developer Central Archive, [Online] Dec. 4, 2001 URL:<http://www.microsoft.com/whdc/archive/Dmfuture.msp> [retrieved Jan. 15, 2008].

* cited by examiner

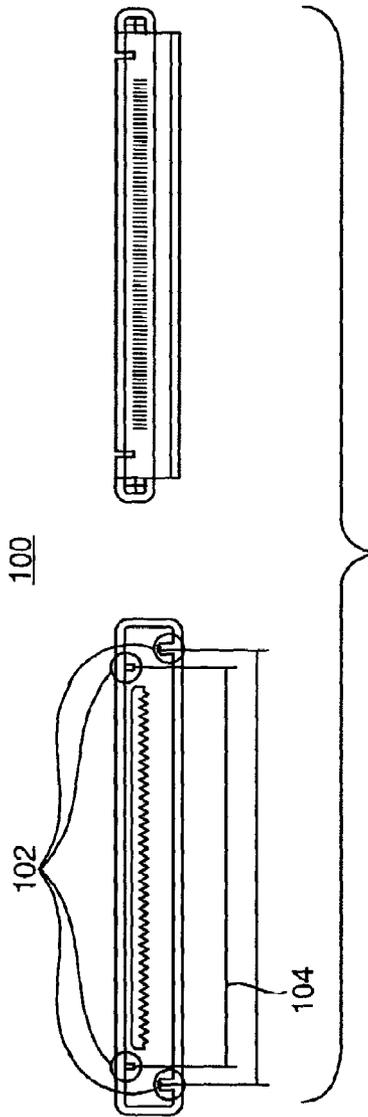


FIG. 1A

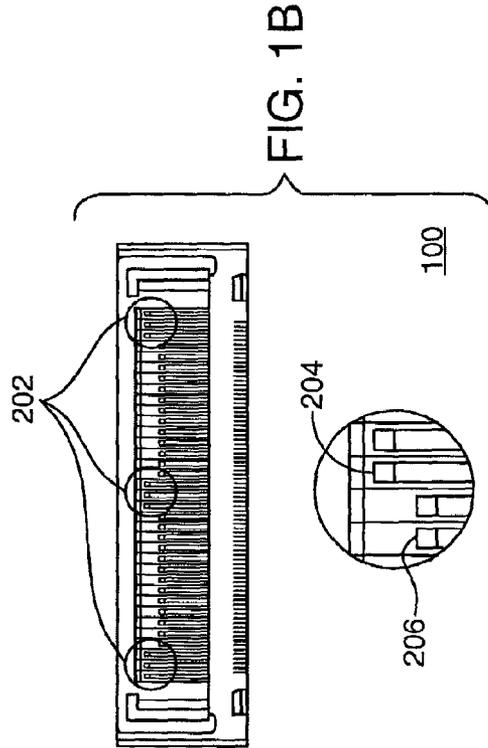


FIG. 1B

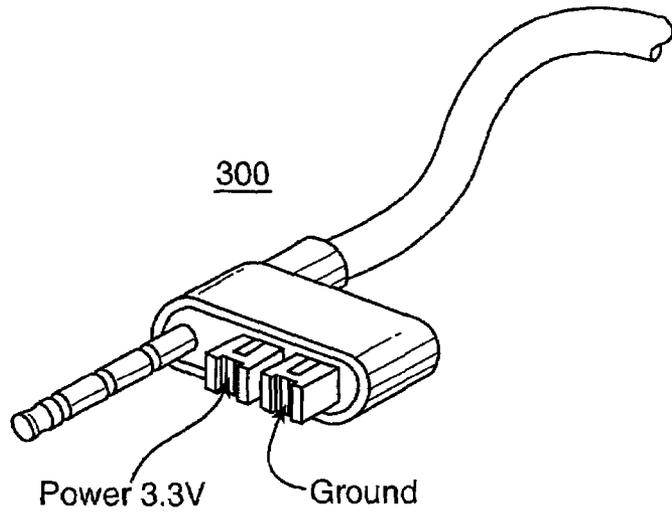
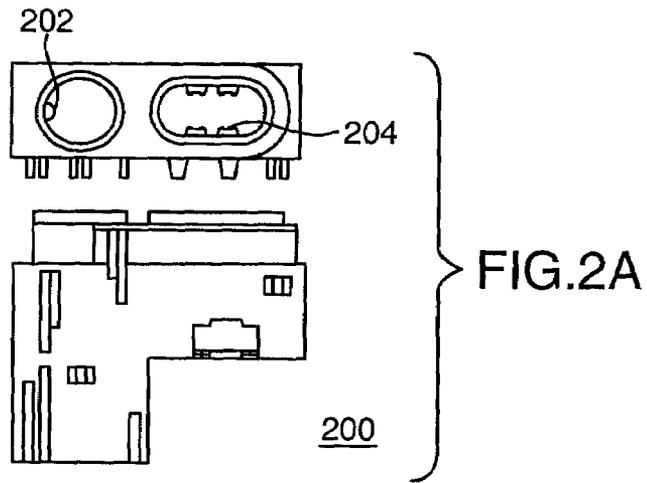


FIG. 2B

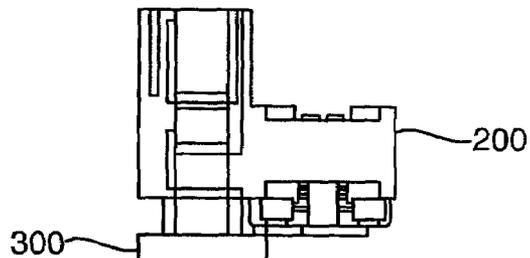


FIG. 2C

Pin	Signal Name	I/O	Function
1	DGND	I	Digital Ground
2	DGND	I	Digital 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. NOT for powering; only to detect USB host
9	TPB-	I/O	Firewire signal
10	Accessory Identify	I	Pull down in dock 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 output from iPod. Current limited to 100mA.
14	Reserved		
15	DGND	GND	Digital ground in iPod
16	DGND	GND	Digital ground in iPod
17	Reserved		
18	Dock Tx	I	Serial protocol (Data to iPod)
19	Dock Rx	O	Serial protocol (Data to iPod)
20	Accessory Detect	I/O	
21	S Video Y	O	Luminance Component
22	S Video C	O	Chrominance Component
23	Video Composite	O	Composite Signal
24	Remote Sense	I	Detect Remote
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 iPod
31	Chassis		Chassis ground for connector shell
32	Chassis		Chassis ground for connector shell

FIG. 3A

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	HP Detect	I	Internal Switch to detect 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	Video Signal
6	Accessory 3.3 V	O	3.3V Accessory power 100mA max
7	Tx	O	Serial protocol (Data from iPod to Device)
8	Rx	I	Serial protocol (Data to iPod from Device)
9	D GND	GND	Digital ground for accessory

FIG. 3B

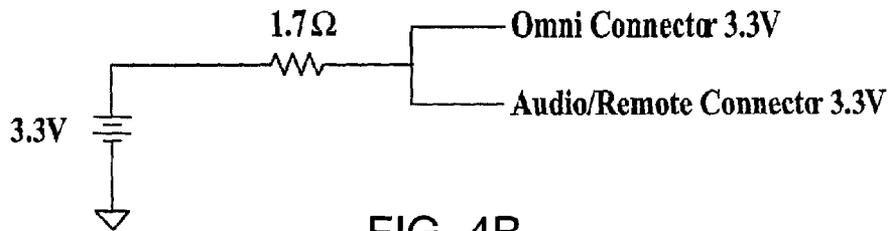


FIG. 4B

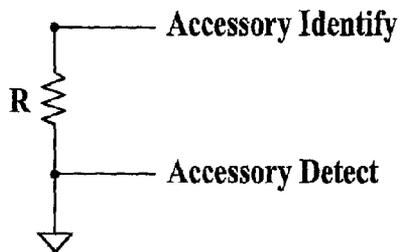


FIG. 4C

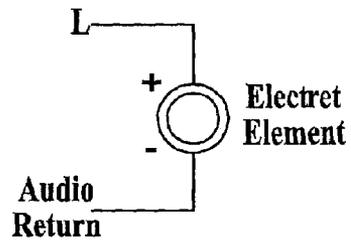


FIG. 4D

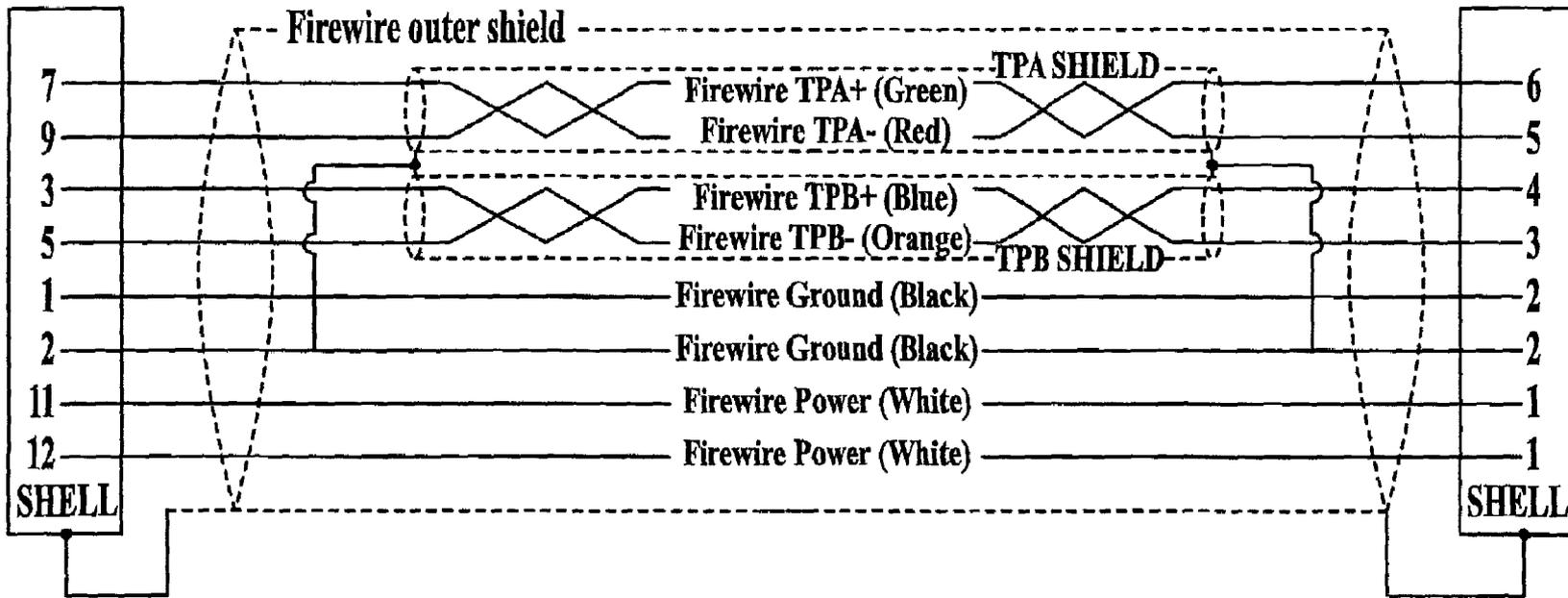


FIG. 4A

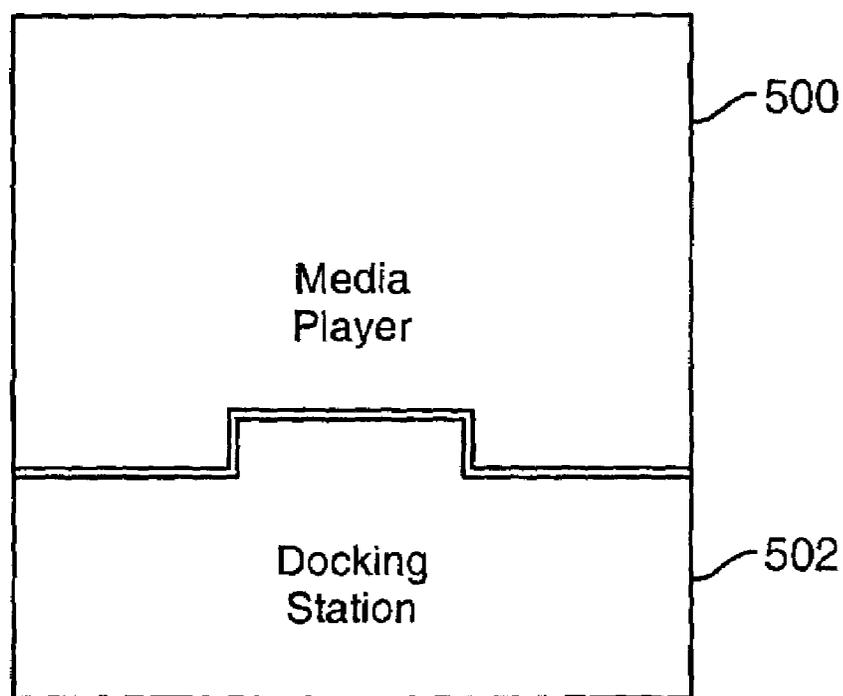


FIG. 5A

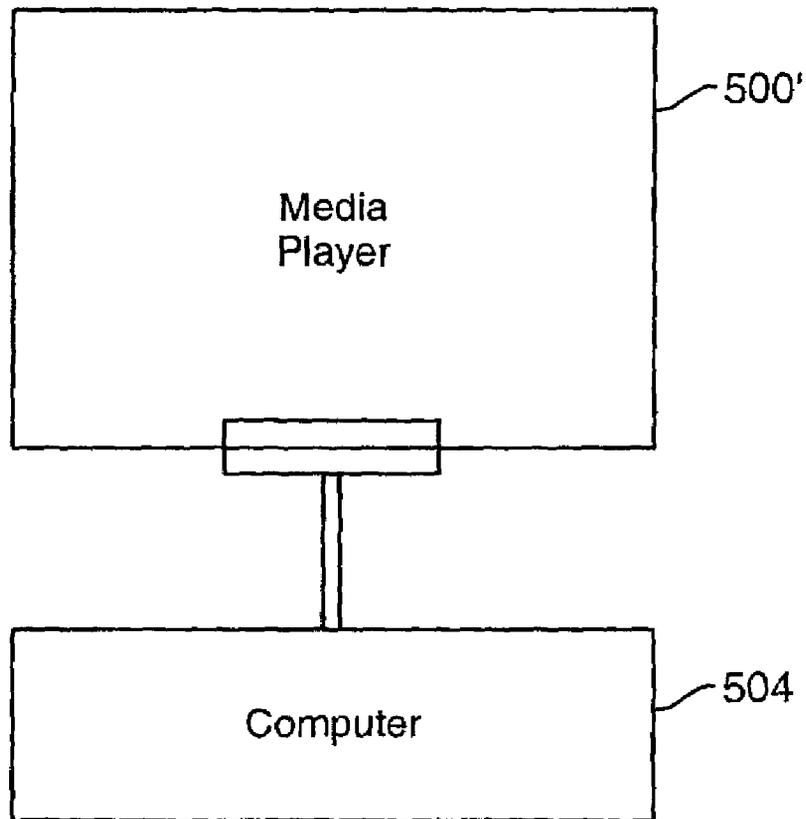


FIG. 5B

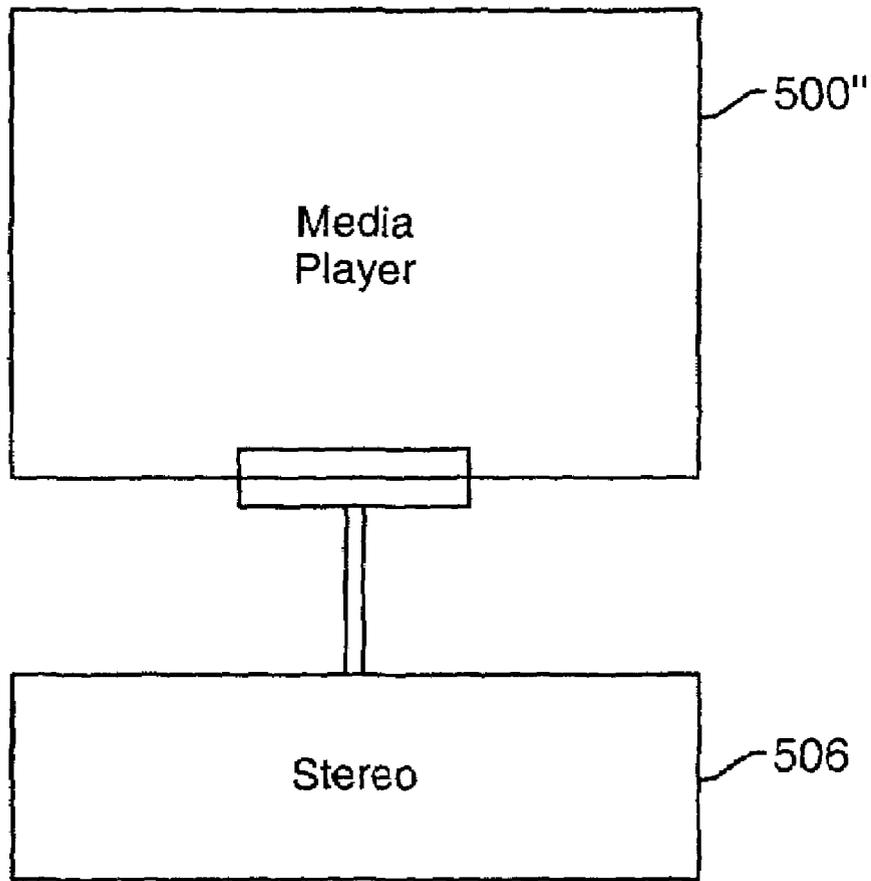


FIG. 5C

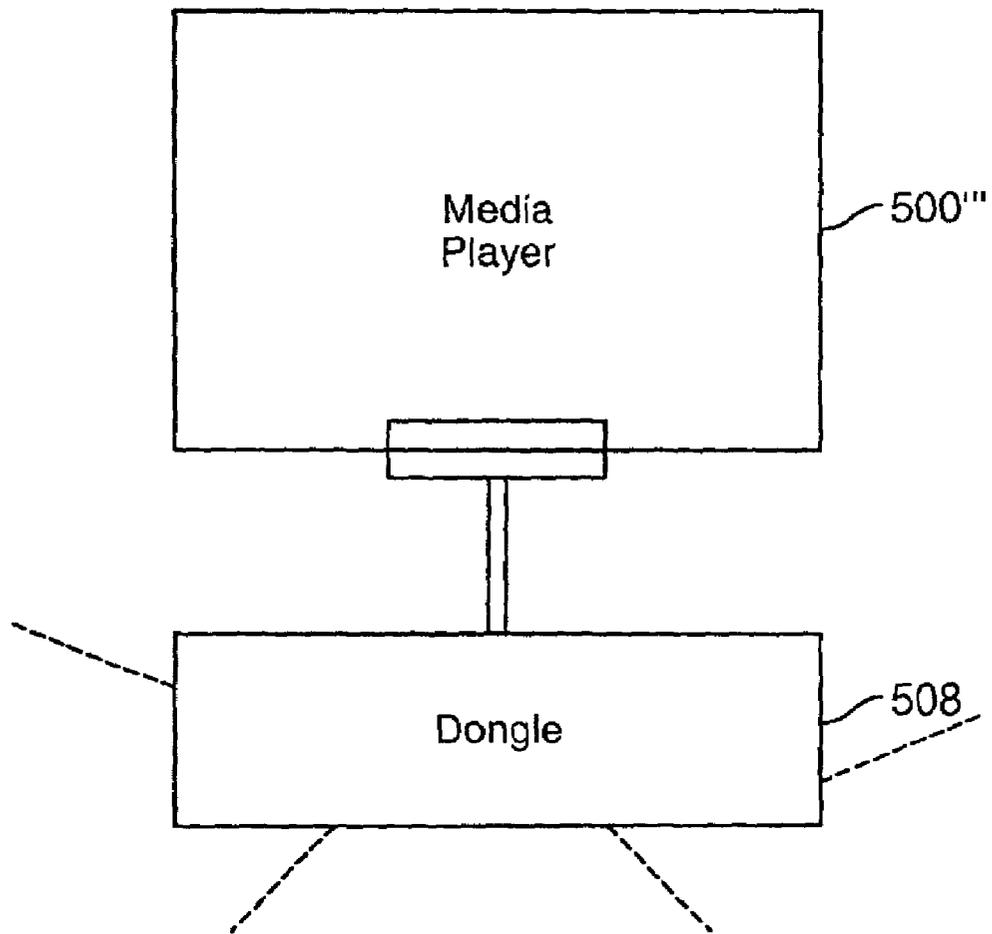


FIG. 5D

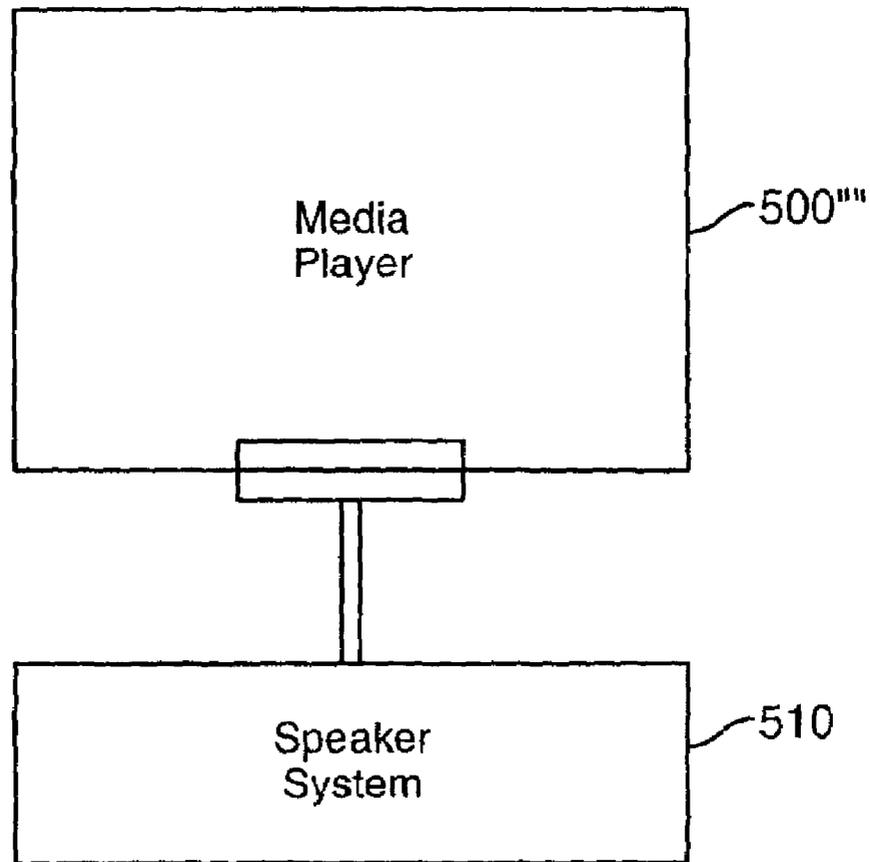


FIG. 5E

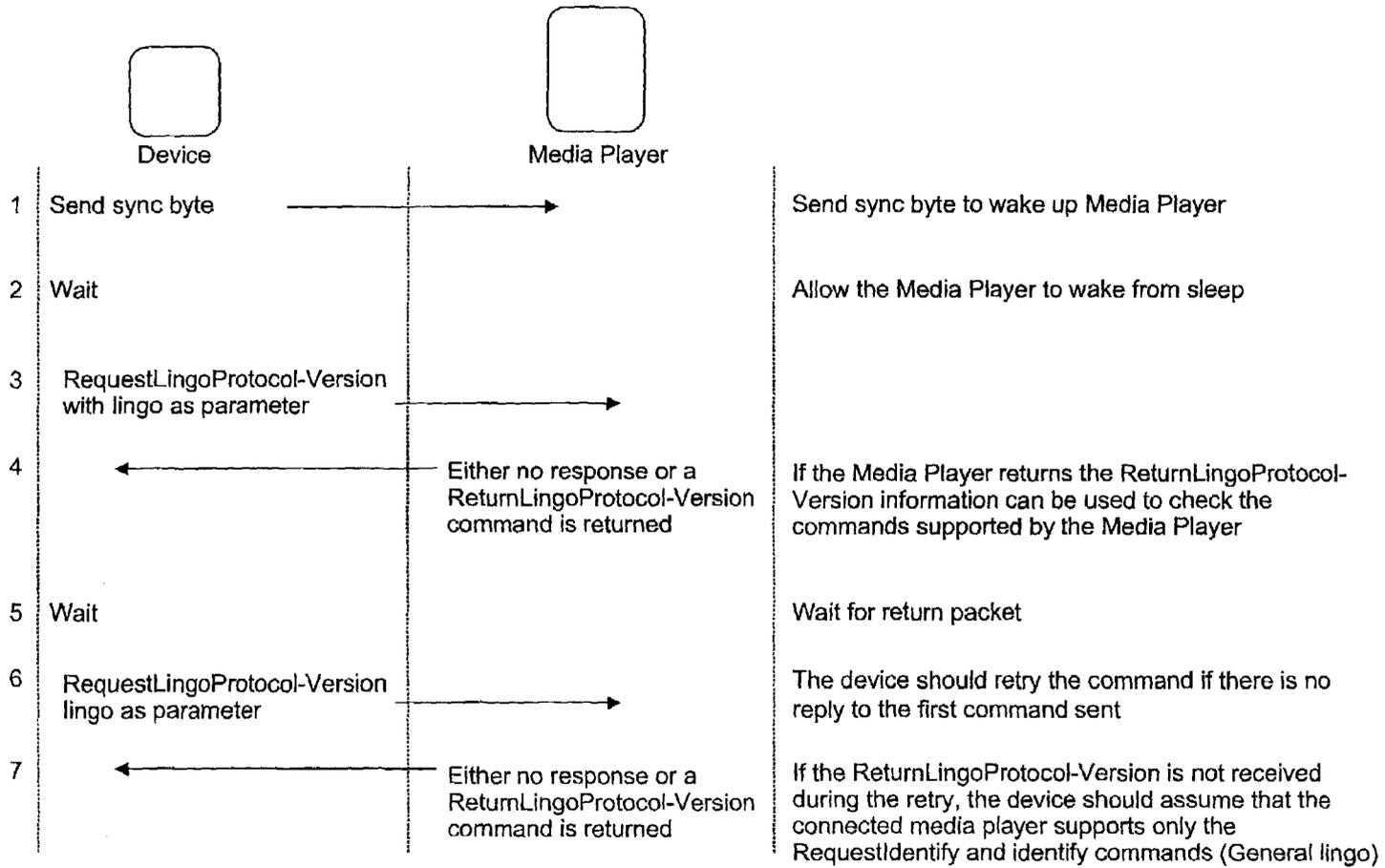


FIG. 6

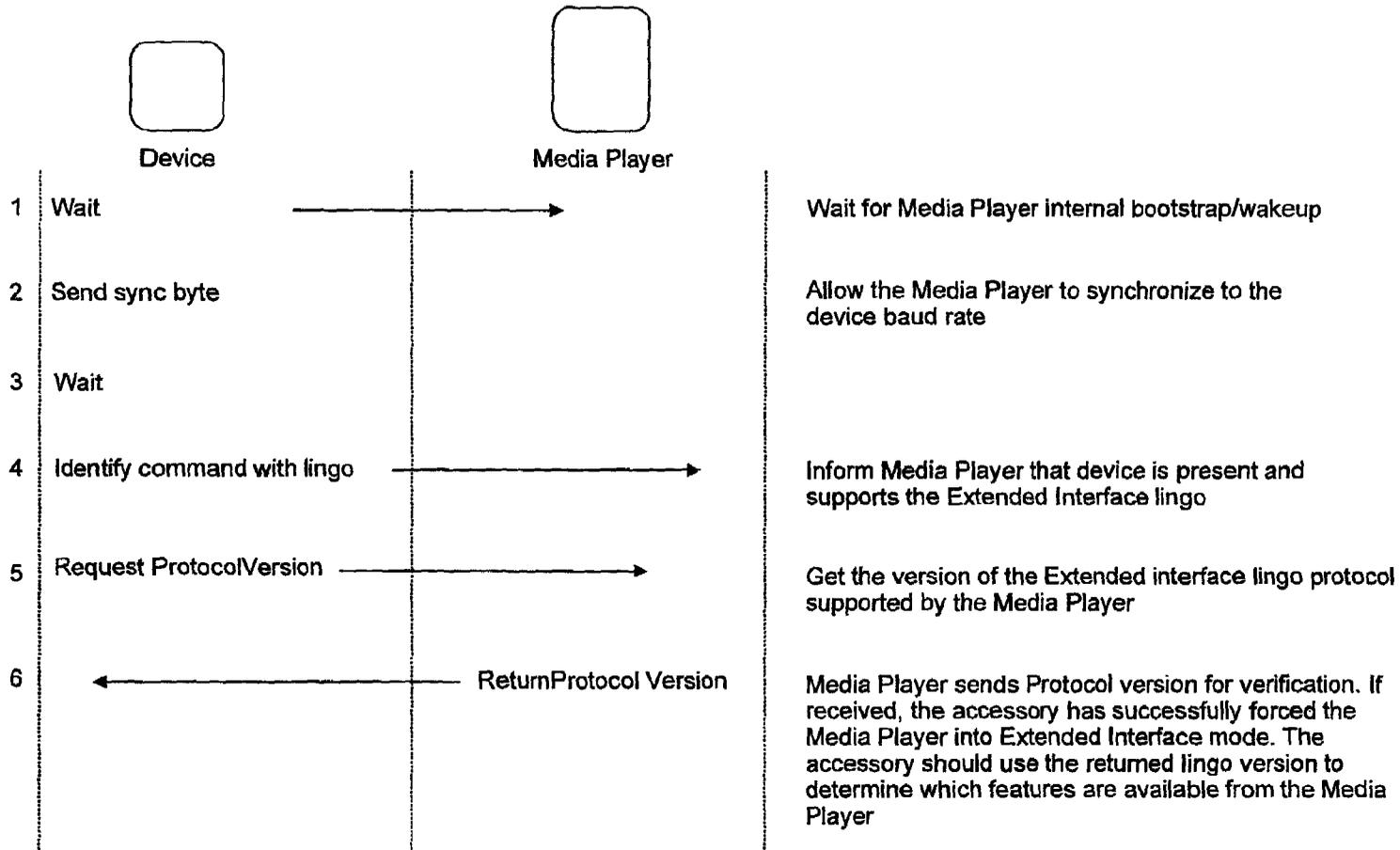


FIG. 7

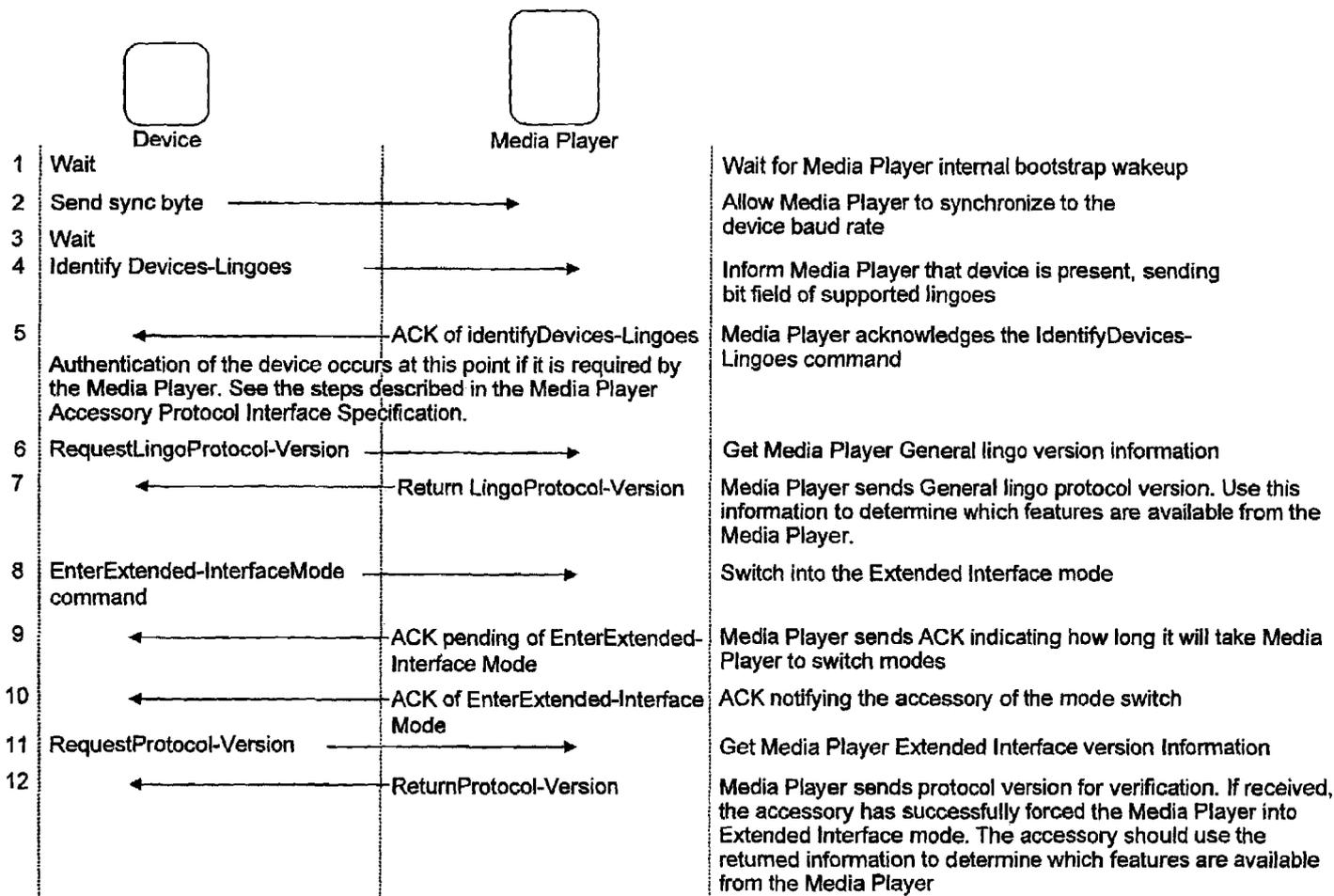


FIG. 8

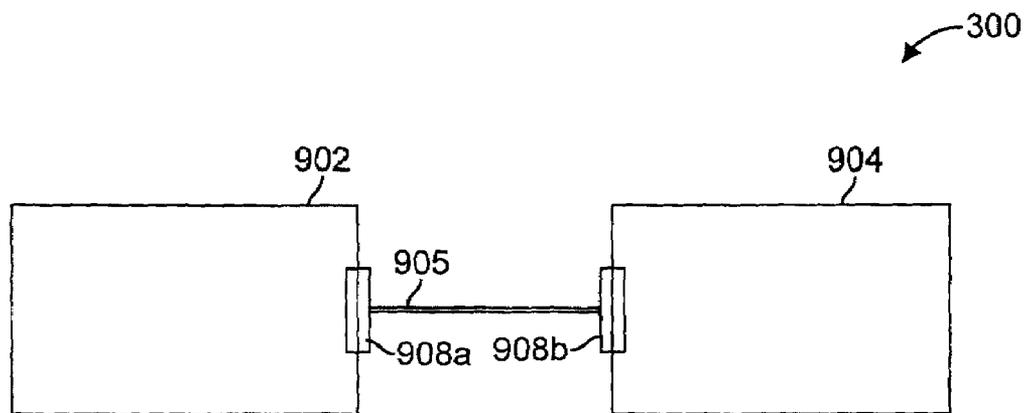


FIG. 9

US 7,529,870 B1

1

**COMMUNICATION BETWEEN AN
ACCESSORY AND A MEDIA PLAYER WITH
MULTIPLE LINGOES**

CROSS-REFERENCE TO RELATED
APPLICATIONS

This application is a continuation-in-part of U.S. application Ser. No. 10/833,689, filed Apr. 27, 2004, entitled "Connector Interface System from a Multi-Communication Device".

This application claims the benefit of: U.S. Provisional Patent Application No. 60/784,306, filed Mar. 20, 2006, entitled "Connector Interface System" which is hereby incorporated by reference.

FIELD OF THE INVENTION

The present invention relates generally to connector interfaces and more particularly to a connector interface system which is utilized in conjunction with media players and their accessories.

BACKGROUND OF THE INVENTION

A media player stores media assets, such as audio tracks or photos, that can be played or displayed on the media player. One example of a media player is the iPod® media player, which is available from Apple Computer, Inc., of Cupertino, Calif. Often, a media player acquires its media assets from a host computer that serves to enable a user to manage media assets. As an example, the host computer can execute a media management application to manage media assets. One example of a media management application is iTunes®, version 6.0, produced by Apple Computer, Inc.

A media player typically includes one or more connectors or ports that can be used to interface to the media player. For example, the connector or port can enable the media player to couple to a host computer, be inserted into a docking system, or receive an accessory device. There are today many different types of accessory devices that can interconnect to the media player. For example, a remote control can be connected to the connector or port to allow the user to remotely control the media player. As another example, an automobile can include a connector and the media player can be inserted onto the connector such that an automobile media system can interact with the media player, thereby allowing the media content on the media player to be played within the automobile.

Currently, the connectors or ports of a media player are open for use so long as a compatible connector or port is utilized. Consequently, numerous third-parties have developed accessory devices for use with other manufacturers' media players.

In a typical connector interface, there is a docking connector that allows for the docking of the media player device to a docking station for another type of communication for the device. A media player also typically includes a remote connector with the ability to output audio. As more multi-media content becomes available (i.e., digital, video graphics, etc.) it is desirable to have a media player which can effectively input and output such data.

Finally, a media player must be able to identify a particular device's functionality to which it is associated. Heretofore, there is no device that includes features that overcome many of the above-stated problems. What is desired is a connector

2

interface system which is utilized in such a device to address all the above-identified issues. The present invention addresses such a need.

SUMMARY OF THE INVENTION

A connector interface system is disclosed. The connector interface system includes an interface and a protocol in communication with the interface which allows a media player to communicate with external devices over a transport link. The protocol includes a core protocol functionality and a plurality of accessory lingo. The accessory lingo comprise a microphone lingo, a simple remote lingo, a display remote lingo, a RF transmitter lingo and an extended interface lingo.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGS. 1A and 1B illustrate a docking connector in accordance with the present invention.

FIGS. 2A-2C illustrate the remote connector in accordance with the present invention.

FIG. 3A illustrates the connector pin designations for the docking connector.

FIG. 3B illustrates the connection pin designations for the remote connector.

FIG. 4A illustrates the Firewire connector interface.

FIG. 4B illustrates the USB connector interface.

FIG. 4C illustrates a reference schematic diagram for accessory detect and identify system for detecting and identifying accessories for the docking connector.

FIG. 4D is a reference schematic of a electric microphone that is within the remote connector.

FIGS. 5A-5E illustrate a media player coupled to a plurality of accessories.

FIG. 6 illustrates the sequence of events that can be utilized to test for the full set of general lingo commands.

FIG. 7 illustrates a single lingo accessory command flow.

FIG. 8 illustrates a multiple lingo command flow.

FIG. 9 illustrates an exemplary system using the connector interface system described herein.

DETAILED DESCRIPTION

Definitions

Device

An external electronic component connected to the media player using the 30-pin connector or the audio/remote connector.

HID

Human Interface Device. HID is a standard USB class of functionality and interface. 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 data to the HID interface of the media player or from the media player to the host. MPAP packets are broken into HID reports before being sent across the transport link and are reassembled on the receiving side.

MPUI

Media player USB Interface. This is a configuration of the media player when attached as a device over USB. This

US 7,529,870 B1

3

configuration allows the media player to be controlled using MPAP, using a USB Human Interface Device (HID) interface as a transport mechanism.

Lingo

The command category used by a device. There is a general lingo that must be supported by all devices. Other lingoes are designed for use by specific devices.

Link

The logical connection between an external device and the media player via serial port or other physical connection.

The present invention relates generally to media players and more particularly to a connector interface system for such devices. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein.

Connector Interface System Overview

To describe the features of the connector interface system in accordance with the present invention in more detail, refer now to the following description in conjunction with the accompanying drawings.

Docking Connector

FIGS. 1A and 1B illustrate a docking connector **100** in accordance with the present invention. Referring first to FIG. 1A, the keying features **102** are of a custom length **104**. In addition, a specific key arrangement is used where one set keys are separated by one length are at the bottom and another set of keys are separated by another length at the top of the connector. The use of this key arrangement prevents noncompliant connectors from being plugged in and potentially causing damage to the device. The connector for power utilizes a Firewire specification for power. The connector includes a first make/last break contact to implement this scheme. FIG. 1B illustrates the first make/last break contact **202** and also illustrates a ground pin and a power pin related to providing an appropriate first make/last break contact. In this example, the ground pin **204** is longer than the power pin **206**. Therefore, the ground pin **204** would contact its mating pin in the docking accessory before the power pin **206**. Therefore internal electrical damage of the electronics of the device is minimized.

In addition, a connector interface system in accordance with the present invention uses both USB and Firewire interfaces as part of the same docking connector alignment, thereby making the design more compatible with different types of interfaces, as will be discussed in detail hereinafter. In so doing, more remote systems and devices can interface with the multi-communication device.

Remote Connector

The connection interface system also includes a remote connector which provides for the ability to output audio and, input audio, provides I/O serial protocol, and an output video. FIG. 2A is a front and top view of a remote connector **200** in accordance with the present invention. As is seen, the remote connector **200** includes a top headphone receptacle **202**, as well as a second receptacle **204** for remote devices. FIG. 2B illustrates a plug **300** to be utilized in the remote connector. The plug **300** allows the features to be provided via the remote connector. FIG. 2C illustrates the plug **300** inserted into the

4

remote connector **200**. Heretofore, all these features have not been implemented in a remote connector. Therefore, a standard headphone cable can be plugged in but also special remote control cables, microphone cables, or video cables could be utilized with the remote connector.

To describe the features of the connector interface system in more detail, please find below a functional description of the docking connector, remote connector and a protocol in accordance with the present invention.

Docking and Remote Connector Specifications

For an example of the connector pin designations for both the docking connector and the remote connector for a multi-communication device such as a media player device by Apple Computer, Inc., refer now to FIGS. 3A and 3B. FIG. 3A illustrates the connector pin designations for the docking connector. FIG. 3B illustrates the connection pin designations for the remote connector.

Docking Connector Specifications

FIG. 4A illustrates a typical FireWire connector interface for the docking connector.

FireWire Power has the following characteristics: 8V-30V DC IN, 10 W Max. FireWire communication is designed to IEEE 1394 A Spec (400 Mb/s).

USB Interface

The media player provides two configurations, or modes, of USB device operation: mass storage and Media Player USB Interface (MPUI). The MPUI allows the media player to be controlled using a Media Player Accessory Protocol (MPAP) which will be described in detail herein, using a USB Human Interface Device (HID) interface as a transport mechanism.

Accessory 3.3 V Power

FIG. 4B illustrates the USB connector interface. The media player accessory power pin supplies voltages, for example, 3.0 V to 3.3V+/-5% (2.85 V to 3.465 V) over the 30-pin connector. A maximum current is shared between the 30-pin and Audio/Remote connectors.

By default, the media player supplies a particular current such as 5 mA. Proper software accessory detect is required to turn on high power (for example, up to 100 mA) during active device usage. When devices are inactive, they must consume less than 5 mA current.

Accessory power is switched off for a period of, for example, approximately 2 seconds during the media player bootstrap process. This is done to ensure that accessories are in a known state and can be properly detected. All accessories are responsible for re-identifying themselves after the media player completes the bootstrap process and transitions accessory power from the off to the on state.

Accessory power is grounded through the DGND pins.

FIG. 4C illustrates a reference schematic diagram for accessory detect and identify system for detecting and identifying accessories for the docking connector. The system comprises a resistor to ground that allows the device to determine what has been plugged into docking connector. There is an internal pullup on Accessory Identify. Two pins are required (Accessory Identify & Accessory Detect)

FIG. 4D is a reference schematic of a electric microphone that is within the remote connector.

Serial Protocol Communication is provided using two pins to communicate to and from device (Rx & Tx). Input and Output levels are 0V=Low, 3.3V=High.

US 7,529,870 B1

5

Media Player Accessory Protocol (MPAP)

Media players connect to a variety of accessories. FIGS. 5A-5E illustrates a media player 500 coupled to different accessories. FIG. 5A illustrates a media player 500 coupled to a docking station 502. FIG. 5B illustrates the media player 500' coupled to a computer 504. FIG. 5C illustrates the media player 500" coupled to a car or home stereo system 506. FIG. 5D illustrates the media player 500''' coupled to a dongle 508 that communicates wirelessly with other devices. FIG. 5E illustrates the media player 500'''' coupled to a speaker system 510. As is seen, what is meant by accessories includes but is not limited to docking stations, chargers, car stereos, microphones, home stereos, computers, speakers, and devices which communicate wirelessly with other devices.

Accordingly a single media player must be able to communicate with different devices having varying functionality. To allow for this communication to be handled efficiently in a system and method in accordance with the present invention, a protocol is provided. As part of the connector interface system and in conjunction with the connectors, the protocol allows for the media player to identify the type of accessory that the media player is connected to and also allows the media player to identify the functionality of the accessory. In so doing the media player can efficiently and effectively interact with the accessory in the appropriate manner.

The Media Player Accessory Protocol (MPAP) allows the media player to communicate with a functional range of external devices. The protocol can be broken into three logical components: the protocol transport link, the protocol core, and the individual accessory lingoes. Devices can use USB and UART serial interfaces as channels to transfer MPAP packets. The protocol core describes the components necessary for communication with all external devices. It includes the basic packet definition and the general lingo, which allows for accessory identification, authentication, and retrieval of media player information. The accessory lingoes comprise the individual dialect commands. Each accessory lingo corresponds to a functional class of external devices.

Every external device must support a protocol transport link, the protocol core, and one or more lingoes, as required for its function. For example, the media player standard in-line remote control is a UART serial device that uses the general lingo and the simple remote lingo. To describe these features in more detail refer now to the following.

Protocol Transport Links

Accessories may communicate with the media player, using MPAP over the serial port link or the USB port link. Those links are described below.

UART Serial Port Link

Accessories using the Media Player Accessory Protocol (MPAP) over the UART serial port link use two pins, RX and TX, to communicate to and from the media player.

A device coupled to the docking connector allows for a standard serial protocol to be utilized. Attaching a serial dock accessory makes any top-attached (remote connector) accessories inactive.

The MPAP builds upon an existing serial specification such as the RS-232 serial specification. However, the signaling levels are non-standard. The RS-232 specification states that a mark is $-7V$ and a space is $+7V$. In MPAP protocol, for example, a mark may be 2.85 V through 3.465 V and a space is 0 V through 0.9 V.

USB Port Link

The media player, for example, may be a USB 2.0-compliant device that supports two mutually exclusive modes of operation:

6

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 synching music and content, transferring files, and so forth.

MPAP enabled device. This is the configuration needed to support MPAP using the Media Player USB Interface (MPUI). This mode must be selected by the USB host before it can be used.

These two mutually exclusive modes of operation are each represented by a USB configuration. When the media player 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.

Accessory Identify Resistor and MPUI

There are two types of media player accessories: serial accessories that communicate with the media player using MPAP and resistor-based accessories that need access to specific media player behaviors.

Resistor-based accessories use an Accessory Identify resistor (RID) to get access to a specific media player behavior. These devices tend to be simple accessories, such as battery packs and car chargers, and have one specific purpose. When attached, these accessories unlock media player features based on the RID used. Accessories that communicate with the media player using MPAP are serial accessories. Serial-based accessories may use the other R_{ID} values if they are to be utilized for unlocking a particular resistor-based behavior.

The presence of a R_{ID} of one resistance value on USB attachment triggers the media player to present the MPUI configuration as the first, or default, configuration. The second configuration becomes media player as a Mass Storage class disk device.

For instance, connecting a media player to a standard host, such as a PC or a Macintosh, along with the R_{ID} causes the media player to be set up as a HID device (the HID interface being a component of the MPUI configuration) and not as a disk device. With no resistor, the media player is set up as a disk device.

It is possible to create an accessory that supports both MPAP over USB and the older UART serial-only media players using the same connector.

Media Player USB Interface (MPUI) Configuration

The MPUI configuration allows the media player to communicate using MPAP over USB. The USB Human Interface Device (HID) interface is the transport link and uses two endpoints for communication: the control endpoint is used for OUT data, while the HID interrupt endpoint is used for IN data.

The media player HID interface utilizes several vendor-specific HID reports, some of which are used to transport data from the host and some of which are used to transport data to the host. The HID report sizes range from a few bytes to several hundred KB and each one has a unique identifier. In order to send data to the media player, a host chooses one or more appropriately-sized HID reports in which to embed the MPAP packet and sends this to the media player HID interface with USB Set-Report command. The media player reassembles the MPAP packet and processes it. The process is repeated in reverse when the media player sends responses or MPAP packets to the host.

HID as a Transport

As mentioned earlier, the HID interface breaks MPAP 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,

US 7,529,870 B1

7

where a set of reports encompasses one or more complete MPAP packets. For instance, a set could be a single HID report containing one MPAP packet or a set of HID reports containing a total of 3 MPAP packets.

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 media player specifies several different report types. The USB host should analyze the HID report descriptor of the media player at runtime to determine which Report ID corresponds to the most appropriate report type for each transfer.

A representative group of lingoos for the media player are listed below.

Lingo Specifications

Lingo

General

Microphone

Simple Remote

Display Remote

Extended Interface

RF transmitter

The general lingo is shared for housekeeping commands across all devices. In addition to the general lingo, an accessory will implement a function specific lingo. The microphone lingo is used by the remote connector on the multi-communication device. The simple remote lingo is used by a standard in-line remote control. The display remote lingo is reserved for a device with similar functionality to the standard remote but with a display for status. The RF transmitter lingo is used for devices that transmit the device analog audio over radio frequencies. To describe the lingoos in more detail, refer now to the following discussion in conjunction with the accompanying figures.

General Lingo Command Summary

Table 1 provides a summary of all commands in the general lingo, including whether or not device authentication is required for use of the command.

TABLE 1

Command	Authentication Required
Request Identify	No
RequestMediaPlayerName	No
RequestMediaPlayerSoftwareVersion	No
RequestMediaPlayerSerialNum	No
RequestMediaPlayerModelNum	No
RequestLingoProtocolVersion	No
Identify	No
IdentifyAccessoryLingoos	No
ACK	No
AckDevAuthenticationInfo	No
AckDevAuthenticationStatus	No
AckMediaPlayerAuthenticationInfo	No
AckMediaPlayerAuthenticationStatus	N/A
ReturnMediaPlayerName	No
ReturnMediaPlayerSoftwareVersion	No
ReturnMediaPlayerSerialNumber	No
ReturnLingoProtocolVersion	No
RetDevAuthenticationInfo	No
RetDevAuthenticationSignature	No
RetMediaPlayerAuthenticationInfo	No
RetMediaPlayerAuthenticationSignature	No
GetDevAuthenticationInfo	No
GetDevAuthenticationSignature	No
GetMediaPlayerAuthenticationInfo	No
GetMediaPlayerAuthenticationSignature	No
NotifyMediaPlayerStateChange	No

8

To identify itself, an accessory should send either an Identify command or an IdentifyAccessoryLingoos command. Accessory devices that support more than one lingo, not including the general lingo, should use the IdentifyAccessoryLingoos command.

The media player may send a RequestIdentify command to the device to ask to re-identify itself.

The remaining general lingo commands can be used to obtain general information from the media player. These commands allow the device to request the name, serial number, model number, and software version number of the media player. The RequestLingoProtocolVersion command allows a device to query the media player for the lingo protocol versions of all supported lingoos on the media player. The ACK command is used by the media player 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.

An accessory may send a RequestLingoProtocolVersion command, to determine which features the connected media player supports. If the media player does not respond to the RequestLingoProtocolVersion command, the device should try the command once more. If no response is received the second time, the device should assume that the connected media player supports the only RequestIdentify and Identify General Lingo commands.

FIG. 6 shows the sequence of the events that can be used to test for the full set of general lingo commands. As is seen, first a sync byte is sent to wake up the media player, then allow the media player allowed to wake from sleep. If the media player returns the ReturnLingoProtocolVersion command, then the general lingo version information can be used to check the commands supported by the media player. Next, wait for return packet and the device should retry the command if there is no reply to the first command send. If the ReturnLingoProtocolVersion command is not received during the retry, the device should assume that the connected media player supports only the RequestIdentify and Identify commands. This assumes the accessory device is connected and has already been through the identification process.

General Lingo Command Details

This section describes the general lingo commands and their packet formats.

Request Identify Command

Direction: media player→accessory

The media player sends this command to prompt accessories to re-identify themselves. If an accessory receives this command, it should respond with either the IdentifyAccessoryLingoos command or the Identify command. Accessories that require authentication or use the USB transport link, or support multiple lingoos should respond using the IdentifyAccessoryLingoos command.

Identify Command

Direction: accessory→media player

The accessory must send this command to notify the media player that an accessory has been attached and to register the lingo it supports. Accessories should identify at boot time and any time they receive a RequestIdentify command from the media player.

The Identify command has facilities for RF Transmitter devices to draw more than a predetermined amount of power (such as 4 mA) from the media player.

ACK Command

Direction: media player→accessory

The media player sends the ACK command to notify the device of command completion status and errors.

US 7,529,870 B1

9

RequestMediaPlayerSoftwareVersion Command
 Direction: media player→accessory
 Retrieves the software version for the media player. The media player responds with a ReturnMediaPlayerSoftwareVersion command containing the major, minor, and revision version numbers.

ReturnMediaPlayerSoftwareVersion Command
 Direction: media player→accessory
 The media player sends this command in response to the RequestMediaPlayerSoftwareVersion command message from the accessory. The media player returns each version number as an individual byte, with the major version number sent first.

RequestMediaPlayerName Command
 Direction: accessory→media player
 Retrieves the name of the media player. The media player responds with a ReturnMediaPlayerName command containing the name of the media player.

ReturnMediaPlayerName Command
 Direction: media player→accessory
 The media player sends this command in response to the RequestMediaPlayerName command message from the accessory.

RequestMediaPlayerSerialNum Command
 Direction: accessory→media player
 Retrieves the serial number string of the media player. The media player responds with a ReturnMediaPlayerSerialNum command containing the serial number.

ReturnMediaPlayerSerialNum Command
 Direction: media player→accessory
 The media player sends this command in response to the RequestMediaPlayerSerialNumber command message from the accessory.

RequestMediaPlayerModelNum Command
 Direction: accessory→media player
 Retrieves information for the media player. The media player responds with a ReturnMediaPlayerModelNum command containing the model number of the media player.

ReturnMediaPlayerModelNum Command
 Direction: media player→accessory
 The media player sends this command in response to the RequestMediaPlayerModelNum command message from the accessory.

IdentifyAccessoryLingoes Command
 Direction: accessory→media player
 The accessory sends this command to signal its presence and to identify its supported lingoes. In response, the media player sends an ACK command. The IdentifyAccessoryLingoes command is used by multi-lingo accessories to report all supported lingoes and should be used in place of the Identify command.

GetDevAuthenticationInfo Command
 Direction: media player→accessory
 The media player sends this command to obtain authentication information from the accessory. The command is sent if and only if the accessory has indicated that it supports authentication in its IdentifyAccessoryLingoes Options bits and has passed a valid, non-zero accessory ID. In response, the accessory sends a RetDevAuthenticationInfo command.

RetDevAuthenticationInfo Command
 Direction: accessory→media player
 The accessory indicates the MPAP authentication version that it supports by returning this command in response to a GetDevAuthenticationInfo command from the media player.

AckDevAuthenticationInfo Command
 Direction: media player→accessory

10

The media player sends this command in response to RetDevAuthenticationInfo command. It indicates the current state of the accessory authentication information.

RequestLingoProtocolVersion Command
 Direction: accessory→media player
 Retrieves version information for any of the lingoes supported by the media player. The media player responds with a ReturnLingoProtocolVersion command containing the major and minor version information of the requested media player lingo.

ReturnLingoProtocolVersion Command
 Direction: media player→accessory
 The media player sends this command in response to the RequestLingoProtocolVersion command message from the accessory. The major and minor version information for the requested lingo are returned.

GetDevAuthenticationSignature Command
 Direction: media player→accessory
 The media player sends this command to authenticate a accessory that has identified itself as requiring authentication. Authentication occurs either immediately upon identification or when the accessory attempts to use a restricted lingo or command. The accessory calculates its digital signature based on the challenge offered by the media player and sends the results back to the media player using a RetDevAuthenticationSignature command.

If the returned signature cannot be verified, the media player responds with a non-zero AckDevAuthenticationStatus command, followed immediately by another GetDevAuthenticationSignature command.

RetDevAuthenticationSignature Command
 Direction: accessory→media player
 The accessory sends this command to the media player in response to GetDevAuthenticationSignature command. The media player verifies the digital signature, calculated by the accessory based on the offered challenge. If verification passes, the media player authenticates the accessory and updates its lingo and command access permissions accordingly. The authentication status is sent to the accessory using AckDevAuthenticationStatus command.

AckDevAuthenticationStatus Command
 Direction: media player→accessory
 The media player sends this command to the accessory in response to the RetDevAuthenticationSignature command. It indicates the current accessory authentication state. If the accessory receives a non-zero status, the accessory has failed authentication and will only be able to use unauthenticated lingo commands.

If the accessory receives a zero status, the media player has successfully authenticated the accessory. The accessory may then use the requested authenticated lingoes and commands. Optionally, the accessory may begin the process of authenticating the media player, by sending a GetMediaPlayerAuthentication command.

GetMediaPlayerAuthenticationSignature Command
 Direction: accessory→media player
 The accessory uses this command to send an offered challenge to the media player for digital signature. In response, the media player returns its signed challenge to the accessory using RetMediaPlayerAuthenticationSignature command. Accessories should implement the authentication retry feature described in GetDevAuthenticationSignature command.

RetMediaPlayerAuthenticationSignature Command
 Direction: media player→accessory
 The media player sends this command to the accessory in response to a GetMediaPlayerAuthenticationSignature command. The accessory verifies the digital signature, calculated

US 7,529,870 B1

11

by the media player based on the offered challenge, and, if verification passes, authenticates the media player. The accessory sends the authentication status to the media player.

AckMediaPlayerAuthenticationStatus Command

Direction: accessory→media player

The accessory sends this command to the media player in response to RetMediaPlayerAuthenticationSignature command. It indicates the current media player authentication state. The accessory should return a non-zero ACK for each failed authentication attempt.

GetMediaPlayerAuthenticationInfo Command

Direction: accessory→media player

The accessory sends this command to obtain authentication information from the media player. The media player should send this command only if the accessory has indicated that it supports authentication in its IdentifyAccessoryLingoes command options bits and the media player has successfully completed the authentication process. In response, the media player sends RetMediaPlayerAuthenticationInfo command.

AckMediaPlayerAuthenticationInfo Command

Direction: accessory→media player

The accessory sends this command to the media player in response to RetMediaPlayerAuthenticationInfo command. It indicates the current state of the media player information version. If the accessory sends a non-zero status, it indicates that it will not be able to authenticate the media player due to a protocol version or authentication index mismatch.

NotifyMediaPlayerStateChange Command

Direction: media player→accessory

The media player sends this notification command when the media player state is about to change to accessories that identify using IdentifyAccessoryLingoes command. If the accessory identifies using Identify command, this notification is not sent. The state change byte indicates the specific media player state transition. If the media player is switching from a power on state to a light sleep state, accessories must immediately reduce their power consumption below the maximum current such as 5 mA. When the media player has transitioned to a deep sleep or hibernate state, self-powered accessories are expected to automatically re-identify themselves when accessory power is restored.

Microphone Lingo Command

The microphone lingo enables combination microphone and speaker accessory devices to record and playback audio. Media player mass storage disk capacities enable the option of supporting a stereo input mode and higher audio sample rates. Media players may therefore be used for high-quality mobile audio recording.

When the media player detects a device speaking the microphone lingo, it may transition into a recorder application where it can create and manage recordings. Based on the microphone device capabilities, the media player recording application may choose to change its appearance based on the presence or absence of certain microphone features. The device should indicate its capabilities to the media player on request. These capabilities may include:

Stereo line input source

Stereo/mono control

Recording level control

Recording level limiter

Microphone accessories can draw power from the media player or supply power to the media player. Accessory device power management is important as media players transition to a smaller physical size at the same time as trying to extend battery life. As an accessory using the microphone lingo, the

12

microphone accessory will be notified of media player state changes, such as transitioning to the power on, light sleep, hibernate, and deep sleep states. Accessory power is in low mode by default and is raised to high power mode only during recording and playback states.

The microphone accessory is responsible for keeping the power consumption below the maximum allowed limits for each media player state. Note that accessory power is completely shut off when the media player enters the hibernate and deep sleep states. When waking from a light sleep state, the microphone accessory is required to re-identify and re-authenticate itself, as with other accessories using authenticated lingoes or commands. On reset or power up, the accessory device should be in low power state (for example, consuming less than 5 mA) with the amplifier off (audio input and output disabled).

Microphone state information is typically retained locally by the accessory player while uninterrupted accessory power (either high or low power) is available. If accessory power is turned off, device state information may be lost. Accessories are not expected to retain state information across accessory power down cycles (hibernate or deep sleep modes).

Media player playback volume level changes may require the accessory to support display remote lingo functionality.

Table 2 lists the commands available as part of the microphone lingo.

TABLE 2

Command	Connector	Authentication Required
BeginRecord	Audio/Remote	No
EndRecord	Audio/Remote	No
BeginPlayback	Audio/Remote	No
EndPlayback	Audio/Remote	No
ACK	30-pin	Yes
GetDevAck	30-pin	Yes
MediaPlayerModeChange	30-pin	Yes
GetDevCaps	30-pin	Yes
RetDevCaps	30-pin	Yes
GetDevCtrl	30-pin	Yes
RetDevCtrl	30-pin	Yes
SetDevCtrl	30-pin	Yes

BeginRecord Command

Direction: media player→accessory

The media player sends this command to notify the accessory that audio recording has started. The accessory does not return a packet to the media player in response to this command.

EndRecord Command

Direction: media player→accessory

The media player sends this command to notify the accessory that audio recording has ended. The accessory does not return a packet to the media player in response to this command.

BeginPlayback Command

Direction: media player→accessory

The media player sends this command to notify the accessory that audio playback has started. The accessory does not return a packet to the media player in response to this command.

EndPlayback Command

Direction: media player→accessory

The media player sends this command to notify the accessory that audio playback has ended. The accessory does not return a packet to the media player in response to this command.

US 7,529,870 B1

13

ACK Command

Direction: accessory→media player

The microphone accessory sends this command in response to a command sent from the media player. Note that some commands do not require an ACK response. The accessory sends an ACK response when a command that does not return any data has completed, a bad parameter is received, or an unsupported or invalid command is received.

GetDevAck Command

Direction: media player→accessory

The media player sends this command to get an ACK response from a microphone accessory. The media player uses this command to “ping” the accessory and determine that it is present and ready to accept commands. In response, the accessory sends the ACK command with command status OK.

MediaPlayerModeChange Command

Direction: media player→accessory

The media player sends this command to the microphone accessory when an audio recording or playback event occurs. The microphone accessory uses the MediaPlayerModeChange command to configure its inputs or outputs and power consumption level for the specified mode. In response, the accessory sends the ACK command with the command status OK. The accessory sends the ACK command when the accessory has completed its mode change.

GetDevCaps Command

Direction: media player→accessory

The media player sends this command to the microphone accessory to determine the features present on the accessory. In response, the accessory sends RetDevCaps command with the payload indicating the capabilities it supports.

RetDevCaps Command

Direction: accessory→media player

The accessory sends this command in response to the command GetDevCaps command sent by the media player. The microphone accessory returns the payload indicating which capabilities it supports.

GetDevCtrl Command

Direction: media player→accessory

The media player sends this command to get the accessory control state for the specified control type. In response, the accessory sends RetDevCtrl command with its current control state. If this command is not supported by the accessory—that is, if the microphone does not have any configurable controls—it should return an ACK command with a bad parameter error status.

RetDevCtrl Command

Direction: accessory→media player

The accessory sends this command in response to the command GetDevCtrl command received from the media player. The accessory returns the current control state for the specified control type. Control types are only supported if the associated capabilities bits are set in the command RetDevCaps command.

SetDevCtrl Command

Direction: media player→accessory

The media player sends this command to set the accessory control state for the specified control type. In response, the accessory sends the ACK command with the command status. If this command is not supported by the accessory—that is, if the microphone does not have any configurable controls—it should return an ACK command with a bad parameter error status.

14

Simple Remote Lingo

A simple remote accessory sends a buttons status command to indicate an updated status of which buttons are held down. The data of the packet is a number of bytes indicating which buttons are currently held down. In one embodiment, the bytes are made up by ORing the masks of the buttons together. The accessory will send a command to indicate all buttons are released. While any buttons are held down the accessory should repeat this packet on a predetermined interval. If no packet of this sort is received by the host for a predetermined time period, the host may assume a packet was lost and go to “all buttons up” mode.

When the user presses and holds down a button, a simple remote accessory should generate the button status packet immediately and repeat it for as long as the button is pressed. If a second button is pressed while the first button is down, the button status packet sent by the accessory should include status for both buttons, and this packet should be repeated for as long as both buttons are held down. Table 3 lists the possible media player button states.

Simple Remote Button Map

TABLE 3

Button	Number
Play/Pause	0
Volume Up	1
Volume Down	2
Next Track	3
Previous Track	4
Next Album	5
Previous Album	6
Stop	7
Play/Resume	8
Pause	9
Mute toggle	10
Next Chapter	11
Previous Chapter	12
Next Playlist	13
Previous Playlist	14
Shuffle setting advance	15
Repeat setting advance	16
Power On	17
Power Off	18
Backlight for 30 seconds	19
Begin FF	20
Begin REW	21
Menu	22
Select	23
Up Arrow	24
Down Arrow	25

Some media player button states are interpreted differently by the media player when pressed and held down. These are as follows:

The Next Track button is treated as a Scan Forward button when pressed and held while a track is playing.

The Previous Track button is treated as a Scan Backward button when pressed and held while a track is playing.

The Play/Pause button is treated as a Power Off button when pressed and held.

The Menu button is treated as a Display Backlight On/Off button when pressed and held.

If the media player is in Browse mode, the Select button is treated as an Add Track to On-The-Go Playlist button when pressed and held.

Repeated Next Track and Previous Track commands (see Table 3) without an intervening button status packet indicating all buttons are up, are interpreted as Fast Forward and Rewind commands. For a locking Fast Forward or Rewind

US 7,529,870 B1

15

button, use the Begin Fast Forward or Begin Rewind commands to start the operation and a Play/Resume command to return to the play state.

The Next and Previous Album commands (see Table 3) have no effect if there is no next or previous album to go in to the Now Playing list.

Below is the command for the simple remote lingo.

ContextButtonStatus Command

Direction: Accessory→media player

The accessory sends this command to the media player when a button event occurs. When all buttons are released, the accessory should send a button status packet with a 0x0 payload to indicate that no buttons are pressed. The media player does not return a packet to the accessory in response to this command.

Display Remote Lingo

The display remote lingo is for accessories that need to control the state of the media player, recreate a portion of the media player UI on a remote display, or control the state of the media player equalizer (EQ). The display remote protocol can be used by simple inline-display remotes (remotes that have single-line display and play control buttons) and more complex accessories that have full multi-line graphical displays to show information about the track, artist, or album; current play or pause state; track position; battery; shuffle and time. By supporting multiple lingoes, an accessory can use the display remote lingo in combination with other lingoes to create a fully functional product. Accessories can also use this lingo to control the state of the media player equalizer. The display remote lingo supports serial accessories attached to the Audio/Remote or 30-pin connector.

The display remote command set uses a single byte command format similar to the general and simple remote lingoes. Accessories using the display remote lingo can identify using the general lingo, with either the identify single lingo or IdentifyAccessoryLingoes multiple lingo commands.

Table 4 illustrates the commands for the display remote lingo.

TABLE 4

Command
ACK
GetCurrentEQProfileIndex
RetCurrentEQProfileIndex
SetCurrentEQProfileIndex
GetNumEQProfiles
RetNumEQProfiles
GetIndexedEQProfileName
RetIndexedEQProfileName
SetRemoteEventNotification
RemoteEventNotification
GetRemoteEventStatus
RetRemoteEventStatus
GetMediaPlayerStateInfo
RetMediaPlayerStateInfo
SetMediaPlayerStateInfo
GetPlayStatus
RetPlayStatus
SetCurrentPlayingTrack
GetIndexedPlayingTrackInfo
RetIndexedPlayingTrackInfo
GetNumPlayingTracks
RetNumPlayingTracks
GetPowerBatteryState
RetPowerBatteryState
GetSoundCheckState
RetSoundCheckState
SetSoundCheckState

16

The commands, their direction and their functions are described in detail hereinbelow.

ACK Command

Direction: media player→accessory

The media player sends this command to acknowledge the receipt of a command from the accessory and return the command status. The command ID field indicates the accessory command for which the response is being sent. The command status indicates the result of the command (success or failure).

GetCurrentEQProfileIndex Command

Direction: accessory→media player

Requests the current equalizer (EQ) profile setting index. In response, the media player sends the RetCurrentEQProfileIndex command packet.

RetCurrentEQProfileIndex Command

Direction: media device→accessory

The media accessory sends this command, returning the current equalizer profile setting index, in response to the GetCurrentEQProfileIndex command packet sent by the accessory.

SetCurrentEQProfileIndex Command

Direction: accessory→media player

Set the current equalizer profile setting index and optionally restores the original equalizer setting on accessory detach. The valid equalizer index range can be determined by sending a GetNumEQProfiles command. In response to this command, the media player returns an ACK packet with the status of this command.

GetNumEQProfiles Command

Direction: accessory→media player

Requests the number of media player equalizer profile settings. In response, the media player sends the RetNumEQProfiles command packet.

RetNumEQProfiles Command

Direction: media player→accessory

Returns the number of equalizer profiles. The media player sends this command in response to the GetNumEQProfiles command packet sent by the accessory.

GetIndexedEQProfileName Command

Direction: accessory→media player

Requests the media player equalizer profile setting name for a given equalizer profile index. In response, the media player sends the RetIndexedEQProfileName command packet. The valid profile index range can be obtained by sending the GetNumEQProfiles command.

RetIndexedEQProfileName Command

Direction: media player→accessory

Returns the media player equalizer profile setting name for the specified equalizer profile index in response to the GetIndexedEQProfileName command.

SetRemoteEventNotification Command

Direction: accessory→media player

Enables asynchronous remote event notification for media player events. On accessory detach, event notification is reset to the default disabled state.

RemoteEventNotification Command

Direction: media player→accessory

The media player sends this command asynchronously whenever an enabled event change has occurred. Use the SetRemoteEventNotification command to control which events are enabled.

GetRemoteEventStatus Command

Direction: accessory→media player

Gets the status of state information that has changed on the media player. In response, the media player sends a RetRemoteEventStatus command, containing a bitmask of

US 7,529,870 B1

17

event states that changed since the last GetRemoteEventStatus command and clears all the remote event status bits. This command may be used to poll the media player for event changes without enabling asynchronous remote event notification.

RetRemoteEventStatus Command

Description: media player→accessory

The media player sends this command in response to the GetRemoteEventStatus command.

GetMediaPlayerStateInfo Command

Direction: accessory→media player

Obtains media player state information. In response, the media player sends a RetMediaPlayerStateInfo command with the requested state information.

RetMediaPlayerStateInfo Command

Direction: media player→accessory

The media player sends this command in response to the GetMediaPlayerStateInfo command.

SetMediaPlayerStateInfo Command

Direction: accessory→media player

Set the media player state. The information type specifies the type of information to update. In response, the media player sends an ACK command with the results of the operation.

GetPlayStatus Command

Direction: accessory→media player

Gets the current media player play status information. In response, the media player sends a RetPlayStatus command with the current play state, track index, track position, and track length.

RetPlayStatus Command

Direction: media player→accessory

The media player sends this command in response to the GetPlayStatus command and returns the current media player play status information.

SetCurrentPlayingTrack Command

Direction: accessory→media player

Sets the currently playing track to the track at the specified index. The total number of playing tracks can be obtained by sending a GetNumPlayingTracks command.

GetIndexedPlayingTrackInfo Command

Direction: accessory→media player

Gets track information for the specified playing track index. The infoType field specifies the type of information to be returned, such as track title, artist title, artist name, album name, track genre, and track chapter information. In response, the media player sends a RetIndexedPlayingTrackInfo command with the requested track information.

RetIndexedPlayingTrackInfo Command

Direction: media player→accessory

The media player sends this command in response to the GetIndexedPlayingTrackInfo command. It returns the requested type of information and data for the specified playing track.

RetNumPlayingTracks Command

Direction: media player→accessory

The media player sends this command in response to the GetNumPlayingTracks command received from the accessory. It returns the total number of tracks queued in the playback engine.

GetPowerBatteryState Command

Direction: accessory→media player

Gets the power and battery level state of the media player. In response, the media player sends a RetPowerBatteryState command with the power and battery information.

18

RetPowerBatteryState Command

Direction: media player→accessory

The media player sends this command in response to the GetPowerBatteryState command and returns the current media player power state and battery level.

GetSoundCheckState Command

Direction: accessory→media player

Gets the current sound check setting. When enabled, sound check adjusts track playback volume to the same level. In response, the media player sends a RetSoundCheckState command with the current sound check state.

RetSoundCheckState Command

Direction: media player→accessory

The media player sends this command in response to the GetSoundCheckState command and returns the current state of the sound check setting.

SetSoundCheckState Command

Direction: accessory→media player

Sets the state of the sound check setting and optionally saves the previous sound check state to be restored on accessory detach. In response to this command, the media player sends an ACK packet with the status of the command.

Extended Interface Lingo Specification

25 Operating Modes

The media player can be considered to operate in two major modes, standard UI mode and extended Interface mode.

Standard UI Mode

This is the standard user interface mode and allows the media player to be driven by its front panel display and buttons.

Extended Interface Mode

The media player transitions into the Extended Interface mode when either of the following occurs:

(1)—A single-lingo accessory is connected to the media player and issues the Identify command, specifying the Extended Interface lingo as its supported lingo.

(2)—A multi-lingo accessory is connected to the media player and issues a General Lingo Command: EnterRemoteUI Mode command.

If the media player is playing a track during this transition, the playback is automatically paused.

The extended interface protocol allows accessories to replace the graphic with a downloaded image set through a SetDisplayImage command. Removing power from the media player while a connection remains results in the media player going into a light sleep state after a predetermined amount of inactivity. The controls of the media player are disabled when in extended interface mode.

The media player transitions back to standard UI mode when any of the following occurs:

(1)—The accessory is disconnected from the media player.

(2)—A multi-lingo accessory issues the General Lingo Command: ExitRemoteUI Mode.

The accessory re-identifies itself, using either of the general lingo commands Identify or IdentifyAccessoryLingo.

If the media player is playing a track during this transition, the playback is automatically paused. Any media player settings with the restore on exit feature state are restored when the media player is disconnected.

Light Sleep State

The media player screen, playback, and most major parts of the media player are off while the media player is in the light sleep state. The media player transitions from extended

US 7,529,870 B1

19

interface mode to the light sleep state when power is detached and playback is idle. A period of inactivity is required before the media player transitions into the light sleep state. When power is restored, the media player returns to the extended interface mode.

A media player will not sleep while it remains attached to an active USB host. The USB host must switch off its host controller to force a media player in extended interface mode into the light sleep state. If the media player is not currently in extended interface mode, playback must be paused before the host controller is turned off. In extended interface mode, there is no need to pause media playback before turning off the host controller because this action generates a disconnect event that causes the media player to exit the extended interface mode and allows the media player to transition into the light sleep state. Attaching USB power to a media player in light sleep state will wake it up.

Switching Between Standard UI and Extended Interface Modes

There are four general lingo commands that allow accessories to determine what mode the media player is in and to switch between the two major modes, Standard UI and Extended Interface. These commands were implemented to allow a device to switch between modes without having to unplug the accessory. Multi-lingo accessories must use these commands to switch into and out of the extended interface mode.

Table 5 lists the general lingo command codes for querying, entering, and exiting the extended interface protocol.

TABLE 5

General Lingo command	Requires authentication UART serial port link	Requires authentication USB port link
RequestRemoteUIMode	No	Yes
ReturnRemoteUIMode	No	Yes
EnterRemoteUIMode	No	Yes
ExitRemoteUIMode	No	Yes

RequestRemoteUIMode Command

Direction: accessory→media player

Requests the Extended Interface mode from the media player. The media player responds with a ReturnRemoteUIMode command.

ReturnRemoteUIMode Command

Direction: media player→accessory

Returns the current operating mode of the media player UI

EnterRemoteUIMode Command

Direction: accessory→media player

The accessory sends this command to the media player to force it to enter the Extended Interface mode. If the media player is already in the extended interface mode, it immediately returns a general lingo ACK command packet, notifying the user that the command was successful.

ExitRemoteUIMode Command

Direction: accessory→media player

The accessory sends this command to the media player to force it to exit the Extended Interface mode. If the media player is already in the standard UI mode, it immediately returns a general lingo ACK command packet, notifying the user that the command was successful.

Using the Extended Interface Protocol

In order to effectively use the extended interface protocol, there are two logical entities that need to be managed while browsing and playing content: the content database engine

20

and the playback engine. The following describes those engines and gives an example of command traffic between an extended interface accessory and a media player.

The Playback Engine

The playback engine is active when the media player is in a playback state, such as play, fast forward, and rewind. It has a special play list, called the Now Playing playlist, that is used to determine what track or content item will be played next. The PlayCurrentSelection command is used to transfer the currently selected database items to the Now Playing Playlist and start the player at a specified item within that list. Changes to the database selection before or after this PlayCurrentSelection command have no effect on the current playback.

The Database Engine

The database engine is always accessible when the unit is awake. It can be manipulated remotely and allows groups of content items to be selected, independently of the playback engine. This allows the user to listen to an existing track or playlist while checking the media player database for another selection. Once a different database selection is made, the user selection (the track or content playlist) is sent to the playback engine. The commands such as ResetDBSelection and GetNumberCategorizedDBRecords are examples of commands that are used to manipulate the Database Engine.

Database Category Hierarchies

The database engine uses categories to classify music and other records stored in the database. Possible categories are playlist, genre, artist, album, track, composer and audiobook. A list of records can be assembled, based on the various selected categories, to create a user list of records (a playlist).

The database categories have a hierarchy by which records are sorted and retrieved. This category hierarchy has an impact on the order in which records should be selected. For example, if a low category, such as album, is selected first, followed by a higher relative category such as genre, the album selection is invalidated and is ignored. When creating a new set of database selections, the accessory should begin by resetting all database selections, using the ResetDBSelection command, and selecting the desired database categories from highest to lowest relative category. A representation of the database hierarchy is shown in Table 6.

TABLE 6

Category	Notes
All (highest level)	This is the state after a ResetDBSelection command. No database categories are selected.
	If the GetNumberCategorizedDBRecords command is sent while in this state, it returns the total number of records for the requested category.
Playlist	When the SelectDBRecord command selects a playlist, all lower database category selections (genre, artist or composer, album, and track) are invalidated.
Genre	When the SelectDBRecord command selects a genre, all lower database category selections (artist or composer, album, and track) are invalidated.
Artist or Composer	When the SelectDBRecord command selects an artist or composer, all album and track category selections are invalidated.
Album	When the SelectDBRecord

TABLE 6-continued

Category	Notes
Song/Audiobook Track (lowest level)	command selects an album, all track category selections are invalidated. When the Select DBRecord command selects a track, either a song or an audiobook, it is automatically transferred from the Database Engine to the Playback Engine.

Extended Interface Mode Commands

Table 7 lists the command for the Extended Interface protocol.

TABLE 7

Command	Target engine
Acknowledge	N/A
GetCurrentPlayingTrackChapterInfo	Playback Engine
GetCurrentPlayingTrackChapterInfo	Playback Engine
GetCurrentPlayingTrackChapterName	Playback Engine
GetAudiobookSpeed	N/A
GetIndexedPlayingTrackInfo	Playback Engine
GetNumberCategorizedDBRecords	Database Engine
GetPlayStatus	Playback Engine
GetCurrentPlayingTrackIndex	Playback Engine
GetIndexedPlayingTrackTitle	N/A
GetIndexedPlayingTrackArtistName	Playback Engine
GetIndexedPlayingTrackAlbumName	Playback Engine
GetShuffle	N/A
GetRepeat	N/A
GetMonoDisplayImageLimits	N/A
GetNumPlayingTracks	Playback Engine
GetColorDisplayImageLimits	N/A
ReturnCurrentPlayingTrackChapterInfo	Playback Engine
ReturnCurrentPlayingTrackChapterPlayStatus	Playback Engine
ReturnCurrentPlayingTrackChapterName	Playback Engine
ReturnAudiobookSpeed	N/A
ReturnIndexedPlayingTrackInfo	N/A
ReturnProtocolVersion	N/A
ReturnMediaPlayerName	N/A
ReturnNumberCategorizedDBRecords	N/A
ReturnCategorizedDatabaseRecord	N/A
ReturnPlayStatus	N/A
ReturnCurrentPlayingTrackIndex	N/A
ReturnIndexedPlayingTrackTitle	N/A
ReturnIndexedPlayingTrackArtistName	N/A
ReturnIndexedPlayingTrackAlbumName	N/A
ReturnShuffle	N/A
ReturnRepeat	N/A
ReturnMonoDisplayImageLimits	N/A
ReturnNumPlayingTracks	N/A
ReturnColorDisplayImageLimits	N/A
SetCurrentPlaying-TrackChapter	Playback Engine
SetAudiobookSpeed	N/A
SetPlayStatusChangeNotification	Playback Engine
SetShuffle	N/A
SetRepeat	N/A
SetDisplayImage	N/A
SetCurrentPlayingTrack	Playback Engine
ReguestProtocolVersion	N/A
ReguestMediaPlayerName	N/A
ResetDBSelection	Database Engine
RetrieveCategorizedDatabaseRecords	Database Engine
PlayStatusChangeNotification	N/A
PlayCurrentSelection	Database and Playback Engines. This command copies items from the database to the Playback Engine.
PlayControl	Playback Engine
SelectSortDBRecord	Database Engine

Each of these commands, their direction and function are described hereinbelow.

ACK Command

Direction: media player→accessory

5 The media player sends this telegram to acknowledge the receipt of a command and return the command status. The command ID field indicates the accessory command for which the response is being sent. The command status indicates the results of the command (success or failure).

10 GetCurrentPlayingTrackChapterInfo Command

Direction: accessory→media player

Applies to: playback engine

Requests the chapter information of the currently playing track. In response, the media player sends a ReturnCurrent-
15 PlayingTrackChapterInfo telegram to the accessory.

ReturnCurrentPlayingTrackChapterInfo Command

Direction: media player→accessory

Returns the chapter information of the currently playing track. The media player sends this telegram in response to the
20 GetCurrentPlayingTrackChapterInfo telegram from the accessory. The chapter information includes the currently playing track's chapter index, as well as the total number of chapters.

SetCurrentPlayingTrackChapter Command

25 Direction: accessory→media player

Applies to: playback engine

Sets the currently playing track chapter. In response to the
SetCurrentPlayingTrackChapter command, the media player sends an ACK telegram with the command status.

30 GetCurrentPlayingTrackChapterPlayStatus

Direction: accessory→media player

Applies to: playback engine

Requests the chapter playtime status of the currently play-
ing track. In one embodiment the status includes the chapter
35 length and the time elapsed within that chapter. In response to a valid telegram, the media player sends a ReturnCurrent-
PlayingTrackChapterPlayStatus telegram to the accessory.

ReturnCurrentPlayingTrackChapterPlayStatus Command

40 Direction: media player→accessory

Returns the play status of the currently playing track chap-
ter. The media player sends this telegram in response to the
GetCurrentPlayingTrackChapterPlayStatus telegram from
the accessory.

GetCurrentPlayingTrackChapterName Command

45 Direction: accessory→media player

Applies to: playback engine

Requests a chapter name in the currently playing track. In
response to a valid telegram, the media player sends a Return-
CurrentPlayingTrackChapterName telegram to the acces-
50 sory.

ReturnCurrentPlayingTrackChapterName Command

Direction: media player→accessory

Returns a chapter name in the currently playing track. In
response to a valid telegram, the media player sends a Return-
CurrentPlayingTrackChapterName telegram to the acces-
55 sory.

ReturnCurrentPlayingTrackChapterName Command

Direction: media player→accessory

Returns a chapter name in the currently playing track. The
60 media player sends this telegram in response to a valid Get-
CurrentPlayingTrackChapterName telegram from the acces-
sory.

GetAudiobookSpeed Command

Direction: accessory→media player

65 Requests the current media player audiobook speed state.
The media player responds with the ReturnAudiobookSpeed
telegram indicating the current audiobook speed.

US 7,529,870 B1

23

ReturnAudiobookSpeed Command
 Direction: media player→accessory
 Returns the current audiobook speed setting. The media player sends this telegram in response to the GetAudiobookSpeed command from the accessory.

Table 8 shows the possible audiobook speed states returned by this command.

TABLE 8

Value	Meaning
0xFF	Slow (-1)
0x00	Normal
0x01	Fast (+1)
0x02-0xFE	Reserved

SetAudiobookSpeed Command
 Direction: accessory→media player
 Sets the speed of audiobook playback. The media player audiobook speed states are listed in Table 8. This telegram has two modes: one to set the speed of the currently playing audiobook and a second to set the audiobook speed for all audiobooks.

GetIndexedPlayingTrackInfo Command
 Direction: accessory→media player
 Applies to: playback engine
 Gets track information for the track at the specified index. The track info type field specifies the type of information to be returned, such as song lyrics, podcast name, episode date, and episode description. In response, the media player sends the ReturnIndexedPlayingTrackInfo command.

ReturnIndexedPlayingTrackInfo Command
 Direction: media player→accessory
 Returns the requested track information type and data. The media player sends this command in response to the GetIndexedPlayingTrackInfo command.

RequestProtocolVersion Command
 Direction: accessory→media player
 Requests the version of the running Extended Interface protocol from the media player. The media player responds with a ReturnProtocolVersion command.

ReturnProtocolVersion Command
 Direction: media player→accessory
 Returns the media player Extended Interface protocol version number. The media player sends this command in response to the RequestProtocolVersion command from the accessory.

RequestMediaPlayerName Command
 Direction: accessory→media player
 Returns the name of the user's media player or "media player" if the media player name is undefined. This allows the media player name to be shown in the human-machine interface.

ReturnMediaPlayerName Command
 Direction: media player→accessory
 The media player sends this command in response to the RequestMediaPlayerName telegram from the accessory.

ResetDBSelection Command
 Direction: accessory→media player
 Applies to: database engine
 Resets the current database selection to an empty state and invalidates the category entry count. That is, it sets the count to 0 for all categories except the playlist category.

SelectDBRecord Command
 Direction: accessory→media player

24

Applies to: database engine. Selecting a single track automatically passes it to the playback engine.

Selects one or more records in the database engine, based on a category relative index.

5 GetNumberCategorizedDBRecords Command
 Direction: accessory→media player
 Applies to: database engine
 Retrieves the number of records in a particular database category.

10 ReturnNumberCategorizedDBRecords Command
 Direction: media player→accessory
 Returns the number of database records matching the specified database category.

RetrieveCategorizedDatabaseRecords Command
 15 Direction: accessory→media player
 Applies to: database engine
 Retrieves one or more database records from the media player, typically based on the results from the GetNumberCategorizedDBRecords command query.

20 ReturnCategorizedDatabaseRecord Command
 Direction: media player→accessory
 Contains information for a single database record.

GetPlayStatus Command
 Direction: accessory→media player
 25 Applies to: playback engine
 Requests the current media player playback status, allowing the accessory to display feedback to the user.

ReturnPlayStatus Command
 Direction: media player→accessory
 30 Returns the current media player playback status.

GetCurrentPlayingTrackIndex Command
 Direction: accessory→media player
 Applies to: playback engine
 Requests the playback engine index of the currently play-
 35 ing track.

ReturnCurrentPlayingTrackIndex Command
 Direction: media player→accessory
 Returns the playback engine index of the current playing track in response to the GetCurrentPlayingTrackIndex telegram from the accessory.

40 GetIndexedPlayingTrackTitle Command
 Direction: accessory→media player
 Applies to: playback engine
 Requests the title name of the indexed playing track from the media player. In response to a valid telegram, the media player sends a ReturnIndexedPlayingTrackTitle telegram to the accessory.

45 ReturnIndexedPlayingTrackTitle Command
 Direction: media player→accessory
 Returns the title of the indexed playing track in response to a valid GetIndexedPlayingTrackTitle telegram from the accessory.

GetIndexedPlayingTrackArtistName Command
 Direction: accessory→media player
 55 Applies to: playback engine
 Requests the name of the artist of the indexed playing track. In response to a valid telegram, the media player sends a ReturnIndexedPlayingTrackArtistName telegram to the accessory.

60 ReturnIndexedPlayingTrackArtistName Command
 Direction: media player→accessory
 Returns the artist name of the indexed playing track in response to a valid GetIndexedPlayingTrackArtistName telegram from the accessory.

65 GetIndexedPlayingTrackAlbumName Command
 Direction: accessory→media player
 Applies to: playback engine

US 7,529,870 B1

25

Requests the album name of the indexed playing track.
ReturnIndexedPlayingTrackAlbumName Command
Direction: media player→accessory
Returns the album name of the indexed playing track in response to a valid GetIndexedPlayingTrackAlbumName telegram from the accessory. 5
SetPlayStatusChangeNotification Command
Direction: accessory→media player
Applies to: playback engine
Sets the state of play status change notifications from the media player to the accessory. 10
PlayStatusChangeNotification Command
Direction: media player→accessory
The media player sends this telegram to the accessory when the media player play status changes, if the accessory has previously enabled notifications using the SetPlayStatusChangeNotification command. This telegram contains details about the new play status. 15
PlayCurrent Selection Command
Direction: accessory→media player
Applies to: playback engine and database engines. This command copies items from the database engine to the playback engine.
Requests playback of the currently selected track or list of tracks. 20
PlayControl Command
Direction: accessory→media player
Applies to: playback engine
Sets the new play state of the media player.
GetShuffle Command
Direction: accessory→media player
Requests the current state of the media player shuffle setting. 25
ReturnShuffle Command
Direction: media player→accessory
Returns the current state of the shuffle setting. 30
SetShuffle Command
Direction: accessory→media player
Sets the media player shuffle mode.
GetRepeat Command
Direction: accessory→media player
Requests the track repeat state of the media player. 35
ReturnRepeat Command
Direction: media player→accessory
Returns the current media track repeat state to the accessory. 40
SetRepeat Command
Direction: accessory→media player
Sets the repeat state of the media player.
SetDisplayImage Command
Direction: accessory→media player
Sets a bitmap image that is shown on the media player display when it is connected to the accessory. 45
GetMonoDisplayImageLimits Command
Direction: accessory→media player
Requests the limiting characteristics of the monochrome image that can be sent to the media player for display while it is connected to the accessory. 50
ReturnMonoDisplayImageLimits Command
Direction: media player→accessory
Returns the limiting characteristics of the monochrome image that can be sent to the media player for display while it is connected to the accessory. 55
GetNumPlayingTracks Command
Direction: accessory→media player
Applies to: playback engine 60

26

Requests the number of tracks in the list of tracks queued to play on the media player.
ReturnNumPlayingTracks Command
Direction: media player→accessory
Returns the number of tracks in the actual list of tracks queued to play, including the currently playing track (if any).
SetCurrentPlayingTrack Command
Direction: accessory→media player
Applies to: playback engine
Sets the index of the track to play in the Now Playing playlist on the media player. 65
SelectSortDBRecord Command
Direction: accessory→media player
Applies to: database engine
Selects one or more records in the media player database, based on a category-relative index.
GetColorDisplayImageLimits Command
Direction: accessory→media player
Requests the limiting characteristics of the color image that can be sent to the media player for display while it is connected to the accessory. 70
ReturnColorDisplayImageLimits Command
Direction: media player→accessory
Returns the limiting characteristics of the color image that can be sent to the media player for display while it is connected to the accessory. 75
RF Transmitter Lingo
The RF Transmitter Lingo is used for accessories that transmit the media player analog audio over radio frequencies (typically an unused frequency in the FM band). The Begin Transmission command packet notifies the external RF transmitter accessory that the media player is entering playback mode. The End Transmission command packet notifies the RF transmitter that the media player is exiting playback mode (that is, it is stopped, entering light sleep mode, and so forth). 80
Begin Transmission Command
Direction: media player→accessory
The media player sends this command to notify the accessory that high power may be used and that it should begin transmitting. 85
End Transmission Command
Direction: media player→accessory
The media player sends this command to notify the accessory to stop transmitting and to stop using accessory high power. 90
Single Lingo Accessory Identification
Accessories should use the general lingo identify command to identify themselves as supporting a single accessory lingo. FIG. 7 illustrates a single-lingo accessory command flow. In this command flow, first the accessory waits for media player internal bootstrap/wakeup and allows the media player to synchronize to the accessory rate. Next, the media player is informed that accessory is present and supports the Extended Interface lingo. Thereafter, the accessory gets (requests) the version of the extended interface lingo protocol supported by the media player and the media player sends the protocol version for verification. If received, the accessory has successfully forced the media player into extended interface mode. The accessory should use the returned lingo version to determine which features are available from the media player. 95
Multiple Lingo Accessory Identification
Accessories should use the general lingo IdentifyAccessoryLingoes command to identify themselves as supporting multiple accessory lingoes (not including the general lingo), or when they are communicating with the media player over the USB port link. 100

US 7,529,870 B1

27

Accessories that support the extended interface mode and identify as a multi-lingo accessory do not automatically switch into the extended interface mode after the identification process completes. These accessories must explicitly switch into and out of the extended interface mode, using the general lingo commands.

Accessories that identify themselves as supporting multiple lingo should use the steps shown in FIG. 8. FIG. 8 illustrates a multiple lingo command flow. In this flow, first the accessory waits for the media player bootstrap/wakeup and allows the media player to synchronize to the accessory rate. Next, the media player is informed that the accessory is present, and the bit field of supported lingo is sent. The media player acknowledges the IdentifyAccessoriesLingos command. Next, the media player general lingo version information is requested, and the media player sends the general lingo protocol version. This information is used to determine which features are available from the media player. The accessory is then switched into the extended interface mode, sending the EnterExtendedInterfaceMode Command, and the media player sends a first ACK indicating how long it will take the media player to switch modes. The media player sends a second ACK notifying the accessory of the mode switch, and the media player sends an extended interface protocol version for verification. If received, the accessory has successfully forced the media player into extended interface mode. The accessory should use the returned information to determine which features are available from the media player. This figure assumes the accessory is using the UART serial port link.

A connector interface system for a communication accessory is disclosed. The interface includes a docking connector. The docking connector includes first make/last break contacts that minimize internal damage to the internal electronics. The docking connector also includes a specific keying arrangement to prevent noncompliant connectors from being plugged in, and thereby minimizes potential damage to the multi-communication accessory. The remote connector provides for the ability to output audio, input audio, and output video. The connector system interface also includes a media player protocol to control accessory features. These controls allow for a user to control the functionality of media player in a variety of modes.

It will be clear that the invention described herein provides significant functional enhancements when incorporated into prior art systems. FIG. 9 shows an exemplary system 900 using the connector described herein. The system includes at least two accessories 902 and 904 to be connected together. Each accessory 902 and 904 contains an enhanced connector system interface 902a and 902b as described herein. They can be connected together either wirelessly or via a cable. Alternatively, one of the accessories can be itself constructed integrally with the connector so that it can be docked in place when mated with the other accessory, for example, as is done with flash-backed memory sticks, or similar accessories that include an integral connector. When connected together, the accessories can exchange data via the connector interface system signals that would not be conveniently possible but for the existence of the connector pins and the protocol.

The connection interface system can be utilized with a plurality of accessory devices interfaces including, but not limited to, car stereo interface, home stereo interfaces, remote controls, radio interfaces, microphones, voice recorders, wireless accessories which allow for communications to other accessories and speakers.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary

28

skill in the art will readily recognize that there could be variations to the embodiments and those variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.

What is claimed is:

1. A method for use in an accessory, the method comprising:

establishing communication with a media player according to a media player accessory protocol comprising a general lingo and an accessory lingo; and providing functionality and capability of the accessory to the media player using the accessory lingo.

2. The method of claim 1 further comprising authenticating the accessory to the media player using the general lingo.

3. The method of claim 1 further comprising identifying the accessory to the media player, wherein the identification includes requesting information indicating a version of a media player accessory protocol from the media player.

4. The method of claim 1 wherein the media player accessory protocol includes a plurality of accessory lingo.

5. The method of claim 1 wherein the accessory comprises any of a car stereo, a home stereo, a remote control, a radio, a voice recorder, a docking station, or a wireless device for communication with other accessories.

6. The method of claim 3 wherein the media accessory protocol includes at least one command that allows the accessory to be controlled by or control the media player.

7. The method of claim 4 wherein the plurality of accessory lingo comprises one or more of a microphone lingo, a simple remote lingo, a display remote lingo, an extended interface lingo or an RF transmitter lingo, wherein each of the accessory lingo includes at least one command which controls a functionality of the accessory or the media player.

8. The method of claim 1 wherein the general lingo comprises one or more of the following commands:

a Request Identify command; an Identify command; an ACK command; a RequestMediaPlayerName command; a ReturnMediaPlayerName command; a RequestMediaPlayerSoftwareVersion command; a ReturnMediaPlayerSoftwareVersion command; a RequestMediaPlayerSerialNum command; a ReturnMediaPlayerSerialNumber command; a RequestMediaPlayerModelNum command; a RequestLingoProtocolVersion command; a ReturnLingoProtocolVersion command; an IdentifyAccessoryLingo command; a GetDevAuthenticationInfo command; a RetDevAuthenticationInfo command; an AckDevAuthenticationInfo command; a GetDevAuthenticationSignature command; a RetDevAuthenticationSignature command; an AckDevAuthenticationStatus command; a GetMediaPlayerAuthenticationInfo command; a RetMediaPlayerAuthenticationInfo command; an AckMediaPlayerAuthenticationInfo command; a GetMediaPlayerAuthenticationSignature command; a RetMediaPlayerAuthenticationSignature command; an AckMediaPlayerAuthenticationStatus command; or a NotifyMediaPlayerStateChange command.

9. The method of claim 7 wherein the microphone lingo comprises one or more of the following commands:

a BeginRecord command; an EndRecord command; a BeginPlayback command; an EndPlayback command; an ACK command; a GetDevAck command; a MediaPlayerModeChange command; a GetDevCaps com-

US 7,529,870 B1

29

mand; a RetDevCaps command; a GetDevCtrl command; a RetDevCtrl command; or a SetDevCtrl command.

10. The method of claim 7 wherein the simple remote lingo comprises:

a ContextButtonStatus command.

11. The method of claim 7 wherein the display remote lingo comprises one or more of the following commands:

an ACK command; a GetCurrentEQProfileIndex command; a RetCurrentEQProfileIndex command; a SetCurrentEQProfileIndex command; a GetNumEQProfiles command; a RetNumEQProfiles command; a GetIndexedEQProfileName command; a RetIndexedEQProfileName command; a SetRemoteEventNotification command; a RemoteEventNotification command; a GetRemoteEventStatus command; a RetRemoteEventStatus command; a GetMediaPlayerStateInfo command; a RetMediaPlayerStateInfo command; a SetMediaPlayerStateInfo command; a GetPlayStatus command; a RetPlayStatus command; a SetCurrentPlayingTrack command; a GetIndexedPlayingTrackInfo command; a RetIndexedPlayingTrackInfo command; a GetNumPlayingTracks command; a RetNumPlayingTracks command; a GetPowerBatteryState command; a RetPowerBatteryState command; a GetSoundCheckState command; a RetSoundCheckState command; or a SetSoundCheckState command.

12. The method of claim 7 wherein the extended interface lingo comprises one or more of the following commands:

an Acknowledge command; a GetCurrentPlaying-TrackChapterInfo command; a ReturnCurrentPlaying-TrackChapterInfo command; a SetCurrentPlaying-TrackChapter command; a GetCurrentPlaying-TrackChapterPlayStatus command; a ReturnCurrentPlaying-TrackChapterPlayStatus command; a GetCurrentPlayingTrackChapterName command; a ReturnCurrentPlaying-TrackChapterName command; a GetAudiobookSpeed command; a ReturnAudiobookSpeed command; a SetAudiobookSpeed command; a GetIndexedPlaying-TrackInfo command; a ReturnIndexedPlaying-TrackInfo command; a RequestProtocolVersion command; a ReturnProtocolVersion command; a RequestMediaPlayerName command; a ReturnMediaPlayerName command; a ResetDBSelection command; a SelectDBRecord command; a GetNumberCategorized-DBRecords command; a ReturnNumber-CategorizedDBRecords command; a RetrieveCategorized-DatabaseRecords command; a ReturnCategorized-DatabaseRecord command; a GetPlayStatus command; a ReturnPlayStatus command; a GetCurrentPlaying-TrackIndex command; a ReturnCurrentPlaying-TrackIndex command; a GetIndexedPlaying-TrackTitle command; a GetIndexedPlaying-TrackArtistName command; a ReturnIndexedPlaying-TrackArtistName command; a GetIndexedPlaying-TrackAlbumName command; a ReturnIndexedPlaying-TrackAlbumName command; a SetPlayStatusChange-Notification command; a PlayStatusChange-Notification command; a PlayCurrentSelection command; a PlayControl command; a GetShuffle command; a ReturnShuffle command; a SetShuffle command; a GetRepeat command; a ReturnRepeat command; a SetRepeat command; a SetDisplayImage command; a GetMonoDisplay-ImageLimits command; a ReturnMonoDisplay-ImageLimits command; a GetNumPlayingTracks command; a ReturnNumPlayingTracks command; a SetCurrentPlayingTrack com-

30

mand; a SelectSortDBRecord command; a GetColorDisplay-ImageLimits command; or a ReturnColorDisplay-ImageLimits command.

13. The method of claim 7 wherein the RF transmitter lingo comprises one or more of the following commands:

a BeginTransmission command; or an EndTransmission command.

14. An accessory comprising:

a component having a functionality; and

a connector interface system in communication with the component;

the connector interface system being capable of communicating with a media player that utilizes a protocol comprising at least one lingo, wherein the accessory is capable of utilizing the protocol to provide the functionality of the component to the media player.

15. The accessory of claim 14 wherein the accessory is capable of being authenticated by the media player with the protocol.

16. The accessory of claim 14 wherein the accessory is capable of being identified by requesting a version of the protocol from the media player.

17. The accessory of claim 14 wherein the protocol includes at least one command that allows the accessory to be controlled by or control the media player.

18. The accessory of claim 14 wherein the at least one lingo comprises a plurality of lingo, wherein the plurality of lingo comprises a general lingo and at least one accessory lingo.

19. The accessory of claim 14 wherein the accessory comprises any of a car stereo, a home stereo, a remote control, a radio, a voice recorder, a docking station, or a wireless device for communication to other accessories.

20. The accessory of claim 18 wherein the at least one accessory lingo is utilized by the accessory to provide the functionality of the component.

21. The accessory of claim 20 wherein the at least one accessory lingo comprises any one or more of a microphone lingo, a simple remote lingo, a display remote lingo, an extended interface lingo, or an RF transmitter lingo.

22. The accessory of claim 21 wherein the plurality of general lingo commands comprise one or more of:

a Request Identify command; an Identify command; an ACK command; a RequestMediaPlayerName command; a ReturnMediaPlayerName command; a RequestMediaPlayerSoftwareVersion command; a ReturnMediaPlayerSoftwareVersion command; a RequestMediaPlayerSerialNum command; a ReturnMediaPlayerSerialNumber command; a RequestMediaPlayerModelNum command; a RequestLingoProtocolVersion command; a ReturnLingoProtocolVersion command; an IdentifyAccessoryLingo command; a GetDevAuthenticationInfo command; a RetDevAuthenticationInfo command; an AckDevAuthenticationInfo command; a GetDevAuthenticationSignature command; a RetDevAuthenticationSignature command; an AckDevAuthenticationStatus command; a GetMediaPlayerAuthenticationInfo command; a RetMediaPlayerAuthenticationInfo command; an AckMediaPlayerAuthenticationInfo command; a GetMediaPlayerAuthenticationSignature command; a RetMediaPlayerAuthenticationSignature command; an AckMediaPlayerAuthenticationStatus command; or a NotifyMediaPlayerStateChange command.

23. The accessory of claim 21 wherein the microphone lingo comprises any one or more of the following commands:

US 7,529,870 B1

31

a BeginRecord command; an EndRecord command; a BeginPlayback command; an EndPlayback command; an ACK command; a GetDevAck command; a MediaPlayerModeChange command; a GetDevCaps command; a RetDevCaps command; a GetDevCtrl command; a RetDevCtrl command; or a SetDevCtrl command.

24. The accessory of claim 21 wherein the simple remote lingo comprises:
a ContextButtonStatus command.

25. The accessory of claim 21 wherein the display remote lingo comprises any one or more of the following commands:
an ACK command; a GetCurrentEQProfileIndex command; a RetCurrentEQProfileIndex command; a SetCurrentEQProfileIndex command; a GetNumEQProfiles command; a RetNumEQProfiles command; a GetIndexedEQProfileName command; a RetIndexedEQProfileName command; a SetRemoteEventNotification command; a RemoteEventNotification command; a GetRemoteEventStatus command; a RetRemoteStatus command; a GetMediaPlayerStateInfo command; a RetMediaPlayerStateInfo command; a SetMediaPlayerStateInfo command; a GetPlayStatus command; a RetPlayStatus command; a SetCurrentPlayingTrack command; a GetIndexedPlayingTrackInfo command; a RetIndexedPlayingTrackInfo command; a GetNumPlayingTracks command; a RetNumPlayingTracks command; a GetPowerBatteryState command; a RetPowerBatteryState command; a GetSoundCheckState command; a RetSoundCheckState command; or a SetSoundCheckState command.

26. The accessory of claim 21 wherein the extended interface lingo comprises any one or more of the following commands:
an Acknowledge command; a GetCurrentPlaying-TrackChapterInfo command; a ReturnCurrentPlaying-TrackChapterInfo command; a SetCurrentPlaying-TrackChapter command; a GetCurrentPlaying-TrackChapterPlayStatus command; a ReturnCurrentPlaying-TrackChapterPlayStatus command; a GetCurrentPlaying-TrackChapterName command; a ReturnCurrentPlaying-TrackChapterName command; a SetAudiobookSpeed command; a GetIndexedPlaying-TrackInfo command; a ReturnIndexedPlaying-TrackInfo command; a RequestProtocolVersion command; a ReturnProtocolVersion command; a RequestMediaPlayerName command; a ReturnMediaPlayerName command; a ResetDBSelection command; a SelectDBRecord command; a GetNumberCategorized-DBRecords command; a ReturnNumber-Categorized-DBRecords command; a RetrieveCategorized-DatabaseRecords command; a ReturnCategorized-DatabaseRecord command; a GetPlayStatus command; a ReturnPlayStatus command; a GetCurrentPlaying-TrackIndex command; a ReturnCurrentPlaying-TrackIndex command; a GetIndexedPlaying-TrackTitle command; a GetIndexedPlaying-TrackArtistName command; a ReturnIndexedPlayingTrackArtistName command; a GetIndexedPlaying-TrackAlbumName command; a ReturnIndexedPlaying-TrackAlbumName command; a SetPlayStatusChange-Notification command; a PlayStatusChange-Notification command; a PlayCurrentSelection command; a PlayControl command; a GetShuffle command; a ReturnShuffle command; a SetShuffle command; a GetRepeat command; a ReturnRepeat command; a SetRepeat command; a SetDisplayImage command; a GetMonoDisplay-Image-

32

Limits command; a ReturnMonoDisplay-ImageLimits command; a GetNumPlayingTracks command; a ReturnNumPlayingTracks command; a SetCurrentPlayingTrack command; a SelectSortDBRecord command; a GetColorDisplay-ImageLimits command; or a ReturnColorDisplay-ImageLimits command.

27. The accessory of claim 21 wherein the RF transmitter lingo comprises any one or more of the following commands:
a BeginTransmission command; or an EndTransmission command.

28. A method for use in an accessory, the method comprising:
establishing communication with a media player according to a media player accessory protocol comprising a general lingo and an accessory lingo;
receiving a command of the accessory lingo from the media player; and
invoking a functionality of the accessory in response to the command.

29. A method for use in an accessory, the method comprising:
establishing communication with a media player according to a media player accessory protocol comprising a general lingo and an accessory lingo; and
sending a command of the accessory lingo to the media player, wherein the command invokes a functionality of the media player.

30. The method of claim 29 further comprising receiving a response to the command from the media player.

31. A computer program product comprising a computer readable storage medium encoded with program instructions, which when executed by a processor in an accessory device cause the processor to execute a method comprising:
establishing communication with a media player according to a media player accessory protocol comprising a general lingo and an accessory lingo; and
providing functionality and capability of the accessory to the media player using the accessory lingo.

32. The computer program product of claim 31 wherein the method executed by the processor further comprises authenticating the accessory to the media player using the general lingo.

33. The computer program product of claim 31 wherein the method executed by the processor further comprises identifying the accessory to the media player, wherein the identification includes requesting information indicating a version of a media player accessory protocol from the media player.

34. The computer program product of claim 33 wherein the media accessory protocol includes at least one command that allows the accessory to be controlled by or control the media player.

35. The computer program product of claim 31 wherein the media player accessory protocol includes a plurality of accessory lingo.

36. The computer program product of claim 35 wherein the plurality of accessory lingo comprises one or more of a microphone lingo, a simple remote lingo, a display remote lingo, an extended interface lingo or an RF transmitter lingo, wherein each of the accessory lingo includes at least one command which controls a functionality of the accessory or the media player.

37. The computer program product of claim 31 wherein the accessory device comprises any of a car stereo, a home stereo, a remote control, a radio, a voice recorder, a docking station, or a wireless device for communication with other accessory devices.

US 7,529,870 B1

33

38. A computer program product comprising a computer readable storage medium encoded with program instructions, which when executed by a processor in an accessory device cause the processor to execute a method comprising:
establishing communication with a media player according to a media player accessory protocol comprising a general lingo and an accessory lingo;
receiving a command of the accessory lingo from the media player; and
invoking a functionality of the accessory device in response to the command.

39. A computer program product comprising a computer readable storage medium encoded with program instructions,

34

which when executed by a processor in an accessory device cause the processor to execute a method comprising:

establishing communication with a media player according to a media player accessory protocol comprising a general lingo and an accessory lingo; and
sending a command of the accessory lingo to the media player, wherein the command invokes a functionality of the media player.

40. The computer program product of claim **39** wherein the method executed by the processor further comprises receiving a response to the command from the media player.

* * * * *

EXHIBIT F



US007529872B1

(12) **United States Patent**
Schubert et al.

(10) **Patent No.:** **US 7,529,872 B1**
(45) **Date of Patent:** **May 5, 2009**

(54) **COMMUNICATION BETWEEN AN ACCESSORY AND A MEDIA PLAYER USING A PROTOCOL WITH MULTIPLE LINGOES**

(75) Inventors: **Emily C. Schubert**, Los Altos, CA (US); **Wang Chun Leung**, Mountain View, CA (US); **Gregory T. Lydon**, Santa Cruz, CA (US); **Scott Krueger**, San Francisco, CA (US); **Paul Holden**, Sunnyvale, CA (US); **John Archibald**, San Francisco, CA (US); **Lawrence G. Bolton**, Fremont, CA (US); **Donald J. Novotney**, San Jose, CA (US); **John B. Filson**, San Jose, CA (US); **David Tupman**, San Francisco, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 207 days.

(21) Appl. No.: **11/479,894**

(22) Filed: **Jun. 30, 2006**

Related U.S. Application Data

(63) Continuation-in-part of application No. 10/833,689, filed on Apr. 27, 2004, now Pat. No. 7,441,062.

(60) Provisional application No. 60/784,306, filed on Mar. 20, 2006.

(51) **Int. Cl.**
G06F 13/42 (2006.01)

(52) **U.S. Cl.** **710/105**; 710/300; 710/16; 710/62; 710/64

(58) **Field of Classification Search** 710/313-15, 710/110, 8-12, 62-64, 72, 16; 455/414-420, 455/556.1-556.2, 571-572

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,673,861 A	6/1987	Dubovsky et al.
4,850,899 A	7/1989	Maynard
4,924,216 A	5/1990	Leung
4,938,483 A	7/1990	Yavetz
5,041,025 A	8/1991	Haitmanek

(Continued)

FOREIGN PATENT DOCUMENTS

EP	1104150	5/2001
----	---------	--------

(Continued)

OTHER PUBLICATIONS

Altec Lansing, "inMotion Users Guide," Corp. Headquarters, 535 Rte.6 & 209, Milford, PA 18337.

(Continued)

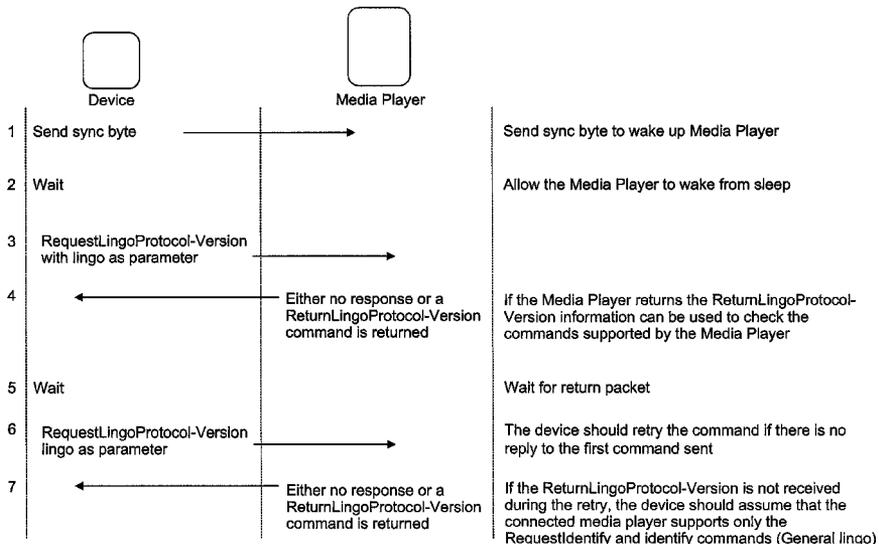
Primary Examiner—Raymond N Phan

(74) *Attorney, Agent, or Firm*—Townsend and Townsend and Crew LLP

(57) **ABSTRACT**

A connector interface system is disclosed. The connector interface system includes an interface and a protocol in communication with which allows a media player to communicate with external accessories over a transport link. The protocol includes a core protocol functionality and a plurality of accessory lingo. The accessory lingo comprise a microphone lingo, a simple remote lingo, a display remote lingo, a RF transmitter lingo and an extended interface lingo.

54 Claims, 14 Drawing Sheets



US 7,529,872 B1

Page 2

U.S. PATENT DOCUMENTS						
			6,485,328	B1	11/2002	Wu
5,055,069	A	10/1991	Townsend et al.			
5,080,603	A	1/1992	Mouissie			
5,104,243	A	4/1992	Harding			
5,108,313	A	4/1992	Adams			
5,150,031	A	9/1992	James et al.			
5,186,646	A	2/1993	Pederson			
5,247,138	A	9/1993	Landmeier			
5,277,624	A	1/1994	Champion			
5,471,128	A	11/1995	Patino et al.			
5,525,981	A	6/1996	Abernethy			
5,586,893	A	12/1996	Mosquera			
5,592,588	A	1/1997	Reekes et al.			
5,618,045	A	4/1997	Kagan et al.			
5,648,712	A	7/1997	Hahn			
5,660,558	A	8/1997	Osanai et al.			
5,727,866	A	3/1998	Kraines et al.			
5,754,027	A	5/1998	Oglesbee et al.			
5,830,001	A	11/1998	Kinoshita			
5,845,217	A	12/1998	Lindell et al.			
5,901,049	A	5/1999	Schmidt et al.			
5,964,847	A	10/1999	Booth et al.			
5,975,957	A	11/1999	Noda et al.			
6,007,372	A	12/1999	Wood			
6,012,105	A	1/2000	Rubbmark et al.			
6,031,797	A	2/2000	Van Ryzint et al.			
6,053,773	A	4/2000	Wu			
6,078,402	A *	6/2000	Fischer et al. 358/1.16			
6,078,789	A	6/2000	Bodenmann et al.			
6,125,455	A	9/2000	Yeo			
6,130,518	A	10/2000	Gabehart et al.			
6,139,373	A	10/2000	Ward et al.			
6,154,773	A	11/2000	Roberts et al.			
6,154,798	A	11/2000	Lin et al.			
6,161,027	A *	12/2000	Poirel 455/572			
6,169,387	B1	1/2001	Kaib			
6,175,358	B1	1/2001	Scott-Jackson et al.			
6,178,514	B1	1/2001	Wood			
6,184,652	B1	2/2001	Yang			
6,184,655	B1	2/2001	Malackowski			
6,203,345	B1	3/2001	Roque et al.			
6,204,637	B1	3/2001	Rengan			
6,206,480	B1	3/2001	Thompson			
6,211,581	B1	4/2001	Farrant			
6,211,649	B1	4/2001	Matsuda			
6,224,420	B1	5/2001	Nishio et al.			
6,230,205	B1	5/2001	Garrity et al.			
6,234,827	B1	5/2001	Nishio et al.			
6,252,380	B1	6/2001	Koenck			
6,261,109	B1	7/2001	Liu et al.			
6,267,623	B1	7/2001	Hisamatsu			
6,268,845	B1	7/2001	Pariza et al.			
6,271,605	B1	8/2001	Carkner et al.			
6,280,251	B1	8/2001	Nishio et al.			
6,283,789	B1	9/2001	Tsai			
6,304,764	B1	10/2001	Pan			
6,314,479	B1	11/2001	Frederick et al.			
6,316,916	B2	11/2001	Bohne			
6,322,396	B1	11/2001	Kwan			
6,344,727	B1	2/2002	Desai et al.			
6,353,894	B1	3/2002	Pione			
6,354,713	B1	3/2002	Leifer et al.			
6,358,089	B1	3/2002	Kuroda et al.			
6,385,596	B1	5/2002	Wiser et al.			
6,394,905	B1	5/2002	Takeda et al.			
6,431,915	B1	8/2002	Ko			
6,453,371	B1	9/2002	Hampson et al.			
6,454,592	B2	9/2002	Takagi			
6,461,173	B1	10/2002	Mizuno et al.			
6,468,110	B2	10/2002	Fujino et al.			
6,478,603	B1	11/2002	Wu			
6,483,428	B1	11/2002	Fish et al.			
			6,489,751	B2	12/2002	Small et al.
			6,505,160	B1	1/2003	Levy et al.
			6,524,119	B2	2/2003	Kato et al.
			6,558,201	B1	5/2003	Begley et al.
			6,577,877	B1	6/2003	Charlier et al.
			6,589,076	B1	7/2003	Davis et al.
			6,591,085	B1	7/2003	Grady
			6,608,264	B1	8/2003	Fouladpour
			6,608,399	B2	8/2003	McConnell et al.
			6,614,232	B1	9/2003	Mukai
			6,616,473	B2	9/2003	Kamata et al.
			6,629,197	B1	9/2003	Bhogal et al.
			6,642,629	B2	11/2003	DeLeeuw
			6,653,813	B2	11/2003	Khatri
			6,663,420	B1	12/2003	Xiao
			6,665,803	B2 *	12/2003	Lunsford et al. 713/320
			6,674,995	B1	1/2004	Meyers et al.
			6,724,339	B2	4/2004	Conway et al.
			6,725,061	B1	4/2004	Hutchinson, IV et al.
			6,728,546	B1	4/2004	Peterson et al.
			6,728,729	B1	4/2004	Jawa et al.
			6,747,859	B2	6/2004	Walbeck et al.
			6,754,468	B1	6/2004	Sieben et al.
			6,761,635	B2	7/2004	Hoshino et al.
			6,774,939	B1	8/2004	Peng
			6,776,626	B2	8/2004	Huang et al.
			6,776,660	B1	8/2004	Kubota et al.
			6,776,665	B2	8/2004	Huang
			6,799,226	B1	9/2004	Robbin et al.
			6,813,528	B1	11/2004	Yang
			6,816,376	B2	11/2004	Bright et al.
			6,830,160	B2	12/2004	Risolia
			6,859,538	B1	2/2005	Voltz
			6,859,854	B2	2/2005	Kwong
			6,931,456	B2	8/2005	Payne et al.
			6,939,177	B2	9/2005	Kato et al.
			6,991,483	B1	1/2006	Milan et al.
			7,004,787	B2	2/2006	Milan
			7,050,783	B2	5/2006	Curtiss et al.
			7,054,888	B2	5/2006	La Chapelle et al.
			7,062,261	B2 *	6/2006	Goldstein et al. 455/419
			7,108,560	B1	9/2006	Chou et al.
			7,127,678	B2	10/2006	Bhesania et al.
			7,127,879	B2	10/2006	Zhu et al.
			7,167,112	B2	1/2007	Andersen et al.
			7,167,935	B2 *	1/2007	Hellberg 710/58
			7,187,947	B1 *	3/2007	White et al. 455/556.1
			7,215,042	B2 *	5/2007	Yan 307/116
			7,281,214	B2	10/2007	Fadell
			7,293,122	B1	11/2007	Schubert et al.
			7,293,227	B2	11/2007	Plastina et al.
			7,299,304	B2	11/2007	Saint-Hilaire et al.
			7,303,282	B2 *	12/2007	Dwyer et al. 353/15
			7,305,506	B1	12/2007	Lydon et al.
			2001/0003205	A1	6/2001	Gilbert
			2001/0005641	A1	6/2001	Matsumoto et al.
			2001/0006884	A1	7/2001	Matsumoto
			2002/0002035	A1	1/2002	Sim et al.
			2002/0010759	A1	1/2002	Hitson et al.
			2002/0029303	A1	3/2002	Nguyen
			2002/0065074	A1	5/2002	Cohn et al.
			2002/0068610	A1	6/2002	Anvekar et al.
			2002/0072390	A1	6/2002	Uchiyama
			2002/0103008	A1	8/2002	Rahn et al.
			2002/0105861	A1	8/2002	Leapman
			2002/0115480	A1	8/2002	Huang
			2002/0132651	A1	9/2002	Jinnouchi
			2002/0151327	A1	10/2002	Levitt
			2002/0152874	A1	10/2002	Vilcauskas et al.
			2002/0156546	A1	10/2002	Ramaswamy
			2002/0156949	A1	10/2002	Kubo et al.
			2002/0173273	A1	11/2002	Spurgat et al.

US 7,529,872 B1

Page 3

2002/0174269	A1	11/2002	Spurgat et al.	JP	10-321302	4/1998
2002/0194621	A1	12/2002	Tran et al.	JP	10-334993	12/1998
2003/0004934	A1	1/2003	Qian	JP	11-288420	10/1999
2003/0011608	A1	1/2003	Wada	JP	2000-214953	8/2000
2003/0028664	A1	2/2003	Tan et al.	JP	2000-223215	8/2000
2003/0041206	A1	2/2003	Dickie	JP	2000-223216	8/2000
2003/0059022	A1	3/2003	Nebiker et al.	JP	2000-223218	8/2000
2003/0067741	A1	4/2003	Alfonso et al.	JP	2001-035603	2/2001
2003/0073432	A1	4/2003	Meade	JP	2001-069165	A 3/2001
2003/0079038	A1	4/2003	Robbin et al.	JP	2001-196133	7/2001
2003/0092809	A1	5/2003	Chen	JP	2001-230021	8/2001
2003/0097379	A1	5/2003	Iretton	JP	2001-332350	11/2001
2003/0110403	A1	6/2003	Crutchfield et al.	JP	2002-025720	1/2002
2003/0151621	A1	8/2003	McEvelly et al.	JP	2002-203641	7/2002
2003/0172209	A1	9/2003	Liu et al.	JP	2002 245719	8/2002
2003/0185395	A1	10/2003	Lee et al.	JP	2002-252566	9/2002
2003/0198015	A1	10/2003	Vogt	JP	3090747	U 10/2002
2003/0220988	A1 *	11/2003	Hymel 709/220	JP	2002-374447	12/2002
2003/0236075	A1	12/2003	Johnson et al.	JP	2003-17165	1/2003
2003/0237043	A1	12/2003	Novak et al.	JP	2003-032351	1/2003
2004/0003300	A1	1/2004	Malueg et al.	JP	2003-274386	9/2003
2004/0019497	A1	1/2004	Volk et al.	JP	2002-342659	11/2003
2004/0039860	A1	2/2004	Mills et al.	WO	WO 99/026330	5/1999
2004/0048569	A1	3/2004	Kawamura	WO	WO 00/039907	7/2000
2004/0090998	A1	5/2004	Chen	WO	WO 00/060450	10/2000
2004/0103223	A1 *	5/2004	Gabehart et al. 710/2	WO	WO 02/049314	6/2002
2004/0116005	A1 *	6/2004	Choi 439/894	WO	WO 03/036541	A1 5/2003
2004/0162029	A1	8/2004	Grady	WO	WO 03/036957	A1 5/2003
2004/0186935	A1	9/2004	Bel et al.	WO	WO 03/073688	9/2003
2004/0194154	A1	9/2004	Meadors et al.	WO	WO 2004/084413	9/2004
2004/0198436	A1	10/2004	Alden	WO	WO 2004/112311	A 12/2004
2004/0224638	A1	11/2004	Fadell et al.	WO	WO 2005/119463	A 12/2005
2004/0235339	A1	11/2004	Sato et al.	WO	WO 2006/080957	A1 8/2006
2004/0249994	A1	12/2004	Shapiro et al.			
2004/0252966	A1	12/2004	Halloway et al.			
2004/0267812	A1	12/2004	Harris et al.			
2004/0267825	A1	12/2004	Novak et al.			
2005/0014119	A1	1/2005	Rudakov			
2005/0014531	A1	1/2005	Findikli			
2005/0014536	A1	1/2005	Grady			
2005/0015355	A1	1/2005	Heller et al.			
2005/0022212	A1	1/2005	Bowen			
2005/0149213	A1	7/2005	Guzak et al.			
2005/0181756	A1	8/2005	Lin			
2005/0207726	A1	9/2005	Chen			
2005/0239333	A1	10/2005	Watanabe et al.			
2005/0240705	A1	10/2005	Novotney et al.			
2005/0281185	A1	12/2005	Kawasaki			
2006/0031545	A1	2/2006	Manders et al.			
2006/0088228	A1	4/2006	Marriott et al.			
2006/0156415	A1	7/2006	Rubinstein et al.			
2006/0184456	A1	8/2006	de Janasz			
2006/0247851	A1	11/2006	Morris			
2006/0258289	A1	11/2006	Dua			
2006/0294209	A1	12/2006	Rosenbloom et al.			
2007/0056012	A1	3/2007	Kwon et al.			
2007/0056013	A1	3/2007	Duncan			
2007/0070856	A1	3/2007	Tebele			
2007/0080823	A1	4/2007	Fu et al.			
2007/0083814	A1	4/2007	Wilbrink et al.			
2007/0086724	A1	4/2007	Grady et al.			
2007/0173197	A1	7/2007	Hsiung			
2007/0173294	A1	7/2007	Hsiung			
2007/0206827	A1	9/2007	Tupman et al.			
2007/0226384	A1	9/2007	Robbin et al.			

FOREIGN PATENT DOCUMENTS

EP	1498899	A	1/2005
EP	1594319	A	11/2005
EP	1672613	A	6/2006
GB	2405718	A	3/2005
JP	07-176351		7/1995

OTHER PUBLICATIONS

Anonymous: "Introduction to Digital Audio," Microsoft Windows Hardware Developer Central Archive, [Online] Dec. 4, 2001 URL:<http://www.microsoft.com/whdc/archive/digitaudio.msp>> [retrieved on Jan. 15, 2008].

Anonymous: "Future of Digital Music in Windows," Microsoft Windows Hardware Developer Central Archive, [Online] Dec. 4, 2001 URL:<http://www.microsoft.com/whdc/archive/digitaudio.msp>> [retrieved on Jan. 15, 2008].

Anonymous; "Windows and Red Book Audio," Microsoft Windows Hardware Developer Central Archive, [Online] Dec. 4, 2001 URL:<http://www.microsoft.com/whdc/archive/Dmfuture.msp>> [retrieved Jan. 15, 2008].

"A Serial Bus on Speed Diagram: Getting Connected with FireWire," downloaded Oct. 16, 2001, PC Magazine: PC Tech (A, Serial Bus on Speed) wysiwyg://51<http://www.zdnet.com/pctech/content/18/10/tu1810.007.html>.

Belkin iPod Voice Recorder, Product Specification Sheet, printed Jun. 16, 2004.

Bindra, "Standard Turns Monitor into I/O Hub," Electronic Engineering Times, vol. 918, Sep. 6, 1996, p. 14.

Brentrup, "Introduction to Public Key Cryptography Demystified," Campus Technology, printed from <http://www.campus-technology.com/article.asp?id=7626> on Oct. 6, 2004.

"Cables to Go," downloaded Oct. 16, 2001 <http://www.cablestogo.com/product.asp?cat%5Fid=601&sku=27028>.

Crawford et al., "Sample rate conversion and bit rate reduction in the studio," IEE Colloquim on Digital Audio Signal Processing, May 22, 1991, pp. 8/1-8/3.

Derman, "Monitors Make Net Connections," Electronic Engineering Times, vol. 933, 1996, pp. 60 and 69.

"ExpressBus™ F5U010," User Guide Packing Checklist, Belkin Components Product Warranty.

"FireWire," downloaded Oct. 16, 2001, [si_wyg://_4_2/http://developer.apple.com/hardware/Fire_Wire](http://developer.apple.com/hardware/Fire_Wire).

"Fire Wire Connector", downloaded Oct. 16, 2001, wysiwyg://176/http://developer.apple.com/...es/Macintosh_CPUsG3/iBook/iBook-27.html.

US 7,529,872 B1

Page 4

- Fried, "FireWire poised to become ubiquitous," downloaded Oct. 16, 2001, CNET News.com, 1394 Trade Association: Press, wysiwyg:/132/http://_113_94ta.org/Press/200_1_ Press/august!8.2_7_.b.html.
- Fried, "New Fire Wire to blaze faster trail", downloaded Oct. 16, 2001, CNET News.com, <http://www.news.cnet.com/news/0-1006-200-6021210.html>.
- "How to Connect Your Computer, PC Hardware", downloaded Oct. 16, 2001, <http://www.scar.utoronto.ca/~ccweb/faculty/connect-howto.html>.
- "IEEE 1394/USB Comparison", downloaded Oct. 16, 2001, www.genitech.com.au/LIBRARY/TechSupport/infobits/firewireusb.htm.
- "Introduction to Public Key Cryptography," Oct. 9, 1998, printed from <http://developer.netscape.com/docs/manuals/security/pkin/contents.html> on Oct. 6, 2004.
- iPod Classic User's Guide, acquired from apple.com, 2002; 44 pages.
- iPod nano Features Guide, acquired from apple.com, 2008; 72 pages.
- iPod touch User's Guide, acquired from apple.com, 2008, 120 pages.
- "iPodDock/iPod Cradle," www.bookendzdocks.com/bookendz/dock_cradle.html, downloaded Feb. 27, 2003.
- Lambert, "Digital Audio Interfaces," Journal of the Audio Engineering Society, Audio Engineering Society, New York, NY, vol. 38, No. 9, (Sep. 1, 1990), pp. 681-684, 686, 68 XP000175146 ISSN: 1549-4950 figures 9, 10.
- Lewis, "On Technology." Fortune Magazine, Dec. 9, 2002.
- "Making UBS Work," downloaded Oct. 16, 2001, PC Magazine: PC Tech [wysiwyg:/155/http://www.zdnet.com/pcmag/pctech/content/18/04/tu1804.001.html](http://www.zdnet.com/pcmag/pctech/content/18/04/tu1804.001.html).
- Menezes et al., "Handbook of Applied Cryptography," Identification and Entity Authentication, pp. 385-424.
- Microsoft, "Media Transport Protocol Implementation Details," 2005, 18 pages.
- "MPV™ Music Profile Specification Revision 1.00" Internet Citation [online] (Jan. 7, 2004) URL:<http://www.osta.org/mpv/public/specs/MPVMusic-Prof-Spec-1.00.pdf> [retrieved Jun. 20, 2006] the whole document.
- Networking Tech Note, "1394 Standards and Specifications," 3 pgs.
- "Neuros MP3 Digital Audio Computer," www.neuroaudio.com, downloaded Apr. 9, 2003.
- "PMC FW2 IEEE1394 FireWire Controller," downloaded Oct. 16, 2001, <http://www.bvmltd.co.uk/PMCfw2ds.html>.
- Severance, "FireWire Finally Comes Home", Michigan State University, Standards, Nov. 1998, pp. 117-118.
- Sinitsyn, "Synchronization Framework For Personal Mobile Servers," Pervasive Computing and Communications Workshops (PERCOMW'04), Proceedings of the Second IEEE Annual Conference, Piscataway, NJ, USA, IEEE, Mar. 14, 2004, pp. 208-212.
- Teener, "Understanding Fire Wire: The IEEE 1394 Standards and Specifications", downloaded Oct. 16, 2001, wysiwyg:119/http://www.chipcenter.com/networking/ieee_1394/main.html.
- "The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition", Published by Standards Information Network, IEEE Press.
- "Universal serial bus specification—rev 2.0," XP002474828, Chapter 9, USB Device Framework, pp. 239-274.
- Vitaliano, "Why FireWire is Hot!Hot!Hot!" downloaded Oct. 16, 2001, "Impact.FireWire.SideBar" <http://www.vxm.com/21R.35.html>.
- Whittle, "Public Key Authentication Framework: Tutorial," First Principles Consulting, Jun. 2, 1996.

* cited by examiner

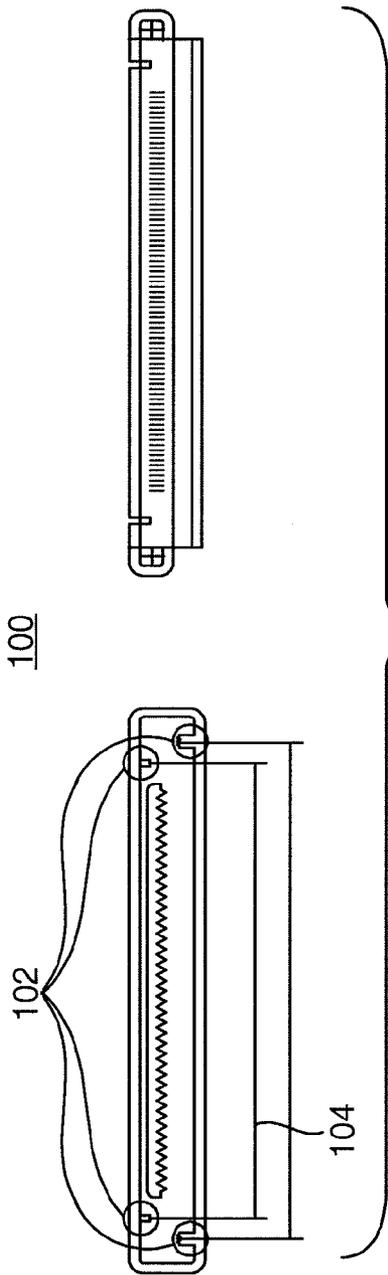


FIG. 1A

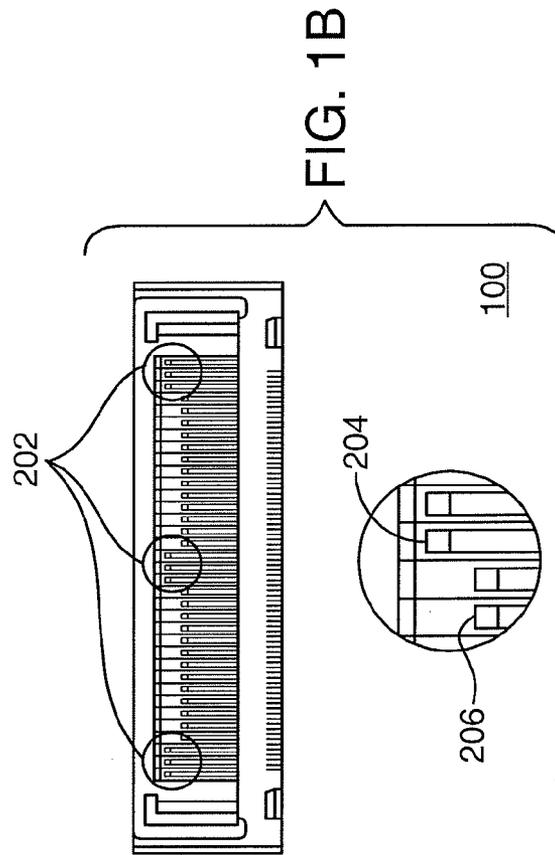


FIG. 1B

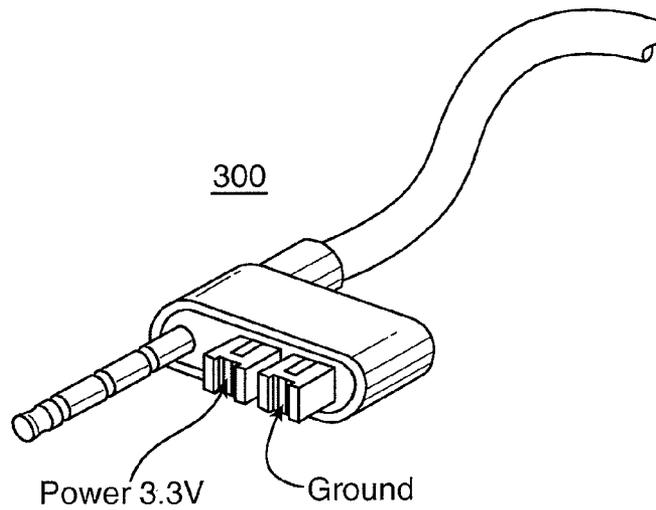
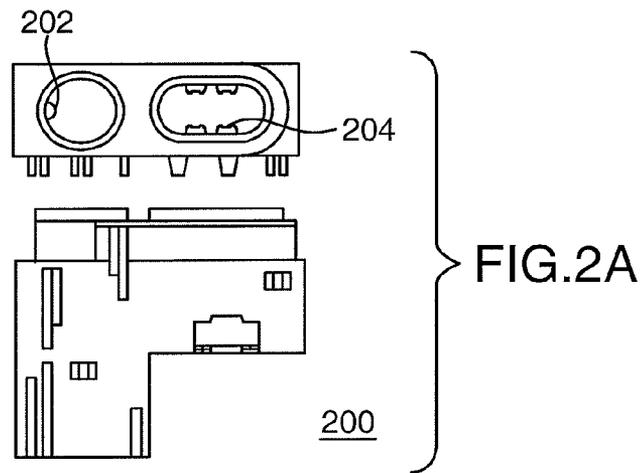


FIG. 2B

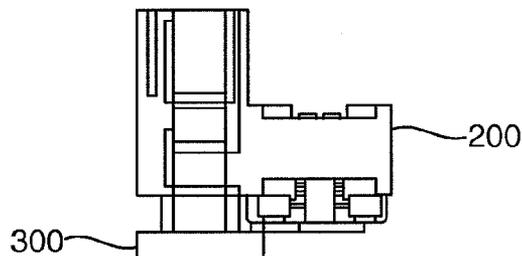


FIG. 2C

Pin	Signal Name	I/O	Function
1	DGND	I	Digital Ground
2	DGND	I	Digital 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. NOT for powering; only to detect USB host
9	TPB-	I/O	Firewire signal
10	Accessory Identify	I	Pull down in dock 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 output from iPod. Current limited to 100mA.
14	Reserved		
15	DGND	GND	Digital ground in iPod
16	DGND	GND	Digital ground in iPod
17	Reserved		
18	Dock Tx	I	Serial protocol (Data to iPod)
19	Dock Rx	O	Serial protocol (Data to iPod)
20	Accessory Detect	I/O	
21	S Video Y	O	Luminance Component
22	S Video C	O	Chrominance Component
23	Video Composite	O	Composite Signal
24	Remote Sense	I	Detect Remote
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 iPod
31	Chassis		Chassis ground for connector shell
32	Chassis		Chassis ground for connector shell

FIG. 3A

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	HP Detect	I	Internal Switch to detect 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	Video Signal
6	Accessory 3.3 V	O	3.3V Accessory power 100mA max
7	Tx	O	Serial protocol (Data from iPod to Device)
8	Rx	I	Serial protocol (Data to iPod from Device)
9	D GND	GND	Digital ground for accessory

FIG. 3B

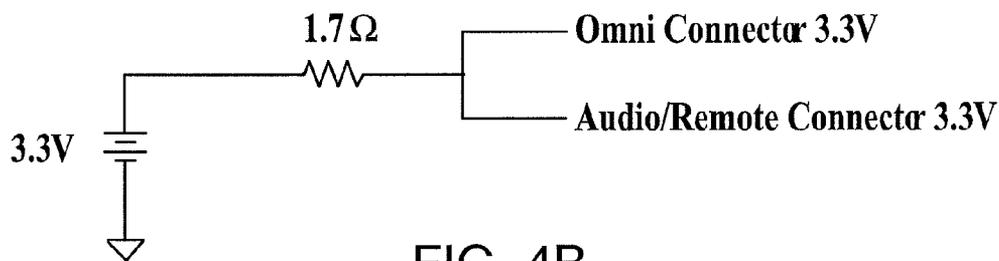


FIG. 4B

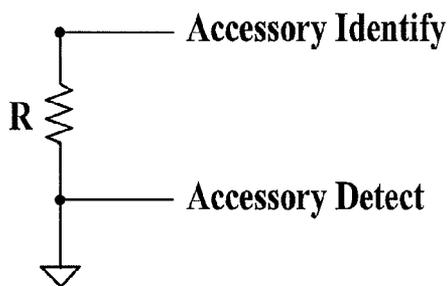


FIG. 4C

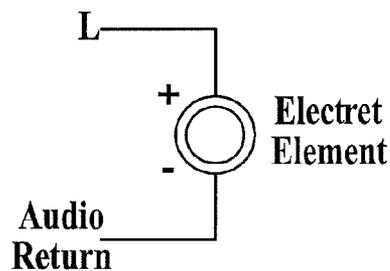


FIG. 4D

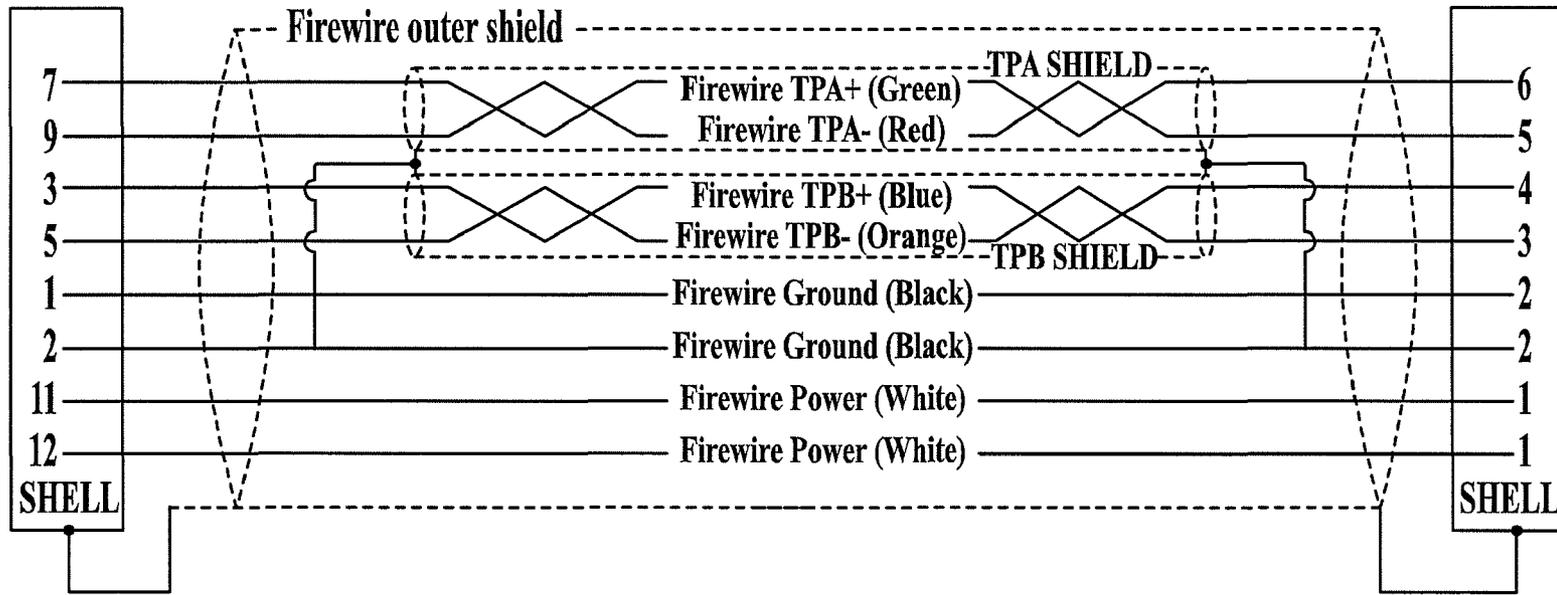


FIG. 4A

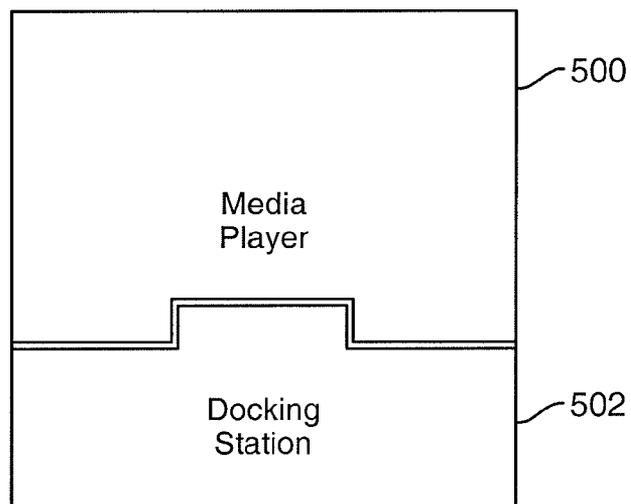


FIG. 5A

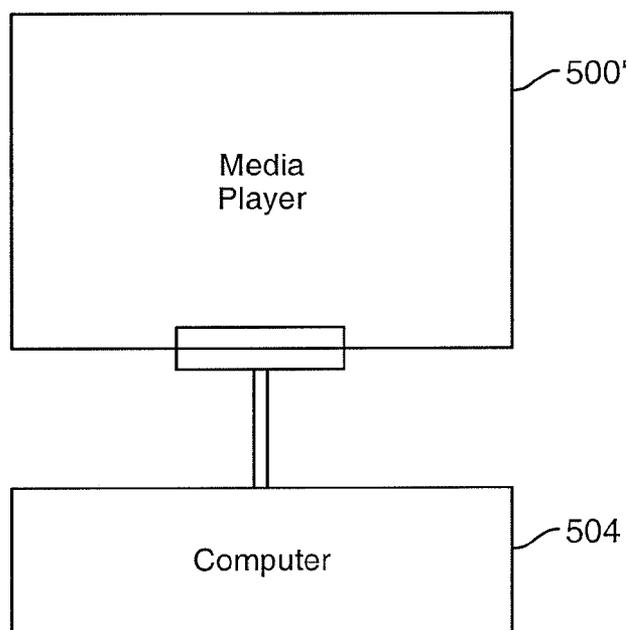


FIG. 5B

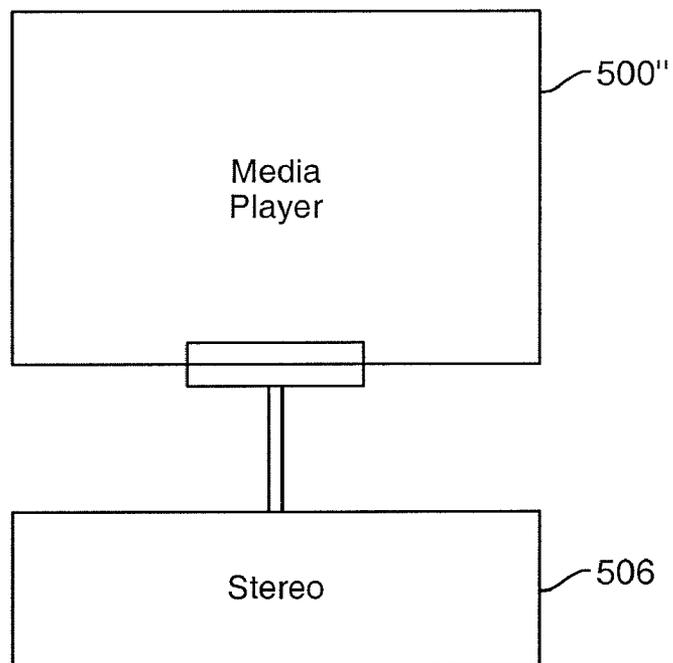


FIG. 5C

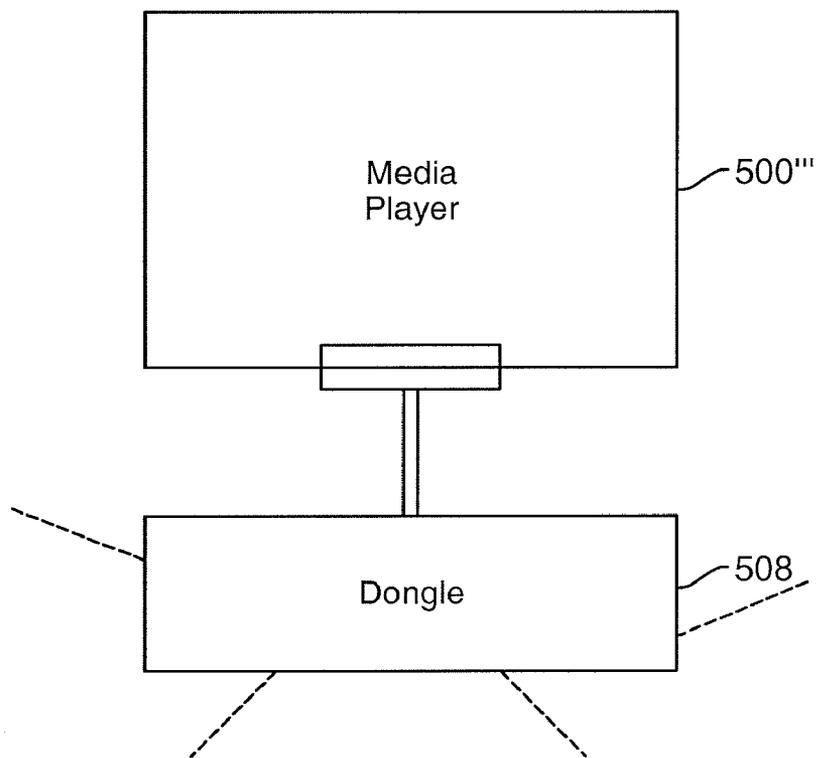


FIG. 5D

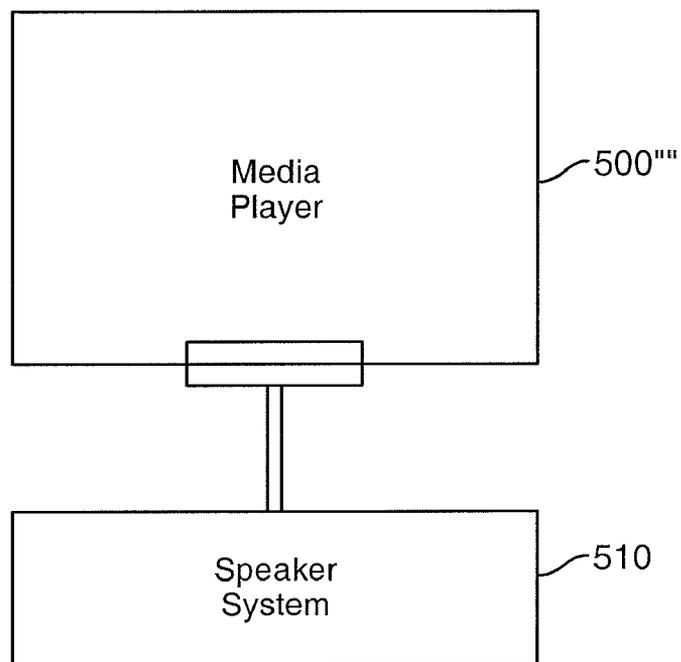


FIG. 5E

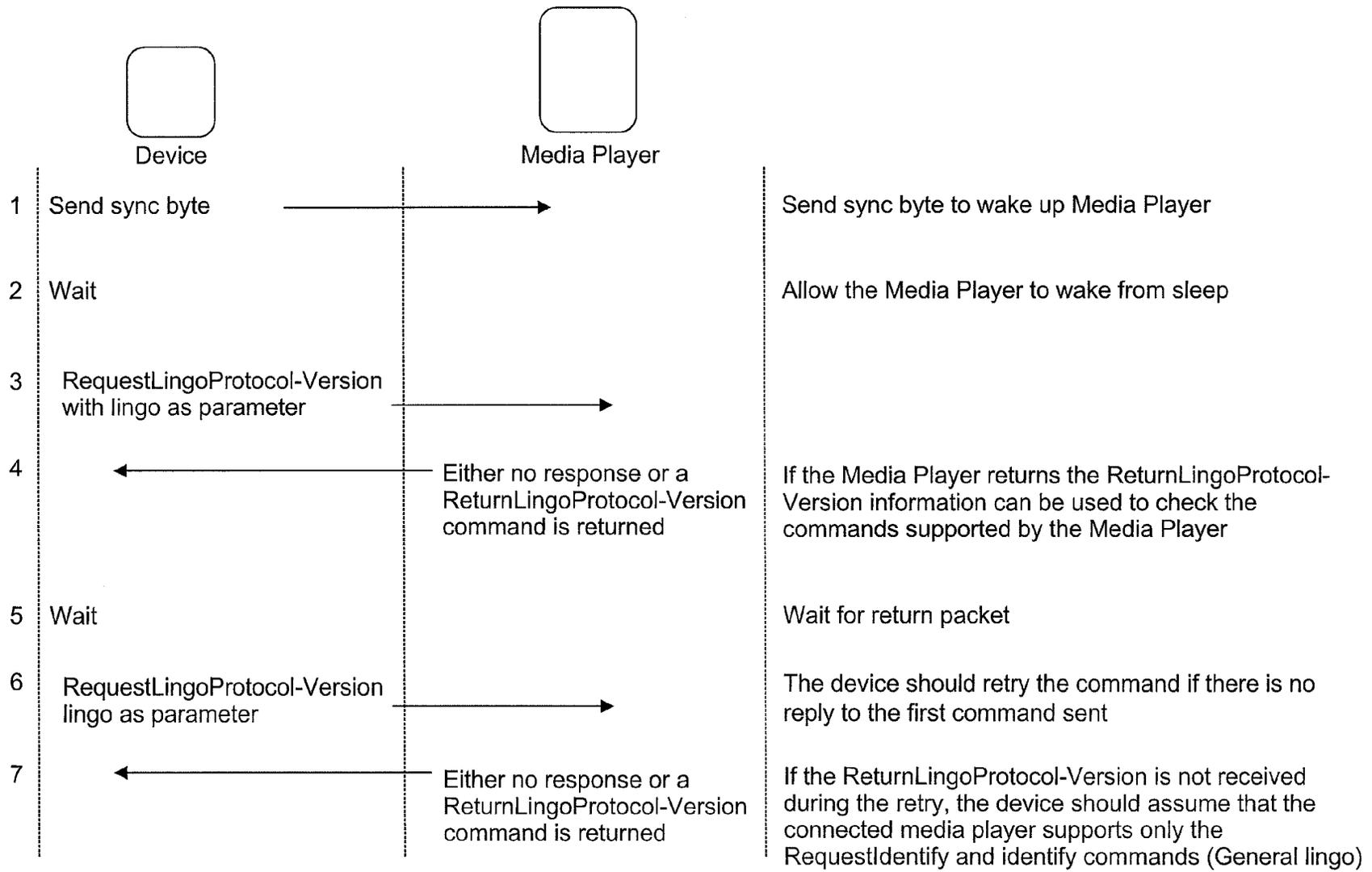


FIG. 6

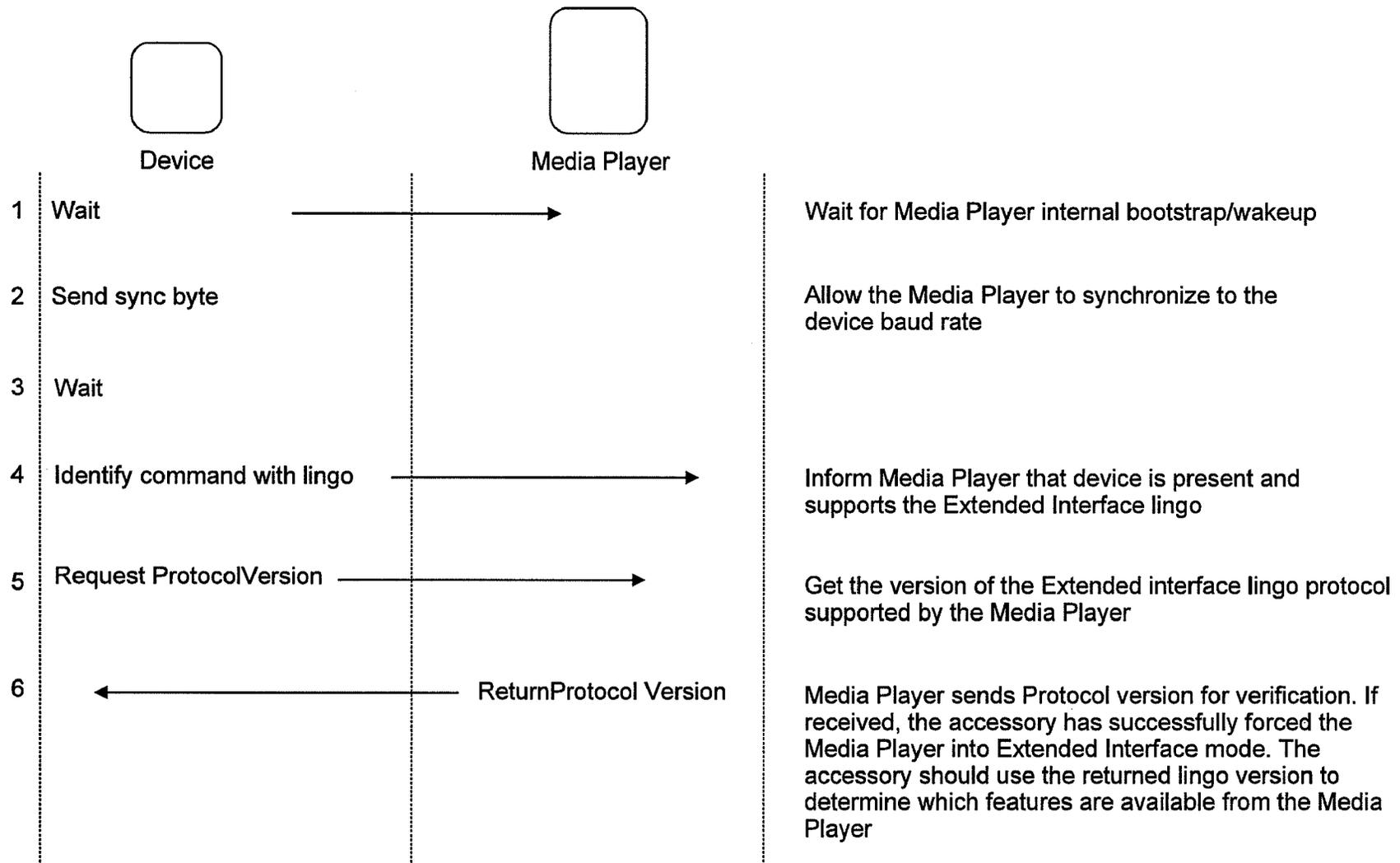


FIG. 7

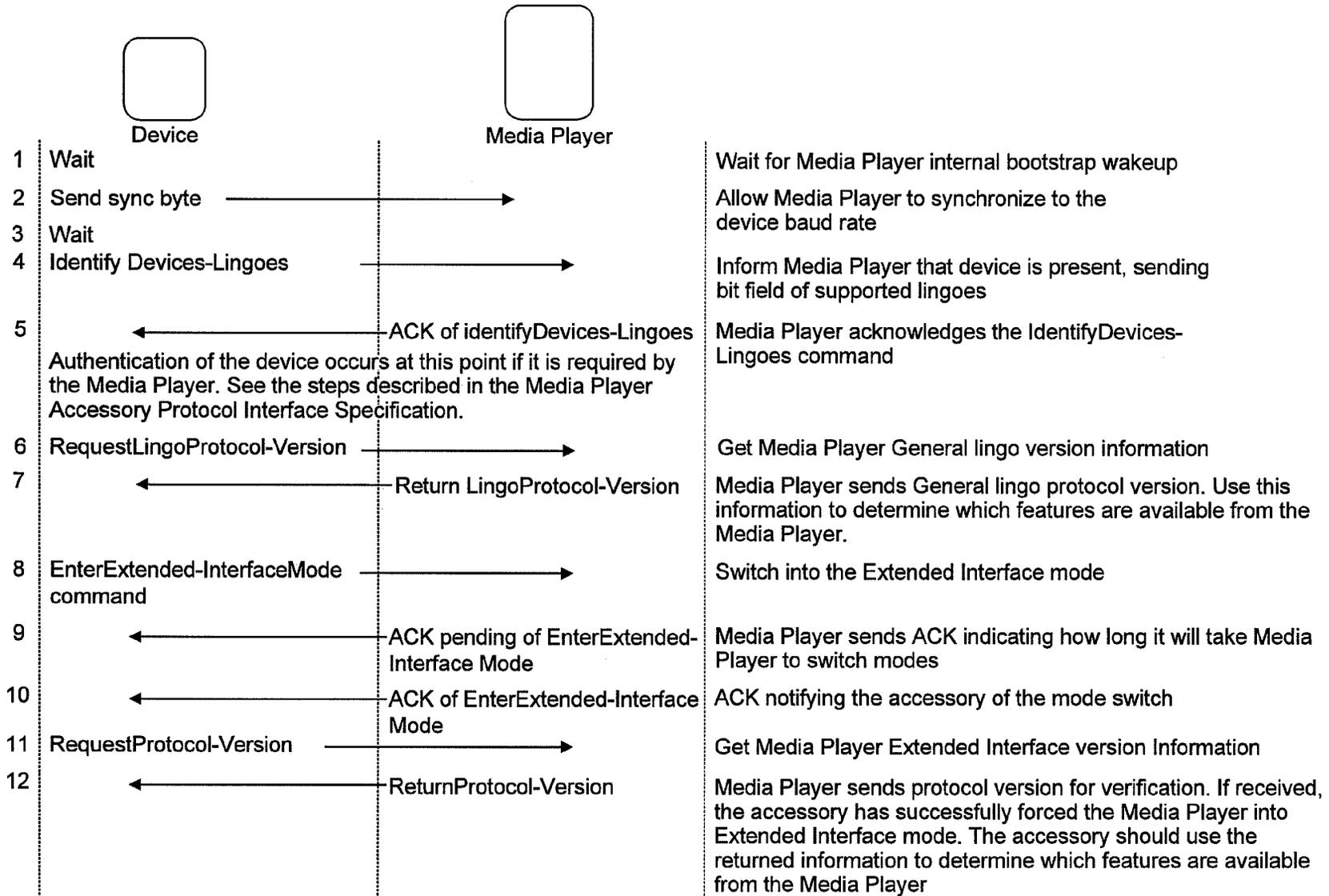


FIG. 8

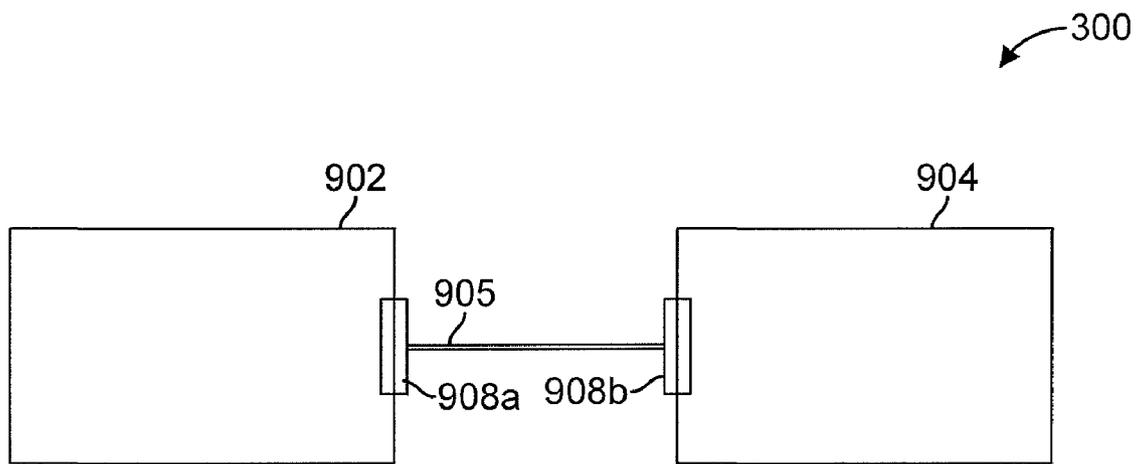


FIG. 9

US 7,529,872 B1

1

**COMMUNICATION BETWEEN AN
ACCESSORY AND A MEDIA PLAYER USING
A PROTOCOL WITH MULTIPLE LINGOES**

CROSS-REFERENCE TO RELATED
APPLICATIONS

This application is a continuation-in-part of U.S. application Ser. No. 10/833,689, filed Apr. 27, 2004, entitled "Connector Interface System from a Multi-Communication Device".

This application claims the benefit of: U.S. Provisional Patent Application No. 60/784,306, filed Mar. 20, 2006, entitled "Connector Interface System" which is hereby incorporated by reference.

FIELD OF THE INVENTION

The present invention relates generally to connector interfaces and more particularly to a connector interface system which is utilized in conjunction with media players and their accessories.

BACKGROUND OF THE INVENTION

A media player stores media assets, such as audio tracks or photos, that can be played or displayed on the media player. One example of a media player is the iPod® media player, which is available from Apple Computer, Inc., of Cupertino, Calif. Often, a media player acquires its media assets from a host computer that serves to enable a user to manage media assets. As an example, the host computer can execute a media management application to manage media assets. One example of a media management application is iTunes®, version 6.0, produced by Apple Computer, Inc.

A media player typically includes one or more connectors or ports that can be used to interface to the media player. For example, the connector or port can enable the media player to couple to a host computer, be inserted into a docking system, or receive an accessory device. There are today many different types of accessory devices that can interconnect to the media player. For example, a remote control can be connected to the connector or port to allow the user to remotely control the media player. As another example, an automobile can include a connector and the media player can be inserted onto the connector such that an automobile media system can interact with the media player, thereby allowing the media content on the media player to be played within the automobile.

Currently, the connectors or ports of a media player are open for use so long as a compatible connector or port is utilized. Consequently, numerous third-parties have developed accessory devices for use with other manufacturers' media players.

In a typical connector interface, there is a docking connector that allows for the docking of the media player device to a docking station for another type of communication for the device. A media player also typically includes a remote connector with the ability to output audio. As more multi-media content becomes available (i.e., digital, video graphics, etc.) it is desirable to have a media player which can effectively input and output such data.

Finally, a media player must be able to identify a particular device's functionality to which it is associated. Heretofore, there is no device that includes features that overcome many of the above-stated problems. What is desired is a connector

2

interface system which is utilized in such a device to address all the above-identified issues. The present invention addresses such a need.

SUMMARY OF THE INVENTION

A connector interface system is disclosed. The connector interface system includes an interface and a protocol in communication with the interface which allows a media player to communicate with external devices over a transport link. The protocol includes a core protocol functionality and a plurality of accessory lingoos. The accessory lingoos comprise a microphone lingo, a simple remote lingo, a display remote lingo, a RF transmitter lingo and an extended interface lingo.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGS. 1A and 1B illustrate a docking connector in accordance with the present invention.

FIGS. 2A-2C illustrate the remote connector in accordance with the present invention.

FIG. 3A illustrates the connector pin designations for the docking connector.

FIG. 3B illustrates the connection pin designations for the remote connector.

FIG. 4A illustrates the Firewire connector interface.

FIG. 4B illustrates the USB connector interface.

FIG. 4C illustrates a reference schematic diagram for accessory detect and identify system for detecting and identifying accessories for the docking connector.

FIG. 4D is a reference schematic of an electret microphone that is within the remote connector.

FIGS. 5A-5E illustrate a media player coupled to a plurality of accessories.

FIG. 6 illustrates the sequence of events that can be utilized to test for the full set of general lingo commands.

FIG. 7 illustrates a single lingo accessory command flow.

FIG. 8 illustrates a multiple lingo command flow.

FIG. 9 illustrates an exemplary system using the connector interface system described herein.

DETAILED DESCRIPTION

Definitions

Device

An external electronic component connected to the media player using the 30-pin connector or the audio/remote connector.

HID

Human Interface Device. HID is a standard USB class of functionality and interface. 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 data to the HID interface of the media player or from the media player to the host. MPAP packets are broken into HID reports before being sent across the transport link and are reassembled on the receiving side.

US 7,529,872 B1

3

MPUI

Media player USB Interface. This is a configuration of the media player when attached as a device over USB. This configuration allows the media player to be controlled using MPAP, using a USB Human Interface Device (HID) interface as a transport mechanism.

Lingo

The command category used by a device. There is a general lingo that must be supported by all devices. Other lingo is designed for use by specific devices.

Link

The logical connection between an external device and the media player via serial port or other physical connection.

The present invention relates generally to media players and more particularly to a connector interface system for such devices. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein.

Connector Interface System Overview

To describe the features of the connector interface system in accordance with the present invention in more detail, refer now to the following description in conjunction with the accompanying drawings.

Docking Connector

FIGS. 1A and 1B illustrate a docking connector **100** in accordance with the present invention. Referring first to FIG. 1A, the keying features **102** are of a custom length **104**. In addition, a specific key arrangement is used where one set of keys are separated by one length and another set of keys are separated by another length at the top of the connector. The use of this key arrangement prevents noncompliant connectors from being plugged in and potentially causing damage to the device. The connector for power utilizes a Firewire specification for power. The connector includes a first make/last break contact to implement this scheme. FIG. 1B illustrates the first make/last break contact **202** and also illustrates a ground pin and a power pin related to providing an appropriate first make/last break contact. In this example, the ground pin **204** is longer than the power pin **206**. Therefore, the ground pin **204** would contact its mating pin in the docking accessory before the power pin **206**. Therefore internal electrical damage of the electronics of the device is minimized.

In addition, a connector interface system in accordance with the present invention uses both USB and Firewire interfaces as part of the same docking connector alignment, thereby making the design more compatible with different types of interfaces, as will be discussed in detail hereinafter. In so doing, more remote systems and devices can interface with the multi-communication device.

Remote Connector

The connection interface system also includes a remote connector which provides for the ability to output audio and, input audio, provides I/O serial protocol, and an output video. FIG. 2A is a front and top view of a remote connector **200** in accordance with the present invention. As is seen, the remote connector **200** includes a top headphone receptacle **202**, as well as a second receptacle **204** for remote devices. FIG. 2B illustrates a plug **300** to be utilized in the remote connector.

4

The plug **300** allows the features to be provided via the remote connector. FIG. 2C illustrates the plug **300** inserted into the remote connector **200**. Heretofore, all these features have not been implemented in a remote connector. Therefore, a standard headphone cable can be plugged in but also special remote control cables, microphone cables, or video cables could be utilized with the remote connector.

To describe the features of the connector interface system in more detail, please find below a functional description of the docking connector, remote connector and a protocol in accordance with the present invention.

Docking and Remote Connector Specifications

For an example of the connector pin designations for both the docking connector and the remote connector for a multi-communication device such as a media player device by Apple Computer, Inc., refer now to FIGS. 3A and 3B. FIG. 3A illustrates the connector pin designations for the docking connector. FIG. 3B illustrates the connection pin designations for the remote connector.

Docking Connector Specifications

FIG. 4A illustrates a typical FireWire connector interface for the docking connector.

FireWire Power has the following characteristics: 8V-30V DC IN, 10 W Max. FireWire communication is designed to IEEE 1394 A Spec (400 Mb/s).

USB Interface

The media player provides two configurations, or modes, of USB device operation: mass storage and Media Player USB Interface (MPUI). The MPUI allows the media player to be controlled using a Media Player Accessory Protocol (MPAP) which will be described in detail herein, using a USB Human Interface Device (HID) interface as a transport mechanism.

Accessory 3.3 V Power

FIG. 4B illustrates the USB connector interface. The media player accessory power pin supplies voltages, for example, 3.0 V to 3.3V+/-5% (2.85 V to 3.465 V) over the 30-pin connector. A maximum current is shared between the 30-pin and Audio/Remote connectors.

By default, the media player supplies a particular current such as 5 mA. Proper software accessory detect is required to turn on high power (for example, up to 100 mA) during active device usage. When devices are inactive, they must consume less than 5 mA current.

Accessory power is switched off for a period of, for example, approximately 2 seconds during the media player bootstrap process. This is done to ensure that accessories are in a known state and can be properly detected. All accessories are responsible for re-identifying themselves after the media player completes the bootstrap process and transitions accessory power from the off to the on state.

Accessory power is grounded through the DGND pins.

FIG. 4C illustrates a reference schematic diagram for accessory detect and identify system for detecting and identifying accessories for the docking connector. The system comprises a resistor to ground that allows the device to determine what has been plugged into docking connector. There is an internal pullup on Accessory Identify. Two pins are required (Accessory Identify & Accessory Detect)

FIG. 4D is a reference schematic of an electret microphone that is within the remote connector.

Serial Protocol Communication is provided using two pins to communicate to and from device (Rx & Tx). Input and Output levels are 0V=Low, 3.3V=High.

US 7,529,872 B1

5

Media Player Accessory Protocol (MPAP)

Media players connect to a variety of accessories. FIGS. 5A-5E illustrates a media player 500 coupled to different accessories. FIG. 5A illustrates a media player 500 coupled to a docking station 502. FIG. 5B illustrates the media player 500' coupled to a computer 504. FIG. 5C illustrates the media player 500" coupled to a car or home stereo system 506. FIG. 5D illustrates the media player 500" coupled to a dongle 508 that communicates wirelessly with other devices. FIG. 5E illustrates the media player 500"" coupled to a speaker system 510. As is seen, what is meant by accessories includes but is not limited to docking stations, chargers, car stereos, microphones, home stereos, computers, speakers, and devices which communicate wirelessly with other devices.

Accordingly a single media player must be able to communicate with different devices having varying functionality. To allow for this communication to be handled efficiently in a system and method in accordance with the present invention, a protocol is provided. As part of the connector interface system and in conjunction with the connectors, the protocol allows for the media player to identify the type of accessory that the media player is connected to and also allows the media player to identify the functionality of the accessory. In so doing the media player can efficiently and effectively interact with the accessory in the appropriate manner.

The Media Player Accessory Protocol (MPAP) allows the media player to communicate with a functional range of external devices. The protocol can be broken into three logical components: the protocol transport link, the protocol core, and the individual accessory lingoes. Devices can use USB and UART serial interfaces as channels to transfer MPAP packets. The protocol core describes the components necessary for communication with all external devices. It includes the basic packet definition and the general lingo, which allows for accessory identification, authentication, and retrieval of media player information. The accessory lingoes comprise the individual dialect commands. Each accessory lingo corresponds to a functional class of external devices.

Every external device must support a protocol transport link, the protocol core, and one or more lingoes, as required for its function. For example, the media player standard in-line remote control is a UART serial device that uses the general lingo and the simple remote lingo. To describe these features in more detail refer now to the following.

Protocol Transport Links

Accessories may communicate with the media player, using MPAP over the serial port link or the USB port link. Those links are described below.

UART Serial Port Link

Accessories using the Media Player Accessory Protocol (MPAP) over the UART serial port link use two pins, RX and TX, to communicate to and from the media player.

A device coupled to the docking connector allows for a standard serial protocol to be utilized. Attaching a serial dock accessory makes any top-attached (remote connector) accessories inactive.

The MPAP builds upon an existing serial specification such as the RS-232 serial specification. However, the signaling levels are non-standard. The RS-232 specification states that a mark is -7V and a space is +7V. In MPAP protocol, for example, a mark may be 2.85 V through 3.465 V and a space is 0 V through 0.9 V.

USB Port Link

The media player, for example, may be a USB 2.0-compliant device that supports two mutually exclusive modes of operation:

6

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 synching music and content, transferring files, and so forth.

MPAP enabled device. This is the configuration needed to support MPAP using the Media Player USB Interface (MPUI). This mode must be selected by the USB host before it can be used.

These two mutually exclusive modes of operation are each represented by a USB configuration. When the media player 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.

Accessory Identify Resistor and MPUI

There are two types of media player accessories: serial accessories that communicate with the media player using MPAP and resistor-based accessories that need access to specific media player behaviors.

Resistor-based accessories use an Accessory Identify resistor (RID) to get access to a specific media player behavior. These devices tend to be simple accessories, such as battery packs and car chargers, and have one specific purpose. When attached, these accessories unlock media player features based on the RID used. Accessories that communicate with the media player using MPAP are serial accessories. Serial-based accessories may use the other RID values if they are to be utilized for unlocking a particular resistor-based behavior.

The presence of a RID of one resistance value on USB attachment triggers the media player to present the MPUI configuration as the first, or default, configuration. The second configuration becomes media player as a Mass Storage class disk device.

For instance, connecting a media player to a standard host, such as a PC or a Macintosh, along with the RID causes the media player to be set up as a HID device (the HID interface being a component of the MPUI configuration) and not as a disk device. With no resistor, the media player is set up as a disk device.

It is possible to create an accessory that supports both MPAP over USB and the older UART serial-only media players using the same connector.

Media Player USB Interface (MPUI) Configuration

The MPUI configuration allows the media player to communicate using MPAP over USB. The USB Human Interface Device (HID) interface is the transport link and uses two endpoints for communication: the control endpoint is used for OUT data, while the HID interrupt endpoint is used for IN data.

The media player HID interface utilizes several vendor-specific HID reports, some of which are used to transport data from the host and some of which are used to transport data to the host. The HID report sizes range from a few bytes to several hundred KB and each one has a unique identifier. In order to send data to the media player, a host chooses one or more appropriately-sized HID reports in which to embed the MPAP packet and sends this to the media player HID interface with USB Set-Report command. The media player reassembles the MPAP packet and processes it. The process is repeated in reverse when the media player sends responses or MPAP packets to the host.

HID as a Transport

As mentioned earlier, the HID interface breaks MPAP 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 MPAP packets. For instance, a set could be a single HID report containing one MPAP packet or a set of HID reports containing a total of 3 MPAP packets.

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 media player specifies several different report types. The USB host should analyze the HID report descriptor of the media player at runtime to determine which Report ID corresponds to the most appropriate report type for each transfer.

A representative group of lingo codes for the media player are listed below.

Lingo Specifications

Lingo

General

Microphone

Simple Remote

Display Remote

Extended Interface

RF transmitter

The general lingo is shared for housekeeping commands across all devices. In addition to the general lingo, an accessory will implement a function specific lingo. The microphone lingo is used by the remote connector on the multi-communication device. The simple remote lingo is used by a standard in-line remote control. The display remote lingo is reserved for a device with similar functionality to the standard remote but with a display for status. The RF transmitter lingo is used for devices that transmit the device analog audio over radio frequencies. To describe the lingo codes in more detail, refer now to the following discussion in conjunction with the accompanying figures.

General Lingo Command Summary

Table 1 provides a summary of all commands in the general lingo, including whether or not device authentication is required for use of the command.

TABLE 1

Command	Authentication Required
Request Identify	No
RequestMediaPlayerName	No
RequestMediaPlayerSoftwareVersion	No
RequestMediaPlayerSerialNum	No
RequestMediaPlayerModelNum	No
RequestLingoProtocolVersion	No
Identify	No
IdentifyAccessoryLingo	No
ACK	No
AckDevAuthenticationInfo	No
AckDevAuthenticationStatus	No
AckMediaPlayerAuthenticationInfo	No
AckMediaPlayerAuthenticationStatus	N/A
ReturnMediaPlayerName	No
ReturnMediaPlayerSoftwareVersion	No
ReturnMediaPlayerSerialNumber	No
ReturnLingoProtocolVersion	No
RetDevAuthenticationInfo	No
RetDevAuthenticationSignature	No
RetMediaPlayerAuthenticationInfo	No
RetMediaPlayerAuthenticationSignature	No
GetDevAuthenticationInfo	No
GetDevAuthenticationSignature	No
GetMediaPlayerAuthenticationInfo	No

TABLE 1-continued

Command	Authentication Required
GetMediaPlayerAuthenticationSignature	No
NotifyMediaPlayerStateChange	No

To identify itself, an accessory should send either an Identify command or an IdentifyAccessoryLingo command. Accessory devices that support more than one lingo, not including the general lingo, should use the IdentifyAccessoryLingo command.

The media player may send a RequestIdentify command to the device to ask to re-identify itself.

The remaining general lingo commands can be used to obtain general information from the media player. These commands allow the device to request the name, serial number, model number, and software version number of the media player. The RequestLingoProtocolVersion command allows a device to query the media player for the lingo protocol versions of all supported lingo codes on the media player. The ACK command is used by the media player 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.

An accessory may send a RequestLingoProtocolVersion command, to determine which features the connected media player supports. If the media player does not respond to the RequestLingoProtocolVersion command, the device should try the command once more. If no response is received the second time, the device should assume that the connected media player supports the only RequestIdentify and Identify General Lingo commands.

FIG. 6 shows the sequence of the events that can be used to test for the full set of general lingo commands. As is seen, first a sync byte is sent to wake up the media player, then allow the media player allowed to wake from sleep. If the media player returns the ReturnLingoProtocolVersion command, then the general lingo version information can be used to check the commands supported by the media player. Next, wait for return packet and the device should retry the command if there is no reply to the first command send. If the ReturnLingoProtocolVersion command is not received during the retry, the device should assume that the connected media player supports only the RequestIdentify and Identify commands. This assumes the accessory device is connected and has already been through the identification process.

General Lingo Command Details

This section describes the general lingo commands and their packet formats.

Request Identify Command

Direction: media player→accessory

The media player sends this command to prompt accessories to re-identify themselves. If an accessory receives this command, it should respond with either the IdentifyAccessoryLingo command or the Identify command. Accessories that require authentication or use the USB transport link, or support multiple lingo codes should respond using the IdentifyAccessoryLingo command.

Identify Command

Direction: accessory→media player

The accessory must send this command to notify the media player that an accessory has been attached and to register the

US 7,529,872 B1

9

lingo it supports. Accessories should identify at boot time and any time they receive a RequestIdentify command from the media player.

The Identify command has facilities for RF Transmitter devices to draw more than a predetermined amount of power (such as 4 mA) from the media player.

ACK Command

Direction: media player→accessory

The media player sends the ACK command to notify the device of command completion status and errors.

RequestMediaPlayerSoftwareVersion Command

Direction: media player→accessory

Retrieves the software version for the media player. The media player responds with a ReturnMediaPlayerSoftwareVersion command containing the major, minor, and revision version numbers.

ReturnMediaPlayerSoftwareVersion Command

Direction: media player→accessory

The media player sends this command in response to the RequestMediaPlayerSoftwareVersion command message from the accessory. The media player returns each version number as an individual byte, with the major version number sent first.

RequestMediaPlayerName Command

Direction: accessory→media player

Retrieves the name of the media player. The media player responds with a ReturnMediaPlayerName command containing the name of the media player.

ReturnMediaPlayerName Command

Direction: media player→accessory

The media player sends this command in response to the RequestMediaPlayerName command message from the accessory.

RequestMediaPlayerSerialNum Command

Direction: accessory→media player

Retrieves the serial number string of the media player. The media player responds with a ReturnMediaPlayerSerialNum command containing the serial number.

ReturnMediaPlayerSerialNum Command

Direction: media player→accessory

The media player sends this command in response to the RequestMediaPlayerSerialNumber command message from the accessory.

RequestMediaPlayerModelNum Command

Direction: accessory→media player

Retrieves information for the media player. The media player responds with a ReturnMediaPlayerModelNum command containing the model number of the media player.

ReturnMediaPlayerModelNum Command

Direction: media player→accessory

The media player sends this command in response to the RequestMediaPlayerModelNum command message from the accessory.

IdentifyAccessoryLingoes Command

Direction: accessory→media player

The accessory sends this command to signal its presence and to identify its supported lingoes. In response, the media player sends an ACK command. The IdentifyAccessoryLingoes command is used by multi-lingo accessories to report all supported lingoes and should be used in place of the Identify command.

GetDevAuthenticationInfo Command

Direction: media player→accessory

The media player sends this command to obtain authentication information from the accessory. The command is sent if and only if the accessory has indicated that it supports authentication in its IdentifyAccessoryLingoes Options bits

10

and has passed a valid, non-zero accessory ID. In response, the accessory sends a RetDevAuthenticationInfo command.

RetDevAuthenticationInfo Command

Direction: accessory→media player

The accessory indicates the MPAP authentication version that it supports by returning this command in response to a GetDevAuthenticationInfo command from the media player.

AckDevAuthenticationInfo Command

Direction: media player→accessory

The media player sends this command in response to RetDevAuthenticationInfo command. It indicates the current state of the accessory authentication information.

RequestLingoProtocolVersion Command

Direction: accessory→media player

Retrieves version information for any of the lingoes supported by the media player. The media player responds with a ReturnLingoProtocolVersion command containing the major and minor version information of the requested media player lingo.

ReturnLingoProtocolVersion Command

Direction: media player→accessory

The media player sends this command in response to the RequestLingoProtocolVersion command message from the accessory. The major and minor version information for the requested lingo are returned.

GetDevAuthenticationSignature Command

Direction: media player→accessory

The media player sends this command to authenticate an accessory that has identified itself as requiring authentication. Authentication occurs either immediately upon identification or when the accessory attempts to use a restricted lingo or command. The accessory calculates its digital signature based on the challenge offered by the media player and sends the results back to the media player using a RetDevAuthenticationSignature command.

If the returned signature cannot be verified, the media player responds with a non-zero AckDevAuthenticationStatus command, followed immediately by another GetDevAuthenticationSignature command.

RetDevAuthenticationSignature Command

Direction: accessory→media player

The accessory sends this command to the media player in response to GetDevAuthenticationSignature command. The media player verifies the digital signature, calculated by the accessory based on the offered challenge. If verification passes, the media player authenticates the accessory and updates its lingo and command access permissions accordingly. The authentication status is sent to the accessory using AckDevAuthenticationStatus command.

AckDevAuthenticationStatus Command

Direction: media player→accessory

The media player sends this command to the accessory in response to the RetDevAuthenticationSignature command. It indicates the current accessory authentication state. If the accessory receives a non-zero status, the accessory has failed authentication and will only be able to use unauthenticated lingo commands.

If the accessory receives a zero status, the media player has successfully authenticated the accessory. The accessory may then use the requested authenticated lingoes and commands. Optionally, the accessory may begin the process of authenticating the media player, by sending a GetMediaPlayerAuthenticationSignature command.

GetMediaPlayerAuthenticationSignature Command

Direction: accessory→media player

The accessory uses this command to send an offered challenge to the media player for digital signature. In response,

US 7,529,872 B1

11

the media player returns its signed challenge to the accessory using RetMediaPlayerAuthenticationSignature command. Accessories should implement the authentication retry feature described in GetDevAuthenticationSignature command.

RetMediaPlayerAuthenticationSignature Command

Direction: media player→accessory

The media player sends this command to the accessory in response to a Get MediaPlayerAuthenticationSignature command. The accessory verifies the digital signature, calculated by the media player based on the offered challenge, and, if verification passes, authenticates the media player. The accessory sends the authentication status to the media player.

AckMediaPlayerAuthenticationStatus Command

Direction: accessory→media player

The accessory sends this command to the media player in response to RetMediaPlayerAuthenticationSignature command. It indicates the current media player authentication state. The accessory should return a non-zero ACK for each failed authentication attempt.

GetMediaPlayerAuthenticationInfo Command

Direction: accessory→media player

The accessory sends this command to obtain authentication information from the media player. The media player should send this command only if the accessory has indicated that it supports authentication in its IdentifyAccessoryLingoes command options bits and the media player has successfully completed the authentication process. In response, the media player sends RetMediaPlayerAuthenticationInfo command.

AckMediaPlayerAuthenticationInfo Command

Direction: accessory→media player

The accessory sends this command to the media player in response to RetMediaPlayerAuthenticationInfo command. It indicates the current state of the media player information version. If the accessory sends a non-zero status, it indicates that it will not be able to authenticate the media player due to a protocol version or authentication index mismatch.

NotifyMediaPlayerStateChange Command

Direction: media player→accessory

The media player sends this notification command when the media player state is about to change to accessories that identify using IdentifyAccessoryLingoes command. If the accessory identifies using Identify command, this notification is not sent. The state change byte indicates the specific media player state transition. If the media player is switching from a power on state to a light sleep state, accessories must immediately reduce their power consumption below the maximum current such as 5 mA. When the media player has transitioned to a deep sleep or hibernate state, self-powered accessories are expected to automatically re-identify themselves when accessory power is restored.

Microphone Lingo Command

The microphone lingo enables combination microphone and speaker accessory devices to record and playback audio. Media player mass storage disk capacities enable the option of supporting a stereo input mode and higher audio sample rates. Media players may therefore be used for high-quality mobile audio recording.

When the media player detects a device speaking the microphone lingo, it may transition into a recorder application where it can create and manage recordings. Based on the microphone device capabilities, the media player recording application may choose to change its appearance based on the presence or absence of certain microphone features. The device should indicate its capabilities to the media player on request. These capabilities may include:

12

Stereo line input source

Stereo/mono control

Recording level control

Recording level limiter

Microphone accessories can draw power from the media player or supply power to the media player. Accessory device power management is important as media players transition to a smaller physical size at the same time as trying to extend battery life. As an accessory using the microphone lingo, the microphone accessory will be notified of media player state changes, such as transitioning to the power on, light sleep, hibernate, and deep sleep states. Accessory power is in low mode by default and is raised to high power mode only during recording and playback states.

The microphone accessory is responsible for keeping the power consumption below the maximum allowed limits for each media player state. Note that accessory power is completely shut off when the media player enters the hibernate and deep sleep states. When waking from a light sleep state, the microphone accessory is required to re-identify and re-authenticate itself, as with other accessories using authenticated lingoes or commands. On reset or power up, the accessory device should be in low power state (for example, consuming less than 5 mA) with the amplifier off (audio input and output disabled).

Microphone state information is typically retained locally by the accessory player while uninterrupted accessory power (either high or low power) is available. If accessory power is turned off, device state information may be lost. Accessories are not expected to retain state information across accessory power down cycles (hibernate or deep sleep modes).

Media player playback volume level changes may require the accessory to support display remote lingo functionality.

Table 2 lists the commands available as part of the microphone lingo.

TABLE 2

Command	Connector	Authentication Required
BeginRecord	Audio/Remote	No
EndRecord	Audio/Remote	No
BeginPlayback	Audio/Remote	No
EndPlayback	Audio/Remote	No
ACK	30-pin	Yes
GetDevAck	30-pin	Yes
MediaPlayerModeChange	30-pin	Yes
GetDevCaps	30-pin	Yes
RetDevCaps	30-pin	Yes
GetDevCtrl	30-pin	Yes
RetDevCtrl	30-pin	Yes
SetDevCtrl	30-pin	Yes

BeginRecord Command

Direction: media player→accessory

The media player sends this command to notify the accessory that audio recording has started. The accessory does not return a packet to the media player in response to this command.

EndRecord Command

Direction: media player→accessory

The media player sends this command to notify the accessory that audio recording has ended. The accessory does not return a packet to the media player in response to this command.

US 7,529,872 B1

13

BeginPlayback Command

Direction: media player→accessory

The media player sends this command to notify the accessory that audio playback has started. The accessory does not return a packet to the media player in response to this command.

EndPlayback Command

Direction: media player→accessory

The media player sends this command to notify the accessory that audio playback has ended. The accessory does not return a packet to the media player in response to this command.

ACK Command

Direction: accessory→media player

The microphone accessory sends this command in response to a command sent from the media player. Note that some commands do not require an ACK response. The accessory sends an ACK response when a command that does not return any data has completed, a bad parameter is received, or an unsupported or invalid command is received.

GetDevAck Command

Direction: media player→accessory

The media player sends this command to get an ACK response from a microphone accessory. The media player uses this command to “ping” the accessory and determine that it is present and ready to accept commands. In response, the accessory sends the ACK command with command status OK.

MediaPlayerModeChange Command

Direction: media player→accessory

The media player sends this command to the microphone accessory when an audio recording or playback event occurs. The microphone accessory uses the MediaPlayerModeChange command to configure its inputs or outputs and power consumption level for the specified mode. In response, the accessory sends the ACK command with the command status OK. The accessory sends the ACK command when the accessory has completed its mode change.

GetDevCaps Command

Direction: media player→accessory

The media player sends this command to the microphone accessory to determine the features present on the accessory. In response, the accessory sends RetDevCaps command with the payload indicating the capabilities it supports.

RetDevCaps Command

Direction: accessory→media player

The accessory sends this command in response to the command GetDevCaps command sent by the media player. The microphone accessory returns the payload indicating which capabilities it supports.

GetDevCtrl Command

Direction: media player→accessory

The media player sends this command to get the accessory control state for the specified control type. In response, the accessory sends RetDevCtrl command with its current control state. If this command is not supported by the accessory—that is, if the microphone does not have any configurable controls—it should return an ACK command with a bad parameter error status.

RetDevCtrl Command

Direction: accessory→media player

The accessory sends this command in response to the command GetDevCtrl command received from the media player. The accessory returns the current control state for the specified control type. Control types are only supported if the associated capabilities bits are set in the command RetDevCaps command.

14

SetDevCtrl Command

Direction: media player→accessory

The media player sends this command to set the accessory control state for the specified control type. In response, the accessory sends the ACK command with the command status. If this command is not supported by the accessory—that is, if the microphone does not have any configurable controls—it should return an ACK command with a bad parameter error status.

Simple Remote Lingo

A simple remote accessory sends a buttons status command to indicate an updated status of which buttons are held down. The data of the packet is a number of bytes indicating which buttons are currently held down. In one embodiment, the bytes are made up by ORing the masks of the buttons together. The accessory will send a command to indicate all buttons are released. While any buttons are held down the accessory should repeat this packet on a predetermined interval. If no packet of this sort is received by the host for a predetermined time period, the host may assume a packet was lost and go to “all buttons up” mode.

When the user presses and holds down a button, a simple remote accessory should generate the button status packet immediately and repeat it for as long as the button is pressed. If a second button is pressed while the first button is down, the button status packet sent by the accessory should include status for both buttons, and this packet should be repeated for as long as both buttons are held down. Table 3 lists the possible media player button states.

Simple Remote Button Map

TABLE 3

Button	Number
6Play/Pause	0
Volume Up	1
Volume Down	2
Next Track	3
Previous Track	4
Next Album	5
Previous Album	6
Stop	7
Play/Resume	8
Pause	9
Mute toggle	10
Next Chapter	11
Previous Chapter	12
Next Playlist	13
Previous Playlist	14
Shuffle setting advance	15
Repeat setting advance	16
Power On	17
Power Off	18
Backlight for 30 seconds	19
Begin FF	20
Begin REW	21
Menu	22
Select	23
Up Arrow	24
Down Arrow	25

Some media player button states are interpreted differently by the media player when pressed and held down. These are as follows:

The Next Track button is treated as a Scan Forward button when pressed and held while a track is playing.

The Previous Track button is treated as a Scan Backward button when pressed and held while a track is playing.

The Play/Pause button is treated as a Power Off button when pressed and held.

The Menu button is treated as a Display Backlight On/Off button when pressed and held.

If the media player is in Browse mode, the Select button is treated as an Add Track to On-The-Go Playlist button when pressed and held.

Repeated Next Track and Previous Track commands (see Table 3) without an intervening button status packet indicating all buttons are up, are interpreted as Fast Forward and Rewind commands. For a locking Fast Forward or Rewind button, use the Begin Fast Forward or Begin Rewind commands to start the operation and a Play/Resume command to return to the play state.

The Next and Previous Album commands (see Table 3) have no effect if there is no next or previous album to go in to the Now Playing list.

Below is the command for the simple remote lingo.

ContextButtonStatus Command

Direction: Accessory→media player

The accessory sends this command to the media player when a button event occurs. When all buttons are released, the accessory should send a button status packet with a 0x0 payload to indicate that no buttons are pressed. The media player does not return a packet to the accessory in response to this command.

Display Remote Lingo

The display remote lingo is for accessories that need to control the state of the media player, recreate a portion of the media player UI on a remote display, or control the state of the media player equalizer (EQ). The display remote protocol can be used by simple inline-display remotes (remotes that have single-line display and play control buttons) and more complex accessories that have full multi-line graphical displays to show information about the track, artist, or album; current play or pause state; track position; battery; shuffle and time. By supporting multiple lingo, an accessory can use the display remote lingo in combination with other lingo to create a fully functional product. Accessories can also use this lingo to control the state of the media player equalizer. The display remote lingo supports serial accessories attached to the Audio/Remote or 30-pin connector.

The display remote command set uses a single byte command format similar to the general and simple remote lingo. Accessories using the display remote lingo can identify using the general lingo, with either the identify single lingo or IdentifyAccessoryLingo multiple lingo commands.

Table 4 illustrates the commands for the display remote lingo.

TABLE 4

Command
ACK
GetCurrentEQProfileIndex
RetCurrentEQProfileIndex
SetCurrentEQProfileIndex
GetNumEQProfiles
RetNumEQProfiles
GetIndexedEQProfileName
RetIndexedEQProfileName
SetRemoteEventNotification
RemoteEventNotification
GetRemoteEventStatus
RetRemoteEventStatus
GetMediaPlayerStateInfo
RetMediaPlayerStateInfo
SetMediaPlayerStateInfo
GetPlayStatus
RetPlayStatus

TABLE 4-continued

Command
SetCurrentPlayingTrack
GetIndexedPlayingTrackInfo
RetIndexedPlayingTrackInfo
GetNumPlayingTracks
RetNumPlayingTracks
GetPowerBatteryState
RetPowerBatteryState
GetSoundCheckState
RetSoundCheckState
SetSoundCheckState

The commands, their direction and their functions are described in detail hereinbelow.

ACK Command

Direction: media player→accessory

The media player sends this command to acknowledge the receipt of a command from the accessory and return the command status. The command ID field indicates the accessory command for which the response is being sent. The command status indicates the result of the command (success or failure).

GetCurrentEQProfileIndex Command

Direction: accessory→media player

Requests the current equalizer (EQ) profile setting index. In response, the media player sends the RetCurrentEQProfileIndex command packet.

RetCurrentEQProfileIndex Command

Direction: media device→accessory

The media accessory sends this command, returning the current equalizer profile setting index, in response to the GetCurrentEQProfileIndex command packet sent by the accessory.

SetCurrentEQProfileIndex Command

Direction: accessory→media player

Set the current equalizer profile setting index and optionally restores the original equalizer setting on accessory detach. The valid equalizer index range can be determined by sending a GetNumEQProfiles command. In response to this command, the media player returns an ACK packet with the status of this command.

GetNumEQProfiles Command

Direction: accessory→media player

Requests the number of media player equalizer profile settings. In response, the media player sends the RetNumEQProfiles command packet.

RetNumEQProfiles Command

Direction: media player→accessory

Returns the number of equalizer profiles. The media player sends this command in response to the GetNumEQProfiles command packet sent by the accessory.

GetIndexedEQProfileName Command

Direction: accessory→media player

Requests the media player equalizer profile setting name for a given equalizer profile index. In response, the media player sends the RetIndexedEQProfileName command packet. The valid profile index range can be obtained by sending the GetNumEQProfiles command.

RetIndexedEQProfileName Command

Direction: media player→accessory

Returns the media player equalizer profile setting name for the specified equalizer profile index in response to the GetIndexedEQProfileName command.

US 7,529,872 B1

17

SetRemoteEventNotification Command

Direction: accessory→media player

Enables asynchronous remote event notification for media player events. On accessory detach, event notification is reset to the default disabled state.

RemoteEventNotification Command

Direction: media player→accessory

The media player sends this command asynchronously whenever an enabled event change has occurred. Use the SetRemoteEventNotification command to control which events are enabled.

GetRemoteEventStatus Command

Direction: accessory→media player

Gets the status of state information that has changed on the media player. In response, the media player sends a RetRemoteEventStatus command, containing a bitmask of event states that changed since the last GetRemoteEventStatus command and clears all the remote event status bits. This command may be used to poll the media player for event changes without enabling asynchronous remote event notification.

RetRemoteEventStatus Command

Description: media player→accessory

The media player sends this command in response to the GetRemoteEventStatus command.

GetMediaPlayerStateInfo Command

Direction: accessory→media player

Obtains media player state information. In response, the media player sends a RetMediaPlayerStateInfo command with the requested state information.

RetMediaPlayerStateInfo Command

Direction: media player→accessory

The media player sends this command in response to the GetMediaPlayerStateInfo command.

SetMediaPlayerStateInfo Command

Direction: accessory→media player

Set the media player state. The information type specifies the type of information to update. In response, the media player sends an ACK command with the results of the operation.

GetPlayStatus Command

Direction: accessory→media player

Gets the current media player play status information. In response, the media player sends a RetPlayStatus command with the current play state, track index, track position, and track length.

RetPlayStatus Command

Direction: media player→accessory

The media player sends this command in response to the GetPlayStatus command and returns the current media player play status information.

SetCurrentPlayingTrack Command

Direction: accessory→media player

Sets the currently playing track to the track at the specified index. The total number of playing tracks can be obtained by sending a GetNumPlayingTracks command.

GetIndexedPlayingTrackInfo Command

Direction: accessory→media player

Gets track information for the specified playing track index. The infoType field specifies the type of information to be returned, such as track title, artist title, artist name, album name, track genre, and track chapter information. In response, the media player sends a RetIndexedPlayingTrackInfo command with the requested track information.

18

RetIndexedPlayingTrackInfo Command

Direction: media player→accessory

The media player sends this command in response to the GetIndexedPlayingTrackInfo command. It returns the requested type of information and data for the specified playing track.

RetNumPlayingTracks Command

Direction: media player→accessory

The media player sends this command in response to the GetNumPlayingTracks command received from the accessory. It returns the total number of tracks queued in the playback engine.

GetPowerBatteryState Command

Direction: accessory→media player

Gets the power and battery level state of the media player. In response, the media player sends a RetPowerBatteryState command with the power and battery information.

RetPowerBatteryState Command

Direction: media player→accessory

The media player sends this command in response to the GetPowerBatteryState command and returns the current media player power state and battery level.

GetSoundCheckState Command

Direction: accessory→media player

Gets the current sound check setting. When enabled, sound check adjusts track playback volume to the same level. In response, the media player sends a RetSoundCheckState command with the current sound check state.

RetSoundCheckState Command

Direction: media player→accessory

The media player sends this command in response to the GetSoundCheckState command and returns the current state of the sound check setting.

SetSoundCheckState Command

Direction: accessory→media player

Sets the state of the sound check setting and optionally saves the previous sound check state to be restored on accessory detach. In response to this command, the media player sends an ACK packet with the status of the command.

Extended Interface Lingo Specification

Operating Modes

The media player can be considered to operate in two major modes, standard UI mode and extended Interface mode.

Standard UI Mode

This is the standard user interface mode and allows the media player to be driven by its front panel display and buttons.

Extended Interface Mode

The media player transitions into the Extended Interface mode when either of the following occurs:

(1)—A single-lingo accessory is connected to the media player and issues the Identify command, specifying the Extended Interface lingo as its supported lingo.

(2)—A multi-lingo accessory is connected to the media player and issues a General Lingo Command: EnterRemoteUIMode command.

If the media player is playing a track during this transition, the playback is automatically paused.

The extended interface protocol allows accessories to replace the graphic with a downloaded image set through a SetDisplayImage command. Removing power from the media player while a connection remains results in the media player going into a light sleep state after a predetermined amount of inactivity. The controls of the media player are disabled when in extended interface mode.

US 7,529,872 B1

19

The media player transitions back to standard UI mode when any of the following occurs:

- (1)—The accessory is disconnected from the media player.
- (2)—A multi-lingo accessory issues the General Lingo Command: ExitRemoteUIMode.

The accessory re-identifies itself, using either of the general lingo commands Identify or IdentifyAccessoryLingo.

If the media player is playing a track during this transition, the playback is automatically paused. Any media player settings with the restore on exit feature state are restored when the media player is disconnected.

Light Sleep State

The media player screen, playback, and most major parts of the media player are off while the media player is in the light sleep state. The media player transitions from extended interface mode to the light sleep state when power is detached and playback is idle. A period of inactivity is required before the media player transitions into the light sleep state. When power is restored, the media player returns to the extended interface mode.

A media player will not sleep while it remains attached to an active USB host. The USB host must switch off its host controller to force a media player in extended interface mode into the light sleep state. If the media player is not currently in extended interface mode, playback must be paused before the host controller is turned off. In extended interface mode, there is no need to pause media playback before turning off the host controller because this action generates a disconnect event that causes the media player to exit the extended interface mode and allows the media player to transition into the light sleep state. Attaching USB power to a media player in light sleep state will wake it up.

Switching Between Standard UI and Extended Interface Modes

There are four general lingo commands that allow accessories to determine what mode the media player is in and to switch between the two major modes, Standard UI and Extended Interface. These commands were implemented to allow a device to switch between modes without having to unplug the accessory. Multi-lingo accessories must use these commands to switch into and out of the extended interface mode.

Table 5 lists the general lingo command codes for querying, entering, and exiting the extended interface protocol.

TABLE 5

General Lingo command	Requires authentication UART serial port link	Requires authentication USB port link
RequestRemoteUIMode	No	Yes
ReturnRemoteUIMode	No	Yes
EnterRemoteUIMode	No	Yes
ExitRemoteUIMode	No	Yes

RequestRemoteUIMode Command

Direction: accessory→media player

Requests the Extended Interface mode from the media player. The media player responds with a ReturnRemoteUIMode command.

ReturnRemoteUIMode Command

Direction: media player→accessory

Returns the current operating mode of the media player UI

EnterRemoteUIMode Command

20

Direction: accessory→media player

The accessory sends this command to the media player to force it to enter the Extended Interface mode. If the media player is already in the extended interface mode, it immediately returns a general lingo ACK command packet, notifying the user that the command was successful.

ExitRemoteUIMode Command

Direction: accessory→media player

The accessory sends this command to the media player to force it to exit the Extended Interface mode. If the media player is already in the standard UI mode, it immediately returns a general lingo ACK command packet, notifying the user that the command was successful.

Using the Extended Interface Protocol

In order to effectively use the extended interface protocol, there are two logical entities that need to be managed while browsing and playing content: the content database engine and the playback engine. The following describes those engines and gives an example of command traffic between an extended interface accessory and a media player.

The Playback Engine

The playback engine is active when the media player is in a playback state, such as play, fast forward, and rewind. It has a special play list, called the Now Playing playlist, that is used to determine what track or content item will be played next. The PlayCurrentSelection command is used to transfer the currently selected database items to the Now Playing Playlist and start the player at a specified item within that list. Changes to the database selection before or after this PlayCurrentSelection command have no effect on the current playback.

The Database Engine

The database engine is always accessible when the unit is awake. It can be manipulated remotely and allows groups of content items to be selected, independently of the playback engine. This allows the user to listen to an existing track or playlist while checking the media player database for another selection. Once a different database selection is made, the user selection (the track or content playlist) is sent to the playback engine. The commands such as ResetDBSelection and GetNumberCategorizedDBRecords are examples of commands that are used to manipulate the Database Engine.

Database Category Hierarchies

The database engine uses categories to classify music and other records stored in the database. Possible categories are playlist, genre, artist, album, track, composer and audiobook. A list of records can be assembled, based on the various selected categories, to create a user list of records (a playlist).

The database categories have a hierarchy by which records are sorted and retrieved. This category hierarchy has an impact on the order in which records should be selected. For example, if a low category, such as album, is selected first, followed by a higher relative category such as genre, the album selection is invalidated and is ignored. When creating a new set of database selections, the accessory should begin by resetting all database selections, using the ResetDBSelection command, and selecting the desired database categories from highest to lowest relative category. A representation of the database hierarchy is shown in Table 6.

US 7,529,872 B1

21

22

TABLE 6

Category	Notes
All (highest level)	This is the state after a ResetDBSelection command. No database categories are selected. If the GetNumberCategorizedDBRecords command is sent while in this state, it returns the total number of records for the requested category.
Playlist	When the SelectDBRecord command selects a playlist, all lower database category selections (genre, artist or composer, album, and track) are invalidated.
Genre	When the SelectDBRecord command selects a genre, all lower database category selections (artist or composer, album, and track) are invalidated.
Artist or Composer	When the SelectDBRecord command selects an artist or composer, all album and track category selections are invalidated.
Album	When the SelectDBRecord command selects an album, all track category selections are invalidated.
Song/Audiobook Track (lowest level)	When the Select DBRecord command selects a track, either a song or an audiobook, it is automatically transferred from the Database Engine to the Playback Engine.

Extended Interface Mode Commands

Table 7 lists the command for the Extended Interface protocol.

TABLE 7

Command	Target engine
Acknowledge	N/A
GetCurrentPlaying-TrackChapterInfo	Playback Engine
GetCurrentPlaying-TrackChapterInfo	Playback Engine
GetCurrentPlaying-TrackChapterName	Playback Engine
GetAudiobookSpeed	N/A
GetIndexedPlaying-TrackInfo	Playback Engine
GetNumberCategorized-DBRecords	Database Engine
GetPlayStatus	Playback Engine
GetCurrentPlaying-TrackIndex	Playback Engine
GetIndexedPlaying-TrackTitle	N/A
GetIndexedPlaying-TrackArtistName	Playback Engine
GetIndexedPlaying-TrackAlbumName	Playback Engine
GetShuffle	N/A
GetRepeat	N/A
GetMonoDisplay-ImageLimits	N/A
GetNumPlayingTracks	Playback Engine
GetColorDisplay-ImageLimits	N/A
ReturnCurrentPlaying-TrackChapterInfo	Playback Engine
ReturnCurrentPlaying-TrackChapterPlayStatus	Playback Engine
ReturnCurrentPlaying-TrackChapterName	Playback Engine
ReturnAudiobookSpeed	N/A
ReturnIndexedPlaying-TrackInfo	N/A
ReturnProtocolVersion	N/A
ReturnMediaPlayerName	N/A

TABLE 7-continued

Command	Target engine
ReturnNumber-CategorizedDBRecords	N/A
ReturnCategorized-DatabaseRecord	N/A
ReturnPlayStatus	N/A
ReturnCurrentPlaying-TrackIndex	N/A
ReturnIndexedPlaying-TrackTitle	N/A
ReturnIndexedPlaying-TrackArtistName	N/A
ReturnIndexedPlaying-TrackAlbumName	N/A
ReturnShuffle	N/A
ReturnRepeat	N/A
ReturnMonoDisplay-ImageLimits	N/A
ReturnNumPlayingTracks	N/A
ReturnColorDisplay-ImageLimits	N/A
SetCurrentPlaying-TrackChapter	Playback Engine
SetAudiobookSpeed	N/A
SetPlayStatusChange-Notification	Playback Engine
SetShuffle	N/A
SetRepeat	N/A
SetDisplayImage	N/A
SetCurrentPlayingTrack	Playback Engine
RequestProtocolVersion	N/A
RequestMediaPlayerName	N/A
ResetDBSelection	Database Engine
RetrieveCategorized-DatabaseRecords	Database Engine
PlayStatusChange-Notification	N/A
PlayCurrentSelection	Database and Playback Engines. This command copies items from the database to the Playback Engine.
PlayControl	Playback Engine
SelectSortDBRecord	Database Engine

Each of these commands, their direction and function are described hereinbelow.

ACK Command

Direction: media player→accessory

The media player sends this telegram to acknowledge the receipt of a command and return the command status. The command ID field indicates the accessory command for which the response is being sent. The command status indicates the results of the command (success or failure).

GetCurrentPlayingTrackChapterInfo Command

Direction: accessory→media player

Applies to: playback engine

Requests the chapter information of the currently playing track. In response, the media player sends a ReturnCurrentPlayingTrackChapterInfo telegram to the accessory.

ReturnCurrentPlayingTrackChapterInfo Command

Direction: media player→accessory

Returns the chapter information of the currently playing track. The media player sends this telegram in response to the GetCurrentPlayingTrackChapterInfo telegram from the accessory. The chapter information includes the currently playing track's chapter index, as well as the total number of chapters.

SetCurrentPlayingTrackChapter Command

Direction: accessory→media player

Applies to: playback engine

US 7,529,872 B1

23

Sets the currently playing track chapter. In response to the SetCurrentPlayingTrackChapter command, the media player sends an ACK telegram with the command status.

GetCurrentPlayingTrackChapterPlayStatus

Direction: accessory→media player

Applies to: playback engine

Requests the chapter playtime status of the currently playing track. In one embodiment the status includes the chapter length and the time elapsed within that chapter. In response to a valid telegram, the media player sends a ReturnCurrentPlayingTrackChapterPlayStatus telegram to the accessory.

ReturnCurrentPlayingTrackChapterPlayStatus Command

Direction: media player→accessory

Returns the play status of the currently playing track chapter. The media player sends this telegram in response to the GetCurrentPlayingTrackChapterPlayStatus telegram from the accessory.

GetCurrentPlayingTrackChapterName Command

Direction: accessory→media player

Applies to: playback engine

Requests a chapter name in the currently playing track. In response to a valid telegram, the media player sends a ReturnCurrentPlayingTrackChapterName telegram to the accessory.

ReturnCurrentPlayingTrackChapterName Command

Direction: media player→accessory

Returns a chapter name in the currently playing track. In response to a valid telegram, the media player sends a ReturnCurrentPlayingTrackChapterName telegram to the accessory.

ReturnCurrentPlayingTrackChapterName Command

Direction: media player→accessory

Returns a chapter name in the currently playing track. The media player sends this telegram in response to a valid GetCurrentPlayingTrackChapterName telegram from the accessory.

GetAudiobookSpeed Command

Direction: accessory→media player

Requests the current media player audiobook speed state. The media player responds with the ReturnAudiobookSpeed telegram indicating the current audiobook speed.

ReturnAudiobookSpeed Command

Direction: mediaplayer→accessory

Returns the current audiobook speed setting. The media player sends this telegram in response to the GetAudiobookSpeed command from the accessory.

Table 8 shows the possible audiobook speed states returned by this command.

TABLE 8

Value	Meaning
0xFF	Slow (-1)
0x00	Normal
0x01	Fast (+1)
0x02-0xFE	Reserved

SetAudiobookSpeed Command

Direction: accessory→media player

Sets the speed of audiobook playback. The media player audiobook speed states are listed in Table 8. This telegram has two modes: one to set the speed of the currently playing audiobook and a second to set the audiobook speed for all audiobooks.

24

GetIndexedPlayingTrackInfo Command

Direction: accessory→media player

Applies to: playback engine

Gets track information for the track at the specified index.

5 The track info type field specifies the type of information to be returned, such as song lyrics, podcast name, episode date, and episode description. In response, the media player sends the ReturnIndexedPlayingTrackInfo command.

ReturnIndexedPlayingTrackInfo Command

10 Direction: media player→accessory

Returns the requested track information type and data. The media player sends this command in response to the GetIndexedPlayingTrackInfo command.

RequestProtocolVersion Command

15 Direction: accessory→media player

Requests the version of the running Extended Interface protocol from the media player. The media player responds with a ReturnProtocolVersion command.

ReturnProtocolVersion Command

20 Direction: media player→accessory

Returns the media player Extended Interface protocol version number. The media player sends this command in response to the RequestProtocolVersion command from the accessory.

25 RequestMediaPlayerName Command

Direction: accessory→media player

Returns the name of the user's media player or "media player" if the media player name is undefined. This allows the media player name to be shown in the human-machine interface.

30 ReturnMediaPlayerName Command

Direction: media player→accessory

The media player sends this command in response to the RequestMediaPlayerName telegram from the accessory.

35 ResetDBSelection Command

Direction: accessory→media player

Applies to: database engine

Resets the current database selection to an empty state and invalidates the category entry count. That is, it sets the count to 0 for all categories except the playlist category.

40 SelectDBRecord Command

Direction: accessory→media player

Applies to: database engine. Selecting a single track automatically passes it to the playback engine.

45 Selects one or more records in the database engine, based on a category relative index.

GetNumberCategorizedDBRecords Command

Direction: accessory→media player

Applies to: database engine

50 Retrieves the number of records in a particular database category.

ReturnNumberCategorizedDBRecords Command

Direction: media player→accessory

Returns the number of database records matching the specified database category.

RetrieveCategorizedDatabaseRecords Command

Direction: accessory→media player

Applies to: database engine

60 Retrieves one or more database records from the media player, typically based on the results from the GetNumberCategorizedDBRecords command query.

ReturnCategorizedDatabaseRecord Command

Direction: media player→accessory

Contains information for a single database record.

65 GetPlayStatus Command

Direction: accessory→media player

Applies to: playback engine

US 7,529,872 B1

25

Requests the current media player playback status, allowing the accessory to display feedback to the user.

ReturnPlayStatus Command

Direction: media player→accessory

Returns the current media player playback status.

GetCurrentPlayingTrackIndex Command

Direction: accessory→media player

Applies to: playback engine

Requests the playback engine index of the currently playing track.

ReturnCurrentPlayingTrackIndex Command

Direction: media player→accessory

Returns the playback engine index of the current playing track in response to the GetCurrentPlayingTrackIndex telegram from the accessory.

GetIndexedPlayingTrackTitle Command

Direction: accessory→media player

Applies to: playback engine

Requests the title name of the indexed playing track from the media player. In response to a valid telegram, the media player sends a ReturnIndexedPlayingTrackTitle telegram to the accessory.

ReturnIndexedPlayingTrackTitle Command

Direction: media player→accessory

Returns the title of the indexed playing track in response to a valid GetIndexedPlayingTrackTitle telegram from the accessory.

GetIndexedPlayingTrackArtistName Command

Direction: accessory→media player

Applies to: playback engine

Requests the name of the artist of the indexed playing track. In response to a valid telegram, the media player sends a ReturnIndexedPlayingTrackArtistName telegram to the accessory.

ReturnIndexedPlayingTrackArtistName Command

Direction: media player→accessory

Returns the artist name of the indexed playing track in response to a valid GetIndexedPlayingTrackArtistName telegram from the accessory.

GetIndexedPlayingTrackAlbumName Command

Direction: accessory→media player

Applies to: playback engine

Requests the album name of the indexed playing track.

ReturnIndexedPlayingTrackAlbumName Command

Direction: media player→accessory

Returns the album name of the indexed playing track in response to a valid GetIndexedPlayingTrackAlbumName telegram from the accessory.

SetPlayStatusChangeNotification Command

Direction: accessory→media player

Applies to: playback engine

Sets the state of play status change notifications from the media player to the accessory.

PlayStatusChangeNotification Command

Direction: media player→accessory

The media player sends this telegram to the accessory when the media player play status changes, if the accessory has previously enabled notifications using the SetPlayStatusChangeNotification command. This telegram contains details about the new play status.

PlayCurrent Selection Command

Direction: accessory→media player

Applies to: playback engine and database engines. This command copies items from the database engine to the playback engine.

26

Requests playback of the currently selected track or list of tracks.

PlayControl Command

Direction: accessory→media player

Applies to: playback engine

Sets the new play state of the media player.

GetShuffle Command

Direction: accessory→media player

Requests the current state of the media player shuffle setting.

ReturnShuffle Command

Direction: media player→accessory

Returns the current state of the shuffle setting.

SetShuffle Command

Direction: accessory→media player

Sets the media player shuffle mode.

GetRepeat Command

Direction: accessory→media player

Requests the track repeat state of the media player.

ReturnRepeat Command

Direction: media player→accessory

Returns the current media track repeat state to the accessory.

SetRepeat Command

Direction: accessory→media player

Sets the repeat state of the media player.

SetDisplayImage Command

Direction: accessory→media player

Sets a bitmap image that is shown on the media player display when it is connected to the accessory.

GetMonoDisplayImageLimits Command

Direction: accessory→media player

Requests the limiting characteristics of the monochrome image that can be sent to the media player for display while it is connected to the accessory.

ReturnMonoDisplayImageLimits Command

Direction: media player→accessory

Returns the limiting characteristics of the monochrome image that can be sent to the media player for display while it is connected to the accessory.

GetNumPlayingTracks Command

Direction: accessory→media player

Applies to: playback engine

Requests the number of tracks in the list of tracks queued to play on the media player.

ReturnNumPlayingTracks Command

Direction: media player→accessory

Returns the number of tracks in the actual list of tracks queued to play, including the currently playing track (if any).

SetCurrentPlayingTrack Command

Direction: accessory→media player

Applies to: playback engine

Sets the index of the track to play in the Now Playing playlist on the media player.

SelectSortDBRecord Command

Direction: accessory→media player

Applies to: database engine

Selects one or more records in the media player database, based on a category-relative index.

GetColorDisplayImageLimits Command

Direction: accessory→media player

Requests the limiting characteristics of the color image that can be sent to the media player for display while it is connected to the accessory.

ReturnColorDisplayImageLimits Command

Direction: media player→accessory

Returns the limiting characteristics of the color image that can be sent to the media player for display while it is connected to the accessory.

RF Transmitter Lingo

The RF Transmitter Lingo is used for accessories that transmit the media player analog audio over radio frequencies (typically an unused frequency in the FM band). The Begin Transmission command packet notifies the external RF transmitter accessory that the media player is entering playback mode. The End Transmission command packet notifies the RF transmitter that the media player is exiting playback mode (that is, it is stopped, entering light sleep mode, and so forth).

Begin Transmission Command

Direction: media player→accessory

The media player sends this command to notify the accessory that high power may be used and that it should begin transmitting.

End Transmission Command

Direction: media player→accessory

The media player sends this command to notify the accessory to stop transmitting and to stop using accessory high power.

Single Lingo Accessory Identification

Accessories should use the general lingo identify command to identify themselves as supporting a single accessory lingo. FIG. 7 illustrates a single-lingo accessory command flow. In this command flow, first the accessory waits for media player internal bootstrap/wakeup and allows the media player to synchronize to the accessory rate. Next, the media player is informed that accessory is present and supports the Extended Interface lingo. Thereafter, the accessory gets (requests) the version of the extended interface lingo protocol supported by the media player and the media player sends the protocol version for verification. If received, the accessory has successfully forced the media player into extended interface mode. The accessory should use the returned lingo version to determine which features are available from the media player.

Multiple Lingo Accessory Identification

Accessories should use the general lingo IdentifyAccessoryLingoes command to identify themselves as supporting multiple accessory lingoes (not including the general lingo), or when they are communicating with the media player over the USB port link.

Accessories that support the extended interface mode and identify as a multi-lingo accessory do not automatically switch into the extended interface mode after the identification process completes. These accessories must explicitly switch into and out of the extended interface mode, using the general lingo commands.

Accessories that identify themselves as supporting multiple lingoes should use the steps shown in FIG. 8. FIG. 8 illustrates a multiple lingo command flow. In this flow, first the accessory waits for the media player bootstrap/wakeup and allows the media player to synchronize to the accessory rate. Next, the media player is informed that the accessory is present, and the bit field of supported lingoes is sent. The media player acknowledges the IdentifyAccessoriesLingoes command. Next, the media player general lingo version information is requested, and the media player sends the general lingo protocol version. This information is used to determine which features are available from the media player. The accessory is then switched into the extended interface mode, sending the EnterExtendedInterfaceMode Command, and the media player sends a first ACK indicating how long it will take the media player to switch modes. The media player

sends a second ACK notifying the accessory of the mode switch, and the media player sends an extended interface protocol version for verification. If received, the accessory has successfully forced the media player into extended interface mode. The accessory should use the returned information to determine which features are available from the media player. This figure assumes the accessory is using the UART serial port link.

A connector interface system for a communication accessory is disclosed. The interface includes a docking connector. The docking connector includes first make/last break contacts that minimize internal damage to the internal electronics. The docking connector also includes a specific keying arrangement to prevent noncompliant connectors from being plugged in, and thereby minimizes potential damage to the multi-communication accessory. The remote connector provides for the ability to output audio, input audio, and output video. The connector system interface also includes a media player protocol to control accessory features. These controls allow for a user to control the functionality of media player in a variety of modes.

It will be clear that the invention described herein provides significant functional enhancements when incorporated into prior art systems. FIG. 9 shows an exemplary system 900 using the connector described herein. The system includes at least two accessories 902 and 904 to be connected together. Each accessory 902 and 304 contains an enhanced connector system interface 902a and 902b as described herein. They can be connected together either wirelessly or via a cable. Alternatively, one of the accessories can be itself constructed integrally with the connector so that it can be docked in place when mated with the other accessory, for example, as is done with flash-backed memory sticks, or similar accessories that include an integral connector. When connected together, the accessories can exchange data via the connector interface system signals that would not be conveniently possible but for the existence of the connector pins and the protocol.

The connection interface system can be utilized with a plurality of accessory devices interfaces including, but not limited to, car stereo interface, home stereo interfaces, remote controls, radio interfaces, microphones, voice recorders, wireless accessories which allow for communications to other accessories and speakers.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will readily recognize that there could be variations to the embodiments and those variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.

What is claimed is:

1. A method of operating an accessory device, the method comprising:
 - establishing a communication link to a media player, wherein the accessory is configured to communicate with the media player via the communication link using a media player accessory protocol comprising a general lingo and a plurality of accessory lingoes supported by the media player; and
 - sending an identification command of the general lingo to the media player via the communication link, wherein the identification command identifies at least one of the accessory lingoes supported by the media player as being supported by the accessory.
2. The method of claim 1 wherein the identification command indicates that the accessory supports exactly one of the

US 7,529,872 B1

29

accessory lingo supported by the media player and identifies the one accessory lingo that the accessory supports.

3. The method of claim 1 wherein the identification command indicates that the accessory supports one or more of the accessory lingo supported by the media player and identifies each of the accessory lingo supported by the accessory.

4. The method of claim 1 wherein the plurality of accessory lingo supported by the media player includes at least one lingo from a group consisting of a microphone lingo, a simple remote lingo, a display remote lingo, an extended interface lingo, and an RF transmitter lingo.

5. The method of claim 1 wherein the identification command identifies a microphone lingo as an accessory lingo supported by the accessory, the method further comprising:

receiving a first command of the microphone lingo from the media player, the first command signaling that recording of audio input is beginning; and

receiving a second command of the microphone lingo from the media player, the second command signaling that recording of audio input is ending.

6. The method of claim 1 wherein the identification command identifies a simple remote lingo as an accessory lingo supported by the accessory, the method further comprising:

sending a command of the simple remote lingo to the media player, the command indicating a status of a user-operable control of the accessory.

7. The method of claim 1 wherein the identification command identifies a display remote lingo as an accessory lingo supported by the accessory, the method further comprising:

sending a first command of the display remote lingo to the media player, the first command requesting information about one or more tracks queued for playback by the media player; and

receiving a second command of the display remote lingo from the media player, the second command providing the requested information about the one or more tracks queued for playback by the media player.

8. The method of claim 1 wherein the identification command identifies a display remote lingo as an accessory lingo supported by the accessory, the method further comprising:

sending a first command of the display remote lingo to the media player, the first command requesting information about a state of the media player;

receiving a second command of the display remote lingo from the media player, the second command providing the requested information about the state of the media player; and

sending a third command of the display remote lingo to the media player, the third command instructing the media player to change the state of the media player.

9. The method of claim 1 wherein the identification command identifies a display remote lingo as an accessory lingo supported by the accessory, the method further comprising:

sending a first command of the display remote lingo to the media player, the first command requesting information indicating a current equalizer setting of the media player;

receiving a second command of the display remote lingo from the media player, the second command providing the information indicating the current equalizer setting of the media player; and

sending a third command of the display remote lingo to the media player, the third command instructing the media player to change the current equalizer setting of the media player.

30

10. The method of claim 1 wherein the identification command identifies an RF transmitter lingo as an accessory lingo supported by the accessory, the method further comprising:

receiving a first command of the RF transmitter lingo from the media player, the first command indicating that the accessory should begin a transmission; and

receiving a second command of the RF transmitter lingo from the media player, the second command indicating that the accessory should end the transmission.

11. The method of claim 1 wherein the at least one accessory lingo includes an extended interface lingo, the method further comprising:

sending a first command of the extended interface lingo to the media player, the first command requesting information about a currently playing track; and

receiving a second command of the extended interface lingo from the media player, the second command providing the requested information about the currently playing track.

12. The method of claim 1 wherein the at least one accessory lingo includes an extended interface lingo, the method further comprising:

sending a first command of the extended interface lingo to the media player, the first command requesting information about a track queued for playback by the media player; and

receiving a second command of the extended interface lingo from the media player, the second command providing the requested information about the track queued for playback by the media player.

13. The method of claim 1 wherein the at least one accessory lingo includes an extended interface lingo, the method further comprising:

sending a first command of the extended interface lingo to the media player, the first command requesting information from a database of tracks stored on the media player; and

receiving a second command of the extended interface lingo from the media player, the second command providing the requested information from the database of tracks stored on the media player.

14. The method of claim 1 wherein the at least one accessory lingo includes an extended interface lingo, the method further comprising:

sending a first command of the extended interface lingo to the media player, the first command instructing the media player to select one or more tracks from a database of tracks stored on the media player; and

sending a second command of the extended interface to the media player, the second command instructing the media player to queue the selected one or more tracks for playback.

15. The method of claim 5 further comprising: in response to receiving the first command of the microphone lingo, transitioning to a high power state; and in response to receiving the second command of the microphone lingo, transitioning to a low power state.

16. The method of claim 5 further comprising: receiving a third command of the microphone lingo from the media player, the third command signaling a beginning of playback of recorded audio input; and receiving a fourth command of the microphone lingo from the media player, the fourth command signaling an ending of playback of recorded audio input.

17. The method of claim 16 further comprising: in response to receiving the third command of the microphone lingo, transitioning to a high power state; and

US 7,529,872 B1

31

in response to receiving the fourth command of the microphone lingo, transitioning to a low power state.

18. The method of claim **10** further comprising:

in response to receiving the first command, transitioning to a high power state and beginning the transmission; and in response to receiving the second command, ending the transmission and transitioning to a low power state.

19. The method of claim **11** wherein the at least one accessory lingo includes an extended interface lingo, the method further comprising:

sending a mode controlling command of the general lingo to the media player, the mode controlling command indicating that the media player should enter an extended interface operating mode.

20. An accessory for use with a media player, the accessory comprising:

an interface capable of communicating with a media player in accordance with a media player accessory protocol, the media player accessory protocol comprising a general lingo and a plurality of accessory lingo supported by the media player, wherein the accessory supports at least one of the plurality of accessory lingo,

wherein the interface is configured to send an identification command of the general lingo to the media player, wherein the identification command identifies at least one of the accessory lingo as being supported by the accessory.

21. The accessory of claim **20** wherein the identification command indicates that the accessory supports exactly one of the plurality of accessory lingo supported by the media player and identifies the one accessory lingo that the accessory supports.

22. The accessory of claim **20** wherein the identification command indicates that the accessory supports one or more of the plurality of accessory lingo supported by the media player and identifies each of the accessory lingo that the accessory supports.

23. The accessory of claim **20** wherein the plurality of accessory lingo supported by the media player includes a microphone lingo, wherein the accessory supports the microphone lingo, and wherein the microphone lingo includes:

a first command receivable by the accessory and signaling a beginning of recording of audio input; and

a second command receivable by the accessory and signaling an ending of recording of audio input.

24. The accessory of claim **20** wherein the microphone lingo further includes:

a third command receivable by the accessory and signaling a beginning of playback of recorded audio input; and

a fourth command receivable by the accessory and signaling an ending of playback of recorded audio input.

25. The accessory of claim **20** wherein the plurality of accessory lingo supported by the media player includes a simple remote lingo, wherein the accessory supports the simple remote lingo, and wherein the simple remote lingo includes one or more commands sendable by the accessory to indicate a status of a user-operable control of the accessory.

26. The accessory of claim **20** wherein the plurality of accessory lingo supported by the media player includes a display remote lingo, wherein the accessory supports the display remote lingo, and wherein the display remote lingo includes a commands sendable by the accessory to control an equalizer setting of the media player.

27. The accessory of claim **20** wherein the plurality of accessory lingo supported by the media player includes a

32

display remote lingo, wherein the accessory supports the display remote lingo, and wherein the display remote lingo includes:

a first command sendable by the accessory to request information about one or more tracks queued for playback by the media player; and

a second command receivable by the accessory and providing information about one or more tracks queued for playback by the media player.

28. The accessory of claim **20** wherein the plurality of accessory lingo supported by the media player includes a display remote lingo, wherein the accessory supports the display remote lingo, and wherein the display remote lingo includes:

a first command sendable by the accessory to request information about a state of the media player;

a second command receivable by the accessory and providing information about the state of the media player; and

a third command sendable by the accessory to change the state of the media player.

29. The accessory of claim **20** wherein the plurality of accessory lingo supported by the media player includes an RF transmitter lingo, wherein the accessory supports the RF transmitter lingo, and wherein the RF transmitter lingo includes:

a first command receivable by the accessory and indicating when the accessory should begin a transmission; and

a second command receivable by the accessory and indicating when the accessory should end a transmission.

30. The accessory of claim **20** wherein the plurality of accessory lingo supported by the media player includes an extended interface lingo, wherein the accessory supports the extended lingo, and wherein the extended interface lingo includes:

a first command sendable by the accessory to request information about a currently playing track;

a second command receivable by the accessory and providing the requested information about the currently playing track;

a third command sendable by the accessory to request information about a track queued for playback by the media player;

a fourth command receivable by the accessory and providing the requested information about the track queued for playback by the media player;

a fifth command sendable by the accessory to request information from a database of tracks stored on the media player; and

a sixth command receivable by the accessory and providing the requested information from the database of tracks stored on the media player.

31. The accessory of claim **30** wherein the extended interface lingo further includes:

a seventh command sendable by the accessory to instruct the media player to select a track for playback.

32. The accessory of claim **30** wherein the general lingo further includes a mode controlling command sendable by the accessory to indicate when the media player should enter an extended interface operating mode.

33. The accessory of claim **30** wherein the general lingo includes at least one command from a group consisting of:

a RequestIdentify command; a RequestMediaPlayerName command; a RequestMediaPlayerSoftwareVersion command; a RequestMediaPlayerSerialNum command; a RequestMediaPlayerModelNum command; a RequestLingoProtocolVersion command; an Identify

US 7,529,872 B1

33

command; an IdentifyAccessoryLingoes command; an ACK command; an AckDevAuthenticationInfo command; an AckDevAuthenticationStatus command; an AckMediaPlayerAuthenticationInfo command; an AckMediaPlayerAuthenticationStatus command; a ReturnMediaPlayerName command; a ReturnMediaPlayerSoftwareVersion command; a ReturnMediaPlayerSerialNumber command; a ReturnLingoProtocolVersion command; a RetDevAuthenticationInfo command; a RetDevAuthenticationSignature command; a RetMediaPlayerAuthenticationInfo command; a RetMediaPlayerAuthenticationSignature command; a GetDevAuthenticationInfo command; a GetDevAuthenticationSignature command; a GetMediaPlayerAuthenticationInfo command; a GetMediaPlayerAuthenticationSignature command; and a NotifyMediaPlayerStateChange command.

34. The accessory of claim 20 wherein the accessory supports a microphone accessory lingo that includes at least one command from a group consisting of:

a BeginRecord command; an EndRecord command; a BeginPlayback command; an EndPlayback command; an ACK command; a GetDevAck command; a MediaPlayerModeChange command; a GetDevCaps command; a RetDevCaps command; a GetDevCtrl command; a RetDevCtrl command; and a SetDevCtrl command.

35. The accessory of claim 20 wherein the accessory supports a simple remote accessory lingo that includes a ContextButtonStatus command.

36. The accessory of claim 20 wherein the accessory supports a display remote accessory lingo that includes at least one command from a group consisting of:

an ACK command; a GetCurrentEQProfileIndex command; a RetCurrentEQProfileIndex command; a SetCurrentEQProfileIndex command; a GetNumEQProfiles command; a RetNumEQProfiles command; a GetIndexedEQProfileName command; a RetIndexedEQProfileName command; a SetRemoteEventNotification command; a RemoteEventNotification command; a GetRemoteEventStatus command; a RetRemoteStatus command; a GetMediaPlayerStateInfo command; a RetMediaPlayerStateInfo command; a SetMediaPlayerStateInfo command; a GetPlayStatus command; a RetPlayStatus command; a SetCurrentPlayingTrack command; a GetIndexedPlayingTrackInfo command; a RetIndexedPlayingTrackInfo command; a GetNumPlayingTracks command; a RetNumPlayingTracks command; a GetPowerBatteryState command; a RetPowerBatteryState command; a GetSoundCheckState command; a RetSoundCheckState command; and a SetSoundCheckState command.

37. The accessory of claim 20 wherein the accessory supports an RF transmitter accessory lingo that includes at least one command from a group consisting of:

a BeginTransmission command; and an EndTransmission command.

38. The accessory of claim 20 wherein the accessory supports an extended interface accessory lingo that includes at least one command from a group consisting of:

an Acknowledge command; a GetCurrentPlayingTrackChapterInfo command; GetCurrentPlayingTrackChapterPlayStatus command; a GetCurrentPlayingTrackChapterName command; a GetAudiobookSpeed command; a GetIndexedPlayingTrackInfo command; a GetNumberCategorizedDB Records command; a GetPlayStatus command; a GetCurrentPlayingTrackIndex

34

command; a GetIndexedPlayingTrackTitle command; a GetIndexedPlayingTrackArtistName command; a GetIndexedPlayingTrackAlbumName command; a GetRepeat command; a GetShuffle command; a GetColorDisplayImageLimits command; a GetNumPlayingTracks command; a GetMonoDisplayImageLimits command; a ReturnCurrentPlayingTrackChapterInfo command; a ReturnCurrentPlayingTrackChapterPlayStatus command; a ReturnCurrentPlayingTrackChapterName command; a ReturnAudiobookSpeed command; a ReturnIndexedPlayingTrackInfo command; a ReturnProtocolVersion command; a ReturnMediaPlayerName command; a ReturnNumberCategorizedDBRecords command; a ReturnCategorizedDatabaseRecord command; a ReturnPlayStatus command; a ReturnCurrentPlayingTrackIndex command; a ReturnIndexedPlayingTrackArtistName command; a ReturnIndexedPlayingTrackAlbumName command; a ReturnShuffle command; a ReturnRepeat command; a ReturnMonoDisplayImageLimits command; a ReturnNumPlayingTracks command; a ReturnColorDisplayImageLimits command; a SetCurrentPlayingTrackChapter command; a SetAudiobookSpeed command; a SetPlayStatusChangeNotification command; a SetShuffle command; a SetDisplayImage command; a SetRepeat command; a SetCurrentPlayingTrack command; a RequestProtocolVersion command; a RequestMediaPlayerName command; a ResetDBSelection command; a SelectDBRecord command; a SelectSortDBRecord command; a RetrieveCategorizedDatabaseRecords command; a PlayStatusChangeNotification command; a PlayCurrentSelection command; and a PlayControl command.

39. The accessory of claim 20 wherein the interface includes a connector having a plurality of signal contacts arranged to be connectable to a plurality of signal contacts of a mating connector of the media player.

40. The accessory of claim 20 wherein the interface includes a wireless communication interface.

41. A method of operating an accessory, the method comprising:

establishing communication with a media player, wherein the accessory is configured to communicate with the media player using a media player accessory protocol comprising a general lingo including an identification command and at least one accessory lingo;

sending the identification command of the general lingo to the media player, wherein the identification command identifies an accessory lingo of the media player accessory protocol as being supported by the accessory; and providing authentication information to the media player using one or more accessory-authentication commands of the general lingo.

42. The method of claim 41 further comprising: obtaining authentication information from the media player using one or more media-player-authentication commands of the general lingo.

43. The method of claim 41 wherein the identification command of the general lingo indicates that the accessory supports exactly one of a plurality of accessory lingoes supported by the media player and identifies the one accessory lingo that the accessory supports.

44. The method of claim 41 wherein the identification command of the general lingo identifies one or more of a plurality of accessory lingoes supported by the media player as being supported by the accessory.

US 7,529,872 B1

35

45. The method of claim 44 wherein the plurality of accessory lingoos supported by the media player includes at least one lingo from a group consisting of a microphone lingo, a simple remote lingo, a display remote lingo, an extended interface lingo and an RF transmitter lingo.

46. A computer program product comprising a computer readable storage medium encoded with program instructions, which when executed by a processor in an accessory device cause the processor to execute a method comprising:

establishing a communication link to a media player, wherein the accessory is configured to communicate with the media player via the communication link using a media player accessory protocol comprising a general lingo and a plurality of accessory lingoos supported by the media player; and

sending an identification command of the general lingo to the media player via the communication link, wherein the identification command identifies at least one of the accessory lingoos supported by the media player as being supported by the accessory.

47. The computer program product of claim 46 wherein the identification command indicates that the accessory supports exactly one of the accessory lingoos supported by the media player and identifies the one accessory lingo that the accessory supports.

48. The computer program product of claim 46 wherein the identification command indicates that the accessory supports one or more of the accessory lingoos supported by the media player and identifies each of the accessory lingoos that the accessory supports.

49. The computer program product of claim 46 wherein the plurality of accessory lingoos supported by the media player includes at least one lingo from a group consisting of a microphone lingo, a simple remote lingo, a display remote lingo, an extended interface lingo, and an RF transmitter lingo.

36

50. A computer program product comprising a computer readable storage medium encoded with program instructions, which when executed by a processor in an accessory device cause the processor to execute a method comprising:

establishing communication with a media player, wherein the accessory is configured to communicate with the media player using a media player accessory protocol comprising a general lingo including an identification command and at least one accessory lingo;

sending the identification command of the general lingo to the media player, wherein the identification command identifies an accessory lingo of the media player accessory protocol as being supported by the accessory; and providing authentication information to the media player using one or more accessory-authentication commands of the general lingo.

51. The computer program product of claim 50 wherein the method further comprises:

obtaining authentication information from the media player using one or more media-player-authentication commands of the general lingo.

52. The computer program product of claim 50 wherein the identification command of the general lingo indicates that the accessory supports exactly one of a plurality of accessory lingoos supported by the media player and identifies the one accessory lingo that the accessory supports.

53. The computer program product of claim 50 wherein the identification command of the general lingo identifies one or more of a plurality of accessory lingoos supported by the media player as being supported by the accessory.

54. The computer program product of claim 50 wherein the plurality of accessory lingoos supported by the media player includes at least one lingo from a group consisting of a microphone lingo, a simple remote lingo, a display remote lingo, an extended interface lingo and an RF transmitter lingo.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,529,872 B1
APPLICATION NO. : 11/479894
DATED : May 5, 2009
INVENTOR(S) : Schubert et al.

Page 1 of 2

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On page 2, in column 1, under "U.S. Patent Documents", line 25, delete "Van Ryzint et al." and insert -- Van Ryzin et al. --, therefor.

On page 3, in column 1, under "U.S. Patent Documents", line 35, delete "Halloway et al." and insert -- Holloway et al. --, therefor.

On page 3, in column 2, under "Other Publications", line 27, delete "Colloquim" and insert -- Colloquium --, therefor.

On page 4, in column 1, under "Other Publications", line 29, delete "conten!" and insert -- content! --, therefor.

On page 4, in column 2, under "Other Publications", line 19, delete "Sepcifications" and insert -- Specifications --, therefor.

In column 4, line 62, after "Detect)" insert -- . --.

In column 6, line 21, delete "RID" and insert -- R_{ID} --, therefor.

In column 6, line 24, delete "RID" and insert -- R_{ID} --, therefor.

In column 6, line 27, delete "RID" and insert -- R_{ID} --, therefor.

In column 6, line 29, delete "RID" and insert -- R_{ID} --, therefor.

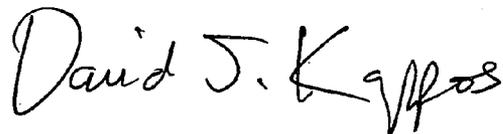
In column 6, line 35, delete "RID" and insert -- R_{ID} --, therefor.

In column 8, line 42, delete "player," and insert -- player. --, therefor.

In column 19, line 66, after "UI" insert -- . --.

Signed and Sealed this

Eighth Day of June, 2010



David J. Kappos
Director of the United States Patent and Trademark Office

CERTIFICATE OF CORRECTION (continued)
U.S. Pat. No. 7,529,872 B1

Page 2 of 2

In column 22, line 28 (within Table 7), delete “ReguestProtocolVersion” and insert -- RequestProtocolVersion --, therefor.

In column 22, line 29 (within Table 7), delete “ReguestMediaPlayerName” and insert -- RequestMediaPlayerName --, therefor.

In column 27, line 59, delete “IdentifyAccessoriesLingos” and insert -- IdentifyAccessoryLingoes --, therefor.

In column 31, line 64, in Claim 26, delete “commands” and insert -- command --, therefor.

In column 33, line 66, in Claim 38, delete “GetNumberCategorizedDB Records” and insert -- GetNumberCategorizedDBRecords --, therefor.

EXHIBIT G



US007580255B2

(12) **United States Patent**
Crooijmans et al.

(10) **Patent No.:** **US 7,580,255 B2**
(45) **Date of Patent:** **Aug. 25, 2009**

(54) **DOCKING STATION FOR HAND HELD ELECTRONIC DEVICES**

(75) Inventors: **Wim Crooijmans**, San Jose, CA (US);
Richard P. Howarth, San Francisco, CA (US);
Phillip M. Hobson, Menlo Park, CA (US)

D422,556 S 4/2000 Okura et al.
D427,970 S * 7/2000 Sage D13/118
6,115,247 A 9/2000 Helot
6,193,546 B1 * 2/2001 Sadler 439/534
D439,908 S 4/2001 Gozani
D440,542 S 4/2001 Hawkins et al.

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 465 days.

(Continued)

FOREIGN PATENT DOCUMENTS

EP 1 104 150 5/2001

(21) Appl. No.: **11/212,302**

(22) Filed: **Aug. 24, 2005**

(Continued)

(65) **Prior Publication Data**

US 2007/0047198 A1 Mar. 1, 2007

OTHER PUBLICATIONS

European Search Report dated Aug. 17, 2007 from EP Application No. 04 76 0202.

(51) **Int. Cl.**
G06F 1/16 (2006.01)

(Continued)

(52) **U.S. Cl.** **361/679.56**; 361/679.41;
710/303; 710/304

Primary Examiner—Jayprakash N Gandhi
Assistant Examiner—Anthony Q Edwards
(74) *Attorney, Agent, or Firm*—Townsend and Townsend and Crew LLP

(58) **Field of Classification Search** 361/679–686;
710/303, 304; 312/223.1, 223.2
See application file for complete search history.

(56) **References Cited**

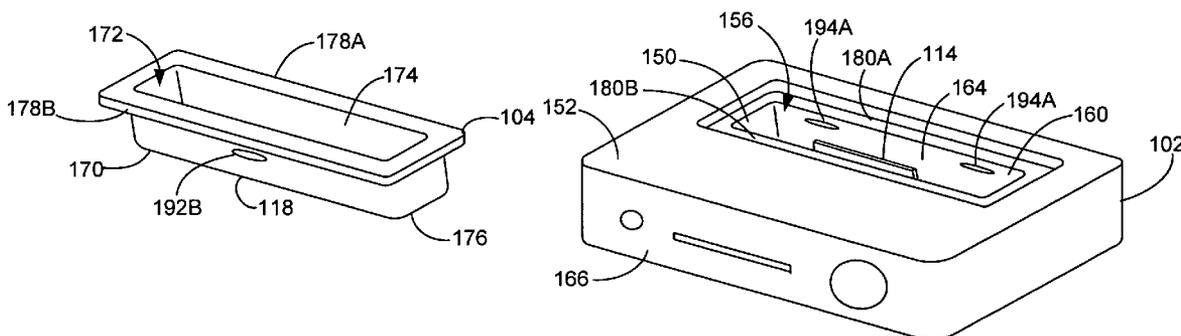
(57) **ABSTRACT**

U.S. PATENT DOCUMENTS

D328,277 S 7/1992 Leman et al.
D336,631 S 6/1993 Ivester
5,233,281 A * 8/1993 Chiang et al. 320/110
5,280,229 A * 1/1994 Faude et al. 320/110
5,357,185 A 10/1994 Chen
D388,764 S 1/1998 Bartling et al.
5,898,290 A 4/1999 Beard et al.
D409,987 S 5/1999 Okumura
D411,166 S 6/1999 Uemura et al.
5,926,005 A * 7/1999 Holcomb et al. 320/113
D416,536 S 11/1999 Ross et al.
D419,160 S 1/2000 Davidson et al.
6,042,414 A 3/2000 Kunert

A docking system is disclosed. The docking system includes a single docking station and a plurality of inserts that are placed in the docking station so that the docking station can accommodate hand held devices with differing sizes and shapes. The docking system is configured with a docking station and insert that matingly engage with one another via a rotating action. The docking system is also configured with a latching retention mechanism that secures the insert to the docking station when the insert is rotated into engagement with the docking station.

6 Claims, 8 Drawing Sheets



US 7,580,255 B2

Page 2

U.S. PATENT DOCUMENTS

D444,121 S 6/2001 Chun et al.
D444,124 S 6/2001 Casey et al.
D450,707 S 11/2001 Francavilla et al.
6,344,727 B1 2/2002 Desai et al.
D454,332 S 3/2002 Yokoyama
6,394,300 B1 5/2002 Bosy
D459,299 S 6/2002 Hughes et al.
D461,476 S 8/2002 Evers et al.
D463,361 S 9/2002 Ruohonen
D466,122 S 11/2002 Moody
6,483,698 B1 * 11/2002 Loh 361/686
D468,305 S 1/2003 Ma et al.
6,524,240 B1 2/2003 Thede
D472,900 S 4/2003 Matsumoto
6,716,058 B2 * 4/2004 Youn 439/535
D495,336 S 8/2004 Andre et al.
6,798,647 B2 9/2004 Dickie
6,813,528 B1 11/2004 Yang
6,898,080 B2 5/2005 Yin et al.
6,961,237 B2 11/2005 Dickie
6,994,575 B1 * 2/2006 Clark et al. 439/173
7,014,486 B1 * 3/2006 Wu et al. 439/248
D525,616 S 7/2006 Andre et al.
7,075,579 B2 * 7/2006 Whitby et al. 348/375
7,238,042 B2 * 7/2007 Chen et al. 439/374
D551,212 S 9/2007 Andre et al.
D551,213 S 9/2007 Andre et al.
D552,085 S 10/2007 Andre et al.
D558,738 S 1/2008 Andre et al.
D558,739 S 1/2008 Andre et al.
7,381,095 B2 * 6/2008 Freeman et al. 439/638
2002/0024794 A1 2/2002 Lin et al.
2002/0032042 A1 3/2002 Poplawsky et al.
2002/0103008 A1 8/2002 Rahn et al.
2002/0115480 A1 8/2002 Huang
2002/0163780 A1 11/2002 Christopher
2002/0173273 A1 11/2002 Spurgat et al.
2002/0186319 A1 12/2002 Whitby et al.
2003/0097379 A1 5/2003 Ireton
2003/0148740 A1 8/2003 Yau et al.
2003/0198015 A1 10/2003 Vogt

2004/0195305 A1 10/2004 Dotson
2004/0224638 A1 11/2004 Fadell et al.
2004/0267825 A1 12/2004 Novak et al.
2005/0047099 A1 * 3/2005 Mori et al. 361/752
2006/0181840 A1 * 8/2006 Cvetko 361/679
2006/0250764 A1 11/2006 Howarth et al.

FOREIGN PATENT DOCUMENTS

WO WO 00/60450 10/2000
WO WO 2004/084413 A2 9/2004
WO WO 2006080659 A1 * 8/2006

OTHER PUBLICATIONS

International Search Report dated May 21, 2007 from PCT Application No. PCT/US2006/048670.
Sinitsyn, Alexander, "A Synchronization Framework for Personal Mobile Servers," Pervasive Computing and Communications Workshops (PERCOMW'04). Proceedings of the Second IEEE Annual Conference, Piscataway, NJ, USA, IEEE, Mar. 14, 2004, pp. 208-212.
"iPod Dock Cradle", downloaded Apr. 11, 2003, www.bookendzdocks.com/bookendz/dock_cradel.htm.
"BookEndz Home Page", downloaded Apr. 11, 2003, www.bookendzdocks.com/bookendz/index.htm.
"Atech Flash Technology Iduo Ipod Dock and card Reader", www.the-gadgeteer.com, downloaded Nov. 21, 2005.
U.S. Appl. No. 29/237,097 entitled "Dock Insert" filed Aug. 24, 2005.
U.S. Appl. No. 29/237,111 entitled "Dock Insert" filed Aug. 24, 2005.
U.S. Appl. No. 29/237,112 entitled "Dock Insert", filed Aug. 24, 2005.
U.S. Appl. No. 29/237,233 entitled "Dock Station", filed Aug. 24, 2005.
U.S. Appl. No. 29/237,084 entitled "Docking Station", filed Aug. 24, 2005.
U.S. Appl. No. 10/423,490 entitled "Media Player System", filed Apr. 25, 2003.
U.S. Appl. No. 11/125,883 entitled "Universal Docking Station for Hand Held Electronic Devices", filed May 9, 2005.
Search/Examination Report from corresponding EP Application No. 06789375.0 dated Jan. 29, 2009, 6 pages.

* cited by examiner

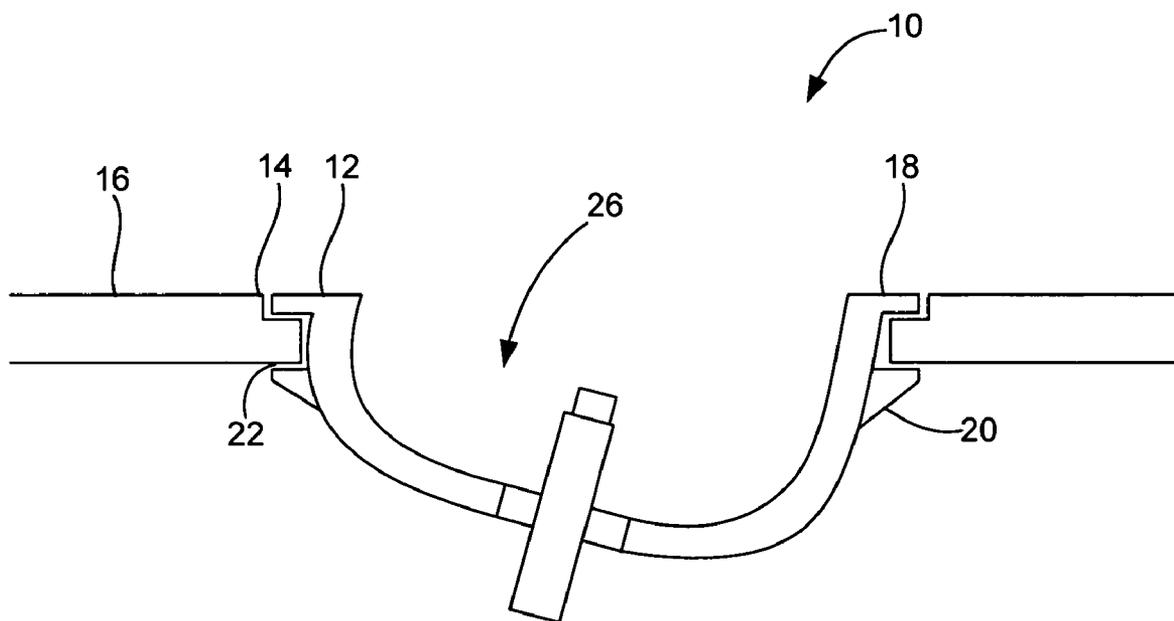


FIG. 1

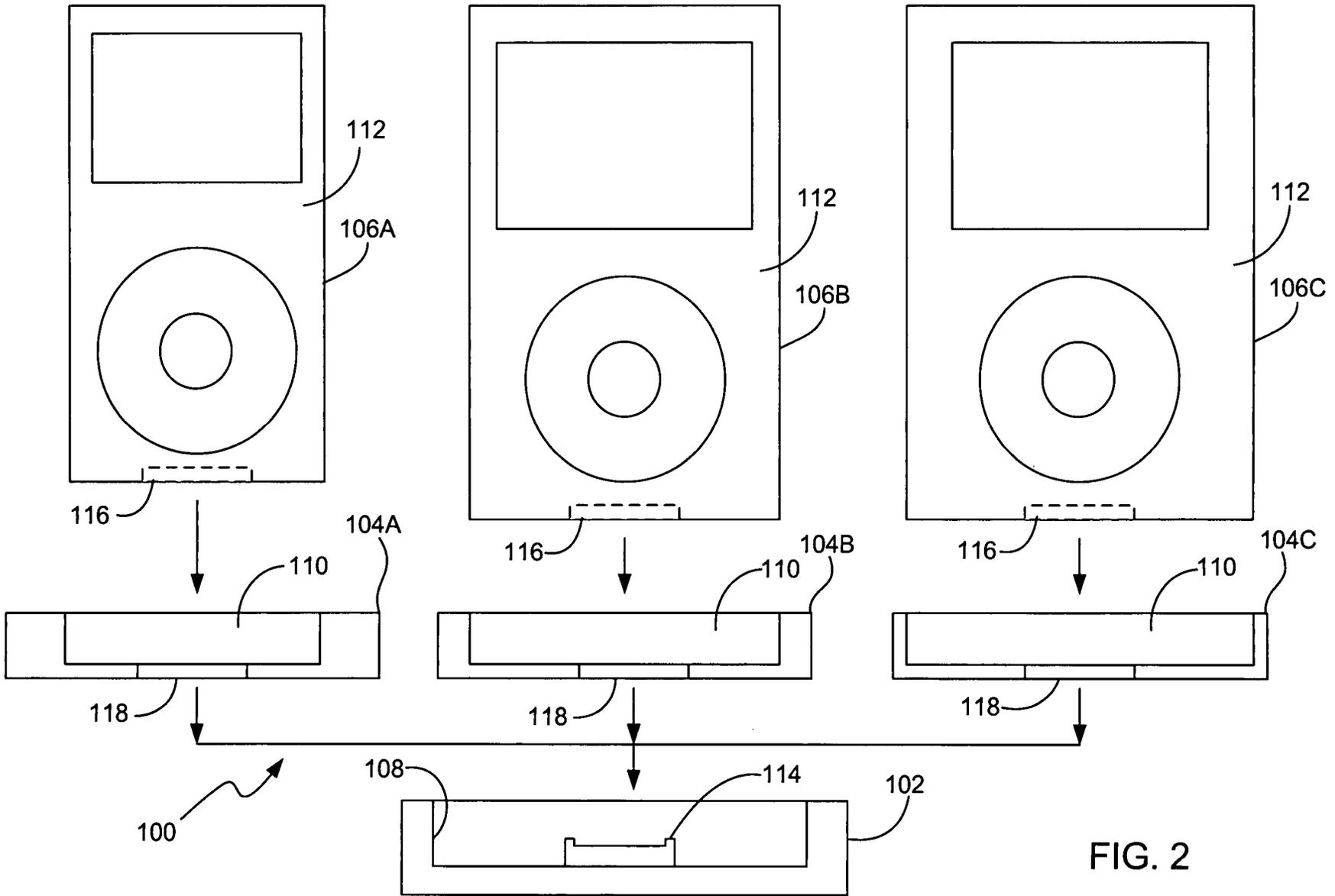


FIG. 2

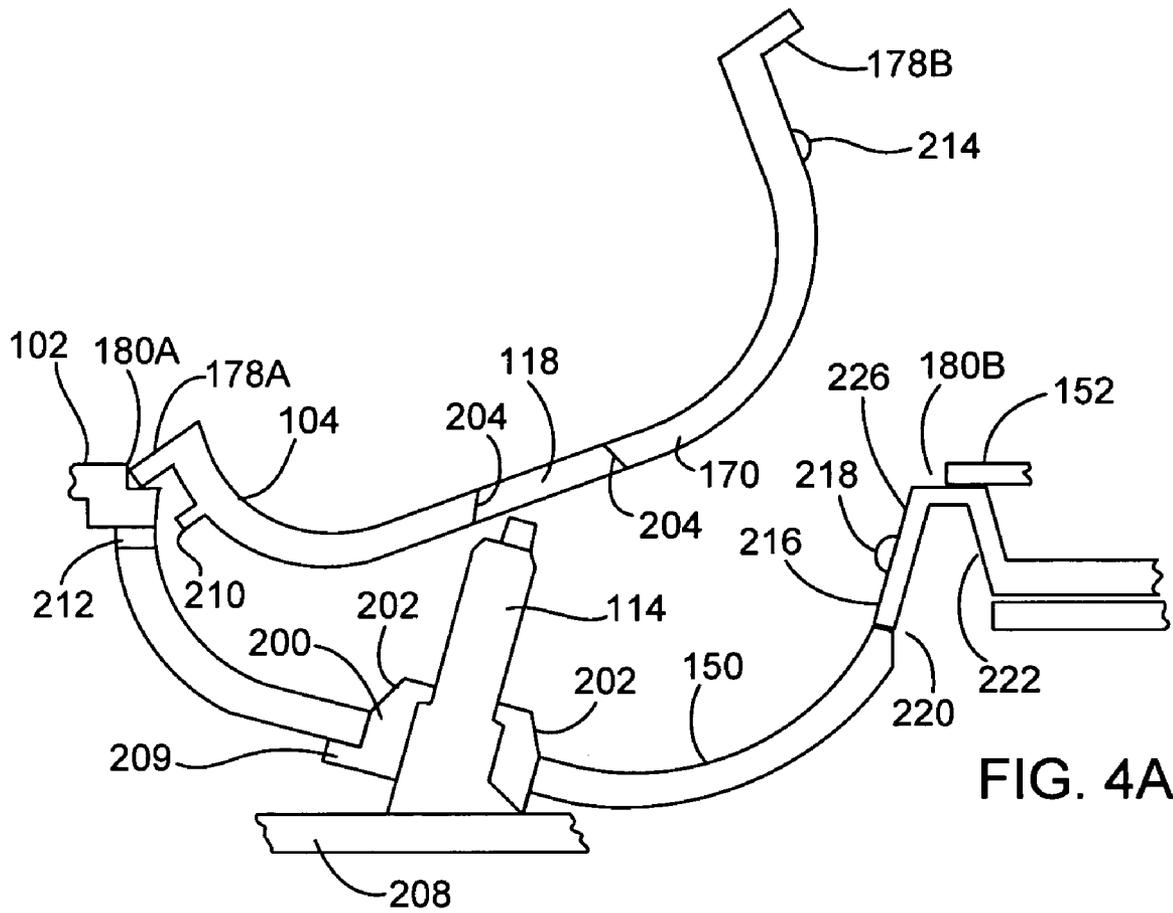


FIG. 4A

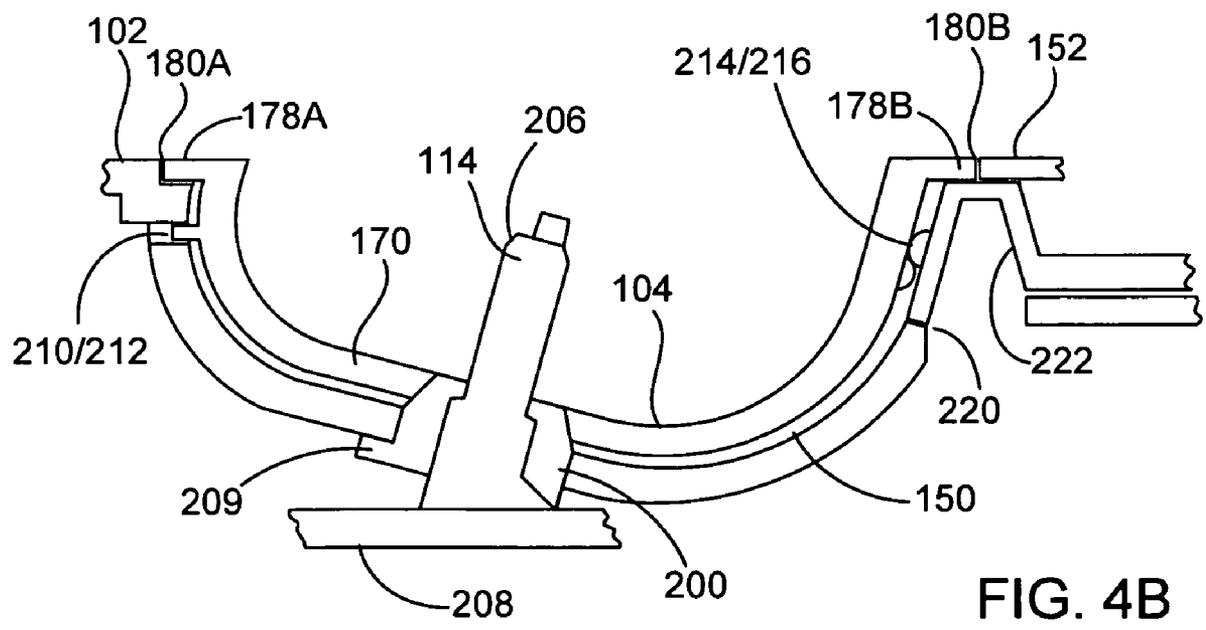


FIG. 4B

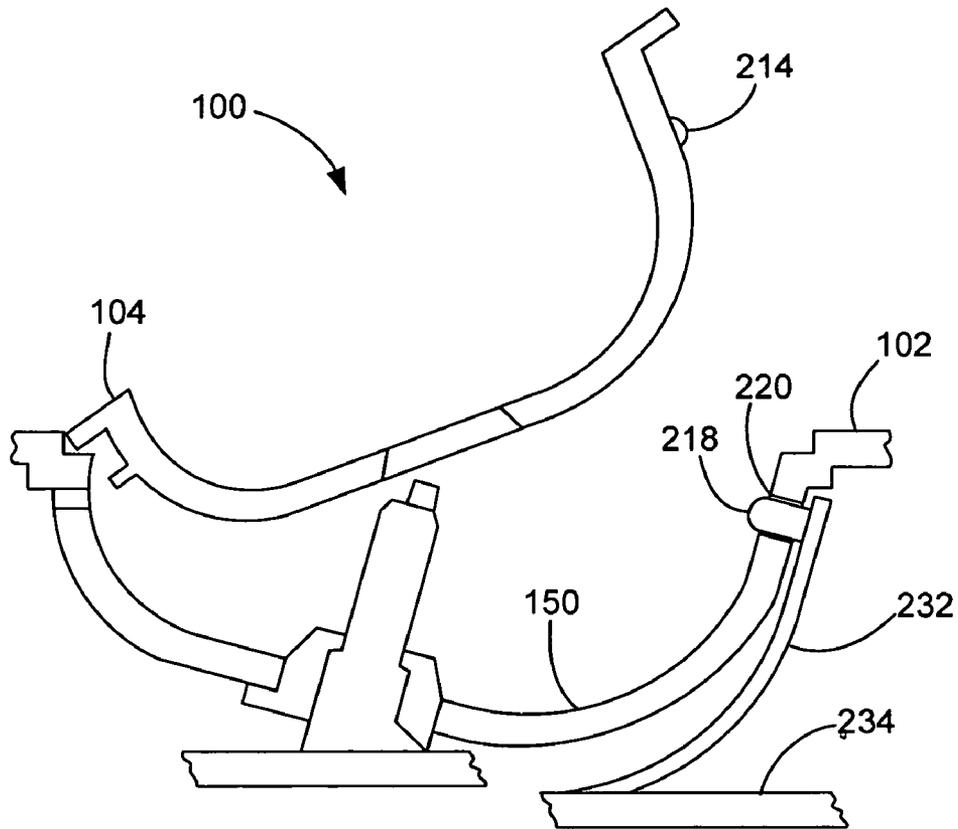


FIG. 5A

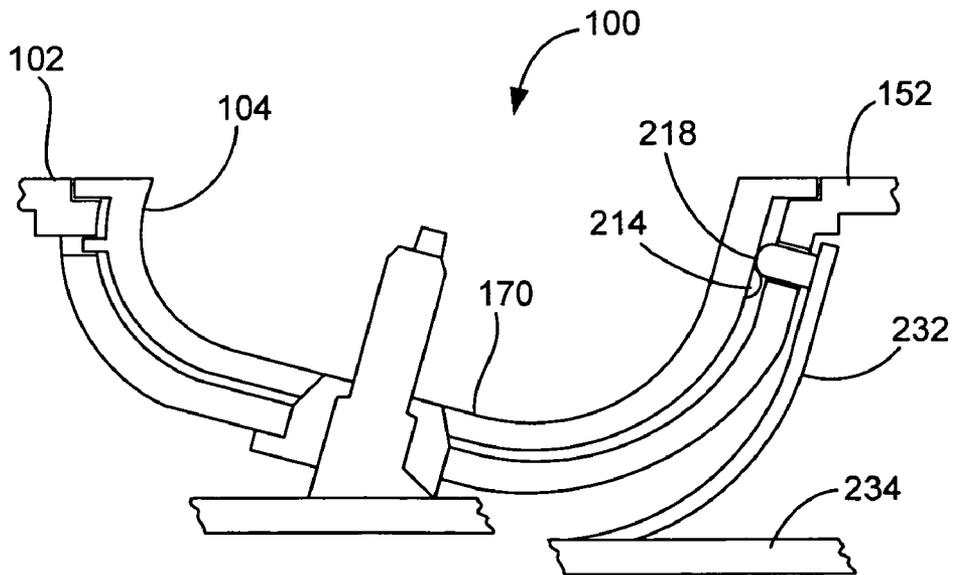


FIG. 5B

U.S. Patent

Aug. 25, 2009

Sheet 6 of 8

US 7,580,255 B2

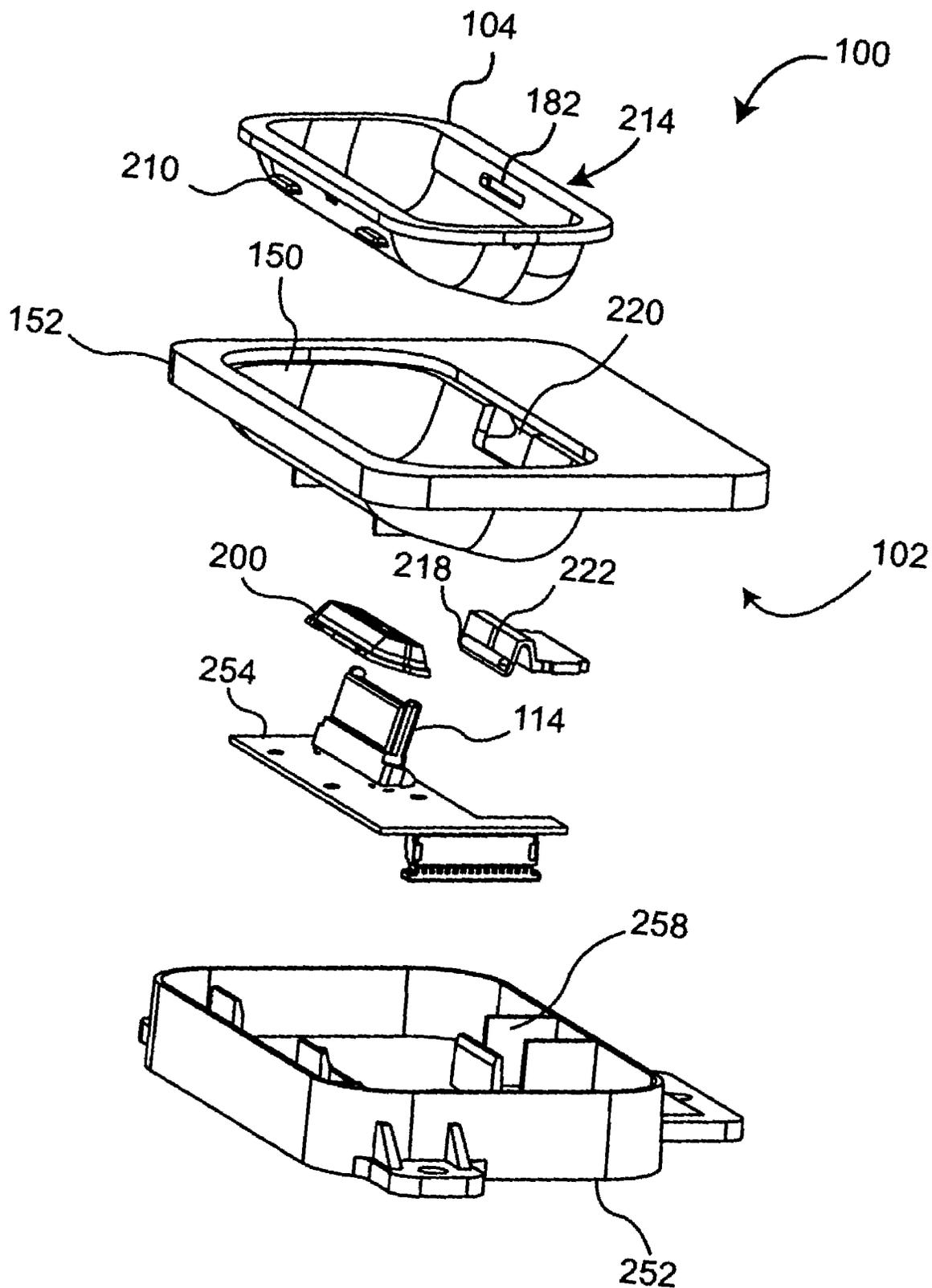


FIG. 6

U.S. Patent

Aug. 25, 2009

Sheet 7 of 8

US 7,580,255 B2

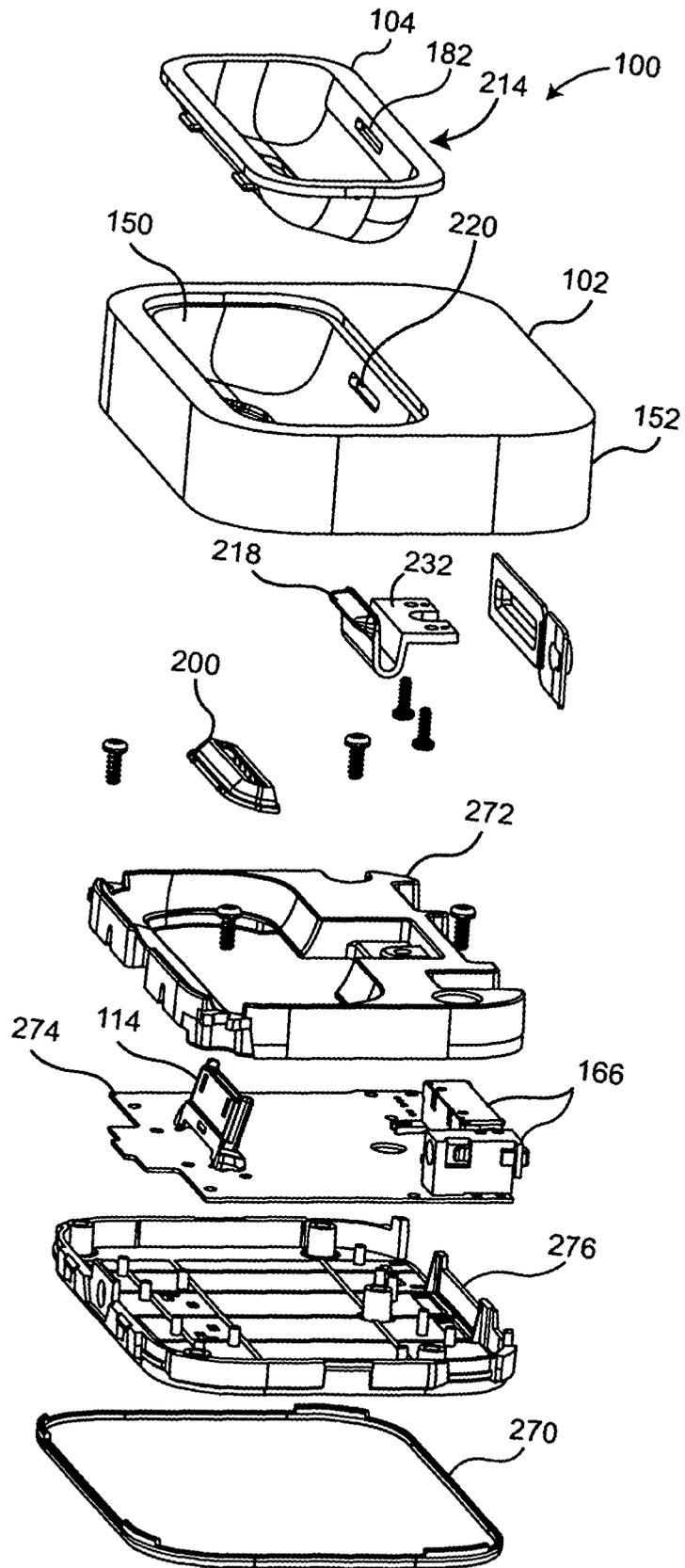


FIG. 7

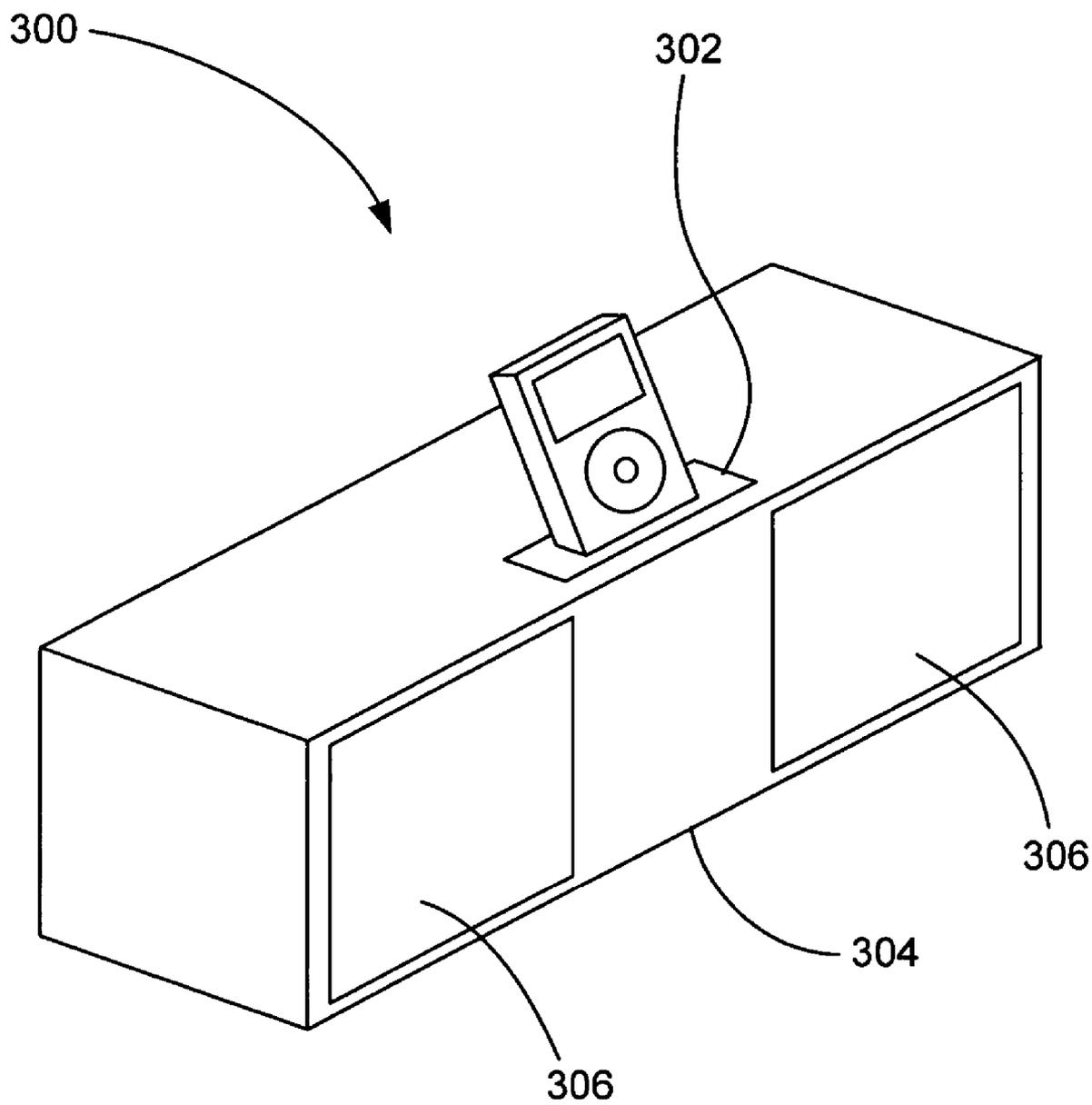


FIG. 8

US 7,580,255 B2

1

DOCKING STATION FOR HAND HELD ELECTRONIC DEVICES

CROSS REFERENCE TO RELATED APPLICATIONS

This application is related to U.S. patent application Ser. No. 10/423,490, titled, "MEDIA PLAYER SYSTEM," filed on Apr. 25, 2003, which is herein incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to docking stations for hand held electronic devices. More particularly, the present invention relates to a docking station that accommodates hand held electronic devices with different sizes and shapes.

2. Description of the Related Art

There exist today many styles of handheld electronic devices as for example cellular phones, personal digital assistants (PDA), media players (e.g., music player or video player), cameras, game player and the like. As is generally well known, these devices come in various shapes and sizes (e.g., thickness, width and height). The size and shape is typically dependent on various form factors including but not limited to ease of use, ergonomics, aesthetics, and the size of the components inside or outside the device. While it is likely that different manufacturers may produce the same type of hand held electronic device with differing shapes and sizes, it is also likely that one particular manufacturer may produce different models of the same type of hand held electronic device with different shapes and sizes.

By way of example, the iPod product line, which is manufactured by Apple Computer of Cupertino, Calif. comes in various sizes and shapes. Particularly, the iPod Mini has a pill shaped cross section and dimensions of 2.0×3.6×0.5 inches for 4 GB versions, while the standard iPod has a substantially rectangular cross and dimensions of 2.4×4.1×0.57 inches for a 20 GB version and 2.4×4.1×0.69 inches for a 40 GB version. Furthermore, the iPhoto model of the iPod has a substantially rectangular cross section similar to the standard version, but with dimensions of 2.4×4.1×0.75 for both 40 GB and 60 GB versions.

Docking stations provide a convenient interface for transferring data between the device and computing devices such as a personal computers or peripheral devices such as speakers, monitors and printers without having to reconnect and disconnect cables. The docking station may also provide an interface for connecting to a power source so that the hand held electronic device can be powered or charged (e.g., battery). In most cases, the docking stations include a slot within which the hand electronic device is received. The slot is configured to have a size and shape that coincides with the size and shape of the hand held device so that the hand held device rests snugly within the slot. Furthermore, the slot includes a connector therein for operatively engaging a port of the hand held electronic device when the hand held electronic device is positioned within the slot. The connector is coupled to the external systems through a cable so that communications between the hand held electronic device and the external systems can take place.

Because the slot has a size and shape that coincides with the size and shape of a particular hand held device, the docking station is typically dedicated to the particular handheld device. Other devices typically cannot be used with the docking station. The slots themselves are simply incapable of

2

supporting more than one device. As a result, users with an arsenal of devices have to wield a variety of docking stations, one for each device in their arsenal. This produces clutter and can be confusing to the user. This also presents problems to the manufacturer of multiple devices in that they need to produce a variety of docking stations (one for each unique device), which adds costs and complexity in the design, manufacture, assembly, and packaging of the devices. This is especially true with devices in the same product line as for example the iPod described above.

Some docking stations come packaged with removable spacers that can change the size and shape of the slot thereby accommodating more than one device.

FIG. 1 is an exemplary side elevation view of a docking station 10 that utilizes a removal spacer 12. The removable spacer 12 is positioned in an opening 14 in the housing 16 of the docking station 10, and includes a lip 18 and a bumper 20 disposed below the lip 18. The lip 18 and bumper 20, which protrude outwardly, completely surround the periphery of the spacer 12. The bumper 20 is formed from a deformable material such as rubber that can deform inwardly during insertion and extraction thereby allowing the bumper 20 to be pressed over an edge 22 of the opening 14. The peripheral lip 18 and the bumper 20 cooperate to form a channel that captures the edge 22 in order to secure the spacer 12 to the housing 16 of the docking station 10. Furthermore, the spacer 12 includes an opening 26 for receiving a connector 28 positioned in the opening 14 of the docking station 10.

Although spacers such as these work for their intended purpose, they still suffer from several drawbacks. For one, the spacer is difficult to insert and remove and has a rough feel, which leaves a negative impression on the user. This is due in part to the vertical positioning of the spacer within the docking station and the large force that is required to overcome the rubber bumper since it is somewhat rigid and extends completely around the spacer. Furthermore, in cases where the insert includes an angled basin, the hole in the bottom of the basin must be made large to allow enough space for the connector (e.g., clearance). This however, leaves gaps, which are aesthetically displeasing and which provide paths for dust and other foreign matter into the docking system.

An improved docking station that utilizes spacers is therefore desired.

SUMMARY OF THE INVENTION

The invention relates, in one embodiment, to a docking station for handheld devices with varying shapes and sizes. The docking station includes a top member with a cavity formed therein. The cavity is configured to rotatably receive a removable insert that accommodates the bottom end of a hand held device with a particular size and shape.

The invention relates, in another embodiment, to a removable insert for hand held devices with varying shapes and sizes. The removable insert includes a basin having an opening and an enclosed region that receives the bottom end of a hand held device with a particular size and shape. The basin is configured for rotatable insertion and extraction to and from a docking station. The removable insert rotates between an initial position, where the removable insert is capable of being inserted or extracted, and a mounted position, where the removable insert is secured in the docking station.

The invention relates, in another embodiment, to a docking system for a handheld device. The docking system includes a docking station. The docking system also includes a remov-

US 7,580,255 B2

3

able insert. The removable insert is configured to rotate relative to the docking station during insertion and extraction from the docking station.

The invention relates, in another embodiment, to a docking station. The docking station includes a top member with a cavity formed therein. The cavity is configured to rotatably receive a removable insert that accommodates the bottom end of a hand held device with a particular size and shape. The docking station also includes a connector disposed in the cavity. The connector protrudes through a bottom wall of the cavity. The docking station further includes a collar surrounding and positioned at the base of the connector. The collar protrudes through the bottom wall of the cavity with the connector. The collar is configured to produce a snug fit with an opening in the removable insert when the removable insert is positioned within the cavity of the top member. The opening in the removable insert is configured to allow rotation of the removable insert into the cavity without hitting the connector.

The invention relates, in another embodiment, to a removable insert for hand held devices with varying shapes and sizes. The removable insert includes a basin having an opening and an enclosed region that receives the bottom end of a hand held device with a particular size and shape. The basin is configured for insertion and extraction within a cavity of a docking station. The removable insert also includes a lip that surrounds and extends away from the opening in the basin. The lip is configured for placement in a recess in the docking station. The recess catches the lip and provides a reference point for the insert during insertion and extraction from the cavity of the docking station. The removable insert further includes one or more tabs extending from the front wall of the basin, and configured to mate with corresponding slots located inside the cavity of the docking station. The removable insert additionally includes a catch formed on the back wall of the basin, and configured to mate with a latch located within the cavity of the docking station.

The invention relates, in another embodiment, to a docking system. The docking system includes a docking station. The docking station includes a top member. The docking station also includes a pocket formed in the top member and configured to receive a plurality of inserts therein. The cavity includes an opening and an enclosed region that are sized to accommodate the largest insert. The enclosed region has sidewalls and a bottom wall that surround the inserts when they are placed in the cavity. The docking station further includes a recess formed in the top member and surrounding the opening in the pocket. The docking station additionally includes a connector located within the pocket and protruding through the bottom wall of the pocket. The docking station also includes a collar surrounding and positioned at the base of the connector. The collar protrudes through the bottom wall of the pocket with the connector.

The docking system also includes a plurality of removable inserts. Each of the removable inserts includes a basin configured for rotatable insertion into the pocket of the docking station. The basin includes an opening and an enclosed region that are sized to accommodate a particular hand held device. The enclosed region has sidewalls and a bottom wall that surround the bottom end of the particular hand held device. The bottom wall includes a hole for receiving the connector and collar when the basin is rotated within the pocket of the docking station. The edge of the hole includes a chamfer so that the basin clears the connector when the basin is rotated within the pocket of the docking station. Each of the removable inserts also includes a lip that surrounds and extends away from the opening in the basin. The lip is configured for

4

placement in the recess in the top member. The recess catches the lip and provides a reference point about which the insert is rotated when inserted and extracted from the pocket of the docking station.

The docking system further includes a retention mechanism configured to secure the inserts within the docking station. The retention mechanism includes one or more slots formed in a front wall of the pocket. The retention mechanism also includes one or more tabs extending from the front wall of the basin, and configured to mate with the slots. The retention mechanism further includes a catch formed on the back wall of the basin. The retention mechanism additionally includes a latch located on the back wall of the pocket and configured to move between a latched position, where the catch is captured by the latch, and an unlatched position, where the catch is released from the latch. The latch includes a detent that springs into engagement with the catch when the insert is rotated into its final position within the pocket of the docking station. The latch also includes a flexure that biases the detent towards the center of the pocket.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may best be understood by reference to the following description taken in conjunction with the accompanying drawings in which:

FIG. 1 is an exemplary side elevation view of a docking station that utilizes a removal spacer.

FIG. 2 is a simplified diagram of a docking system, in accordance with one embodiment of the present invention.

FIG. 3A is a front perspective view of a docking system, in accordance with one embodiment of the present invention.

FIG. 3B is a rear perspective view of the docking system shown in FIG. 3A, in accordance with one embodiment of the present invention.

FIGS. 4A and 4B are side elevation views of a docking system, in accordance with one embodiment of the present invention.

FIGS. 5A and 5B are side elevation views of a docking system, in accordance with one embodiment of the present invention.

FIG. 6 is a perspective view of a docking system, in accordance with one embodiment of the present invention.

FIG. 7 is a perspective view of a docking system, in accordance with one embodiment of the present invention.

FIG. 8 is a perspective view of a sound system with an integrated docking station, in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates generally to a docking station for portable electronic devices. The docking station provides a platform for quickly and easily coupling a hand held electronic device to another system or device as for example a computer, a power source, or peripheral devices such as a monitor, a keyboard, speakers, etc. A primary advantage of using a docking station is that the user does not have to separately connect one device at a time.

The docking station may be a stand alone unit that communicates with other devices or systems through wired (e.g., cables) or wireless (e.g., Bluetooth) connections, or alternatively, the docking station may be integrated directly into the other devices or systems. In either case, the docking station includes an opening or slot that receives the portable electronic device. The opening is typically configured to support the portable electronic device in an upright (e.g., vertical) or

US 7,580,255 B2

5

laid down (e.g., horizontal) position while maintaining access to the U.I. portion of the portable electronic device.

The dock also includes a way of interfacing the portable electronic device with the other devices or systems. By way of example, the opening may include one or more connectors that engage one or more ports on the portable electronic device when the portable electronic device is placed in the opening. In the case of a stand alone unit, the connector may be coupled to other connectors, ports or transceivers that provide external connections to the other devices or systems. In the case of an integrated docking station, the connector may be wired directly to the components of the host device.

More particularly, the present invention relates to docking stations that utilize removable inserts that accommodate portable hand held devices with different sizes and shapes. The docking station includes an opening that receives all the various inserts, and each of the inserts has a slot or opening that accommodates a particular hand held device. The number of inserts generally depends on the number of devices that are served by the docking station. One aspect of the invention relates to inserts that are rotated in and out of the docking station. Another aspect of the invention relates to a quick release latching mechanism that allows the insert to be temporarily secured to the docking station. Both features allow the inserts to be inserted and extracted into the docking station with simplicity, ease and minimal effort (thereby leaving a positive impression on the user).

Embodiments of the invention are discussed below with reference to FIGS. 2-8. However, those skilled in the art will readily appreciate that the detailed description given herein with respect to these figures is for explanatory purposes as the invention extends beyond these limited embodiments.

FIG. 2 is a simplified diagram of a docking system 100, in accordance with one embodiment of the present invention. The docking system 100 includes a single docking station 102 and a plurality of inserts 104. Each of the inserts 104 are received by the docking station 102 and are configured to accommodate hand held devices 106 with different sizes and shapes. In essence, the inserts allow different dimensioned devices to be placed in the same docking station. As shown, there is an insert 104A-C for each differently configured hand held device 106A-C. The hand held devices 106A-C may for example correspond to media players such as the iPod class of media players manufactured by Apple Computer, Inc. of Cupertino Calif.

In most cases, the docking station 102 includes an opening 108 for receiving each of the inserts 104 and the inserts 104 include an opening 110 for receiving a hand held device 106 with a particular size and shape. That is, the opening 108 in the docking station 102 is sized and dimensioned to allow each of the inserts 104 to be positioned therein, and the openings 110 in the inserts 104 are sized and dimensioned to allow a particular device to be positioned therein. Furthermore, the openings 108 and 110 may be configured to support the hand held device 106 in an upright (e.g., vertical or angled), or laid down (e.g., horizontal) position. When upright, the openings 108 and 110 are typically configured to surround the front, back and left and right sides of the hand held device 106 and when laid down, the openings 108 and 110 are typically configured to surround the top and bottom sides and the left and right sides of the hand held device 106. In either case, access to the U.I. portion 112 of the hand held device 106 is preferably maintained.

When coupled with an insert 104, the hand held device 106 is configured to correctly interface with the docking station 102 in order to allow communications between the hand held device 106 and other devices or systems connected to the

6

docking station 102. The docking station 102 may for example include a connector 114 inside the opening 108 that couples to a port 116 on the hand held device 106 through a hole 118 in the insert 104. The opening 110 in the insert 104 is configured to guide the port 116 of handheld device 106 into alignment with the connector 114 that is disposed within the hole 118.

The docking station 102 may be a stand alone unit that communicates with other devices or systems through wired (e.g., cables) or wireless (e.g., Bluetooth) connections, or alternatively, the docking station 102 may be integrated directly into the other devices or systems. An example of a docking station that can be used may be found in U.S. patent application Ser. No. 10/423,490, which is herein incorporated by reference.

Referring to FIGS. 3A and 3B, one embodiment of the docking system 100 will be described in greater detail. In this embodiment, the docking station 102 is a stand alone unit. It should be appreciated, however, that the features of the stand alone docking station may be equally applied to integrated docking stations.

As shown, the docking station 102 includes a cavity 150 disposed in a top member 152. The cavity 150 is a depression or pocket capable of receiving each of the various inserts 104 therein (and possibly the largest hand held device). The cavity includes an opening 156 and an enclosed region having side-walls 160 and a bottom wall 162. The insert 104 is placed within the opening 156 and rests inside the cavity 150 where it is surrounded by the various walls 160 and 162. The size and shape of the cavity 150 is generally configured to accommodate the largest insert 104. The top member 152, on the other hand, is a portion of a housing 154 of the docking station 102.

The connector 114 is located within the cavity 150, and typically protrudes from the bottom wall 162 of the cavity 150. The connector 114 may for example be attached to a PCB and extend through a hole 164 in the bottom wall 162. In integrated systems, the connector 114 is typically hard wired to the components of the host system (e.g., boom box). In stand alone systems, the connector 114 may electrically connect to various I/O interfaces 166 located on the side, and more particularly the backside of the docking station 102 so that the docking station 102 can communicate with other devices and systems. By way of example, the I/O interfaces may include audio and video jacks, data ports, electrical outlets, and/or the like.

Referring to the inserts 104, the inserts 104 include a basin 170 for receiving the bottom end of the hand held device. Similar to the cavity 150, the basin 170 includes an opening 172 and an enclosed region formed by side walls 174 and a bottom wall 176. However, unlike the cavity 150, the opening 172 and walls 174 and 176 are sized and dimensioned to correspond to a particular hand held device. That is, the inner periphery of the basin 170 coincides with the outer periphery of a particular hand held device. Furthermore, the basin 170 has a depth that supports the handheld device in an upright position while still providing access to the user interface of the hand held device when the hand held device is disposed within the insert 104 and docking station 102. Moreover, the depth and outer periphery of the basin 170 is smaller than the depth and inner periphery of the cavity 150 so that the basin 170 can be seated properly within the cavity 150.

The basin 170 may be vertical or angled. In the illustrated embodiment, the basin 170 is angled so that the hand held device is placed in a tilted orientation when situated in the docking station 102. By way of example, the angle may be about 15 degrees from vertical.

US 7,580,255 B2

7

The hole 118 for receiving the connector 114 is disposed within the bottom wall 176 of the basin 170. The connector 114 extends through the hole 118 and into the basin 170 where it eventually plugs into a port on the bottom end of the hand held device. In most cases, the connector 114 is positioned parallel to the side walls 174. Thus, the connector 114 has the same angle as the basin 170. In fact, the side walls 174 of the basin 170 are used to guide the hand held device over and into engagement with the connector 114.

The insert 104 may be supported by the walls of the cavity 150 and/or the top member 152 of the docking station 102. In the illustrated embodiment, the insert 104 includes a lip 178 which surrounds the opening 172 in the basin 170 and which rests on the top member 152 of the docking station 102. More particularly, the top member 152 includes a recess portion 180 that surrounds the opening 156 of the cavity 150, and that receives the lip 178 therein. The lip 178 typically has an outer profile that coincides with the inner profile of the recess portion 180. This is done to reduce gaps between the insert 104 and the docking station 102 and provide mating surfaces for positioning and supporting the insert 104 relative to the docking station 102, i.e., helps align the hole 118 with the connector 114. Furthermore, the depth of the recess portion 180 typically coincides with the thickness of the lip 178 so that the top surface of the insert 104 lies flush with the top surface of the top member 152 when the insert 104 is seated within the docking station 102. This particular arrangement provides a clean and continuous appearance that is aesthetically pleasing to the user. This may also help hide the fact that an insert is being used.

In accordance with one embodiment of the present invention, the insert 104 is positioned within the docking station 102 via rotating action. That is, the insert 104 is configured to rotate relative to the docking station 102 between a released position and a mounted position. The rotating action provides a more elegant solution for inserting and removing the insert 104 to and from the docking station 102.

During insertion, a user places the front lip 178A of the insert 104 within the front recess 180A of the docking station 102 and rotates the insert 104 about this interface. The recess 180A catches the front lip 178A and provides a reference surface about which the insert 104 can be rotated. At the end of the rotating action, the entire lip 178 is positioned within the recess 180 thereby placing the insert 104 in the desired relationship with the docking station 102. That is, the basin 170 is placed within the cavity 150, and the connector 114 extends through the hole 118 in the bottom of the basin 170.

During extraction or removal, a user lifts the back end of the insert 104 away from the docking station 102 and rotates the insert 104 about the lip/recess interface. As the insert 104 is rotated, the basin 170 is pulled out of the cavity 150 and the connector 114 is removed from the hole 118. At the end of rotation, the insert 104 is freed from the docking station 104 and thus it can be removed from the docking station 102. In some cases, the back end of the insert 104 may include a void 182 that forms a lip that can be used as a handle to lift the insert 104 away from the docking station 102.

In accordance with another embodiment, the docking system 100 includes a retention mechanism for securing the insert 104 to the docking station 102. The retention mechanism consists of at least two features, an insert feature 192 and a docking station feature 194. These features are cooperatively positioned so that when the insert 104 is inserted into the docking station 102 (e.g., via the rotating action), the features 192 and 194 engage with one another thus securing the insert 104 to the docking station 102. In most cases, the features 192 and 194 are strategically positioned about the

8

periphery of the basin 170 and the cavity 150. For example, the features 192 and 194 may be located on the side walls including the front, back, right and left walls. Furthermore, the mating features 192 and 194 are typically positioned in an opposed relationship. For example, if there are mating features 192 and 194 on the front wall then there tend to be mating features 192 and 194 on the back wall. The features may be widely varied. By way of example, the features may include nubs, grooves, channels, catches, hooks, slots, guides, tabs and/or the like that mate with one another in order to hold the insert 104 in place. The features 192 and 194 may also include more complex mechanisms such as latches, locks, snaps and/or friction couplings.

In the illustrated embodiment, the front side of the insert 104 includes one or more tabs or hooks 192A that mate with one or more slots 194A located in the front side of the cavity 150 (or vice versa). Further, the backside of the insert 104 includes a catch 192B that mates with a latch 194B located in the backside of the cavity 150. The latch 194B is configured to move between a latched and unlatched position. In the latched position, the latch 194B captures the catch 192B thereby securing the insert 104 to the docking station 102. In the unlatched position, the catch 192B is released from the latch 194B thereby allowing the insert 104 to be removed from the docking station 102. The catch may be widely varied. For example, the catch may be selected from voids, ribs, pockets, openings, etc.

During insertion, as for example when the insert 104 is rotated inwards, the slots 194A are configured to capture the tabs 192A and the latch 194B is configured to capture the catch 192B thereby securing the insert 104 to the docking station 102. The catch 192B may for example be captured by the latch 194B by moving the latch 194B from the unlatched to the latched position during or after the rotating action.

During extraction, as for example when the insert 104 is rotated outwards, the catch 192B is configured to disengage from the latch 194B, and the tabs 192A are configured to disengage from the slots 194A thereby releasing the insert 104 from the docking station 102. The catch 192B may for example be disengaged from the latch 194B by moving the latch 194B from the latched to the unlatched position before or during the rotating action.

In some cases, the insert 104 may additionally include a pull or handle 182 for helping disengage the catch from the latch and/or to help remove the insert from the cavity. The pull 182 provides a surface that a user can grab in order to lift the insert away from the cavity. By way of example, the pull 182 may be a recess or a ridge, both of which can be grasped by a finger of the user.

Referring to FIGS. 4A and 4B, one embodiment of the docking system 100 will be described in greater detail. In this embodiment, the insert 104 is configured to rotate into and out of the docking station 102. For example, the insert 104 may be rotated between a released position (FIG. 4A) and a latched position (FIG. 4B). In the released position, the insert 104 may be removed from the docking station 102. In the latched position, the insert 104 is secured to the docking station 102. As mentioned above, during rotation, the insert 104 pivots about an axis created at the interface between the front lip 178A and the front recess 180A. That is, the edge of the lip 178A abuts the edge of the recess 180A thereby forcing the insert 104 to rotate about this point. As the insert 104 is rotated, the basin 170 is inserted into the cavity 150 and the remaining portions of the lip 178 come into contact with the remaining portions of the recess 180 until the back lip 178B and back recess 180B are fully engaged. The back recess 180B serves as an abutment stop to the rotating insert 104. In

its final position, the lip 178 and thus the insert 104 is supported by the recessed portion 180 of the docking station 102.

Because the insert 104 is rotated, the hole 118 in the bottom of the basin 170 is made larger than the connector 114 so that there is enough clearance when the insert 104 is rotated into position within the docking station 102. The larger hole 118 provides sufficient space for the hole 118 to rotate around the connector 114, i.e., the connector does not impede the motion of the insert. Unfortunately, however, this leaves unsightly gaps between the edges of the hole 118 and the connector 114 when the insert 104 is finally positioned in the docking station. In order to correct this problem, the connector 114 is configured with a collar or sleeve 200 that has an outer periphery that coincides with the inner periphery of the hole 118. The collar 200 fills any gaps that would normally occur between the edge of the hole 118 and the connector 114 thereby making the docking station 102 more aesthetically pleasing while preventing particle accumulation.

As shown, the collar 200 is positioned at the base of the connector 114 and has a height that keeps it from interfering with the rotating insert 104. The collar 200 is configured to surround the connector 114, and in most cases, the inner periphery of the collar 200 coincides with the outer periphery of the connector 114. The collar 200 may include a chamfer 202 that matingly engages a corresponding chamfer 204 on the edges of the hole 118. The chamfers 202 and 204 help seat the insert 104 properly with respect to the connector 114. The chamfer 204 also provides more clearance between the rotating insert 104 and the connector 114 (without the chamfer 204 the insert 104 may not be able rotate past the connector 114 making removal or insertion more difficult). The connector 114 itself may further include a tapered edge 206 to provide extended clearance between it and the edge of the hole 118 when the insert 104 is rotated into place within the docking station 102.

When assembled, the connector 114 is positioned on a PCB 208, and the collar 200 is placed over the connector 114. Thereafter, the top member 152 is set in place over the connector 114 thereby trapping the collar 200 between the connector 114 and the bottom wall of the cavity 150. The connector 114 and collar 200 are inserted through an opening in the bottom wall of the cavity 150. The outer periphery of the collar generally coincides with the inner periphery of the opening in the bottom wall (e.g., tight fit). In order to keep the collar 200 trapped, the collar 200 may include a lip 209 that protrudes outwardly and that engages the back surface of the bottom wall.

In order to secure the insert 104 in the cavity 150, the docking system 100 also includes a latching/retention mechanism. In one embodiment, the insert 104 includes one or more tabs 210 along the front surface underneath the lip 178A. The tabs 210 are configured to matingly engage corresponding slots 212 located along the front wall of the cavity 150 underneath the recessed portion 180A. The insert 104 also includes a rib 214 that protrudes along and away from the back surface of the insert 104 underneath the lip 178B. The rib 214 is configured to receive a latch 216 located on the back surface of the cavity 150 underneath the recessed portion 180B. When placed in mating engagement, the tabs 210 and latch 216 cause the insert 104 to be secured to the docking station 102. The depth of the tab/slot interface and rib/latch interface is generally placed so that the lip 178 is placed within the recess 180 when the insert 104 is fully inserted in the docking station 102.

As shown, the latch 216 includes a detent 218 that springs above the rib 214 when the insert 104 is rotated into position within the docking station 102. The detent 218 is disposed

through an opening 220 in the backside of the cavity 150 and is biased towards the center of the cavity 150 by a spring action. The detent 218 may for example be attached to a flexure 222 that biases the detent 218 to a forward position. When the insert 104 is rotated into the docking station 102, the rib 214 engages the detent 218 of the docking station 102 and forces the detent 218 to a rearward position away from the center of the cavity 150 (via a cam action). When the detent 218 moves past the rib, the detent 218 resumes its forward position thereby trapping the rib 214 underneath the detent 218. In order to remove the insert 104, a user simply pulls up on the insert 104. During the pulling action, the detent 218 slides against the edges of the rib 214. When a significant pulling force has been provided, the detent 218 moves to the rearward position thereby releasing the rib 214 from the detent 218. Once released, the insert 104 is free to rotate out of the cavity 150 of the docking station 102.

To elaborate, the detent 218 is attached or integrally connected with the flexure 222. The flexure 222 is disposed within the opening 220 in the backside of the cavity 150 and is attached to a structural element 224 of the docking station 102 such as a housing component or a framing component. The flexure 222 is spring biased in a forward position so that the front face 226 of the flexure 222 is positioned through the opening 220 and lies substantially flush with the surface of the back wall of the cavity 150. The detent 218 is located on the front face 226 and protrudes substantially outward from the front face 226. The detent 218 may be rounded to provide a smooth cam action with the bottom of the basin 170 and eventually with the rib 214. The rib 214 may also be rounded for at least the same reasons.

When the insert 104 is rotated into the docking station 102, the outer surface of the basin 170 engages the detent 218 and forces the detent 218 to a rearward position away from the center of the cavity 150 against the spring action of the flexure 222. That is, the flexure 222 bends backwards under the cam force thereby allowing the detent 218 to move out of the cavity 150 and into the interior of the docking station 102. When the rib 214 is positioned underneath the detent 218, the flexure 222 resumes its natural forward position thereby trapping the rib 214 below the detent 218. As a result, the insert 104 is prevented from moving out of the docking station 102 on its own.

In order to remove the insert 104, a user simply pulls up on the insert 104, and when a significant pulling force has been provided, the flexure 222 flexes thereby releasing the rib 214 from the detent 218. Once released, the insert 104 is free to rotate out of the docking station 102. Using this arrangement, the user simply has to overcome the spring bias at the detent/rib interface when rotating the insert 104 in and out of the docking station. When designed appropriately, the insert can be inserted and extracted with simplicity, ease and minimal effort.

It is generally believed that the greater the spring force, the greater the holding force between the insert 104 and the docking station 102. If the force is too great, however, the insert 104 may be difficult to remove. The spring force is therefore designed to provide limited holding power. For example, enough holding power to secure the insert 104 to the docking station 102 while still allowing a user to pull and push the insert 104 into and out of the docking station 102. One advantage of this configuration is that the insert 104 is not locked or snapped in and therefore it may be easily pulled out and pushed into the docking station 102 during the rotating action, i.e., the insert 104 simply rotates in and out.

In one embodiment, the retention mechanism includes a pair of tabs 210 and a pair of corresponding slots 212 at the

US 7,580,255 B2

11

front interface, and a single rib **214** and a corresponding latch **216** at the back interface. The pair of tab/slot interfaces are typically spatially separated an equal distance from the center of the docking station **102**, and the rib/latch interface is typically placed in the center of the docking station **102**. As should be appreciated, this triangulation provides three point stability between the insert **104** and the docking station **102**.

It should be noted that the rib for interfacing with the detent is not a limitation. Alternatively, the insert may include a recess, slot or void for matingly receiving the detent. In each of these cases, the detent becomes trapped within the space provided by the recess, slot or void via the force of the flexure thereby securing the insert inside the opening. In one implementation, the rib is a bottom wall of a pocket that further includes a top wall and side walls thereby forming a void that extends away from the back surface of the insert. This particular implementation provides greater stability to the rib when long lengths are used as for example when the depth of the basin is small.

Referring to FIGS. **5A** and **5B**, another embodiment of the docking system **100** will be described in greater detail. This embodiment is similar to the embodiment shown and described in FIGS. **4A** and **4B**, however, unlike that embodiment, only the detent **218** is disposed through the opening **220** in the backside of the cavity **150** (not the flexure **222**). The opening **220** is sized and dimensioned to only receive the detent **218** therethrough. Furthermore, the detent **218** is integrated with an elongated flexure **232** (or spring arm) that extends along the interior of the back wall of the cavity **150** where it attaches to a structural component **234** such as a housing member or framing member. The structural element **234** may for example be a bottom member of the docking station **102**. Similar to the embodiment described in FIG. **4**, the flexure **232** is capable of bending so as to allow the detent **218** to move inward and outward through the opening **220** when the insert **104** is rotated in and out of the docking station **102**. Alternatively, the flexure **232** may be attached to the top housing **152**.

FIG. **6** is an exploded perspective view of a docking system that is capable of being integrated into another device, in accordance with one embodiment of the present invention. The docking system may for example correspond to the docking system described in FIGS. **4A** and **4B**. As mentioned previously, integrated docking stations are integrated or built into other devices such as personal computers, sound systems, video systems, recording equipment, communication equipment and peripheral devices such as printers, keyboards and the like. In the integrated docking station, the base is typically provided by the housing of the device that it is integrated with, i.e., the docking station utilizes the housing of the device rather than having its own housing.

In this embodiment, the docking station **102** includes a top member **152** and a frame **252**. The top member **152** is typically located at an outer surface of the housing of the host device, and the frame **252** is typically located inside the housing of the host device. The top member **152** may be an integral portion of the housing of the host device or it may be a separate component that is positioned within the outer surface of the host device. In either case, the top member forms a portion of the housing of the host device, and is generally a portion of the top surface of the host device. The frame **252** may also be an integral portion of the housing of the host device or it may be a separate component that is seated inside and fixed to the housing of the host device. For example, if not integrally formed therewith, the frame **252** may be attached to the housing of the host device via glues, fasteners, welds, snaps, and/or the like.

12

The top member **152** is positioned above and attached to the frame **252**. The manner in which the top member **152** is attached to the frame **252** may be widely varied. For example, the top member **152** may be attached to the frame **252** (or vice versa) via glues, fasteners, welds, snaps, and/or the like. In one implementation, the top member **152** is attached to the frame **252** via a snapping mechanism. The top member **152** may therefore be attached to the frame **252** via a snap action, i.e., snaps into place. The snapping mechanism typically includes a snap feature **253A** on the top member **152** that engages a corresponding snap feature **253B** on the frame **252**. In some cases, the snap features **253** may also serve as stiffening elements for both the top member **152** and the frame **252**.

The frame **252** is configured to support the flexure **222** that is disposed in the opening **220** of the top member **152**. By way of example, the flexure **222** may be attached to a support platform **258** of the frame **252**. The support platform **258** provides a reference surface for correctly placing the flexure **222** relative to the top member **152**. Any suitable attachment technique may be used, including for example, fasteners, glues, welds, snaps and/or the like. The frame **252** is also configured to support a printed circuit board (PCB) **254**. The PCB **254** includes the connector **114** and various electronic components for operating the docking system and delivering signals between the docking system and the host device. Any suitable attachment technique may be used, including for example, fasteners, glues, welds, snaps and/or the like.

It should be noted that the embodiment shown in FIG. **6** is not a limitation and that the configuration of the integrated docking system may be widely varied. By way of example, in some cases, a frame may not be used. In cases such as this, the PCB and flexure may be attached to the top member.

FIG. **7** is an exploded perspective view of a stand alone docking system, in accordance with another embodiment of the present invention. By way of example, the docking system may generally correspond to the docking system described in FIGS. **5A** and **5B**. The stand alone docking system generally includes a docking station that is separate from the host device. The docking station generally communicates with the host device via wired or wireless connections. For example, the docking station may be connected to the host device via a cable.

In this embodiment, the docking station **102** includes a top member **152** and a bottom member **276** that form the housing of the docking station **102**. The top member **152** and bottom member **276** when assembled are configured to enclose internally various electrical and structural components. The top member **152** and bottom member **276** when assembled are also configured to define the shape or form of the docking station **102**. The shape may for example be substantially rectangular (as shown) or alternatively the shape may be circular, cubical and the like. By way of example, the top and bottom members **152** and **270** may be attached using any suitable technique including for example screws, bolts, adhesives, welds, latches, snaps, etc. In the illustrated embodiment, the top and bottom members are attached via a snap mechanism so as to preserve the outer surfaces of the docking station. In this embodiment, the sides of the top member are slide over the sides of the bottom member so that the snap features can engage one another.

The docking station also includes various components that are located inside the assembled top and bottom members. The docking station may for example include a steel weight **272** for helping stabilize the docking station, and a printed circuit board (PCB) **274** for supporting the electronic circuitry of the docking station including for example the con-

US 7,580,255 B2

13

necter **114** as well as the various jacks and ports **166** of the docking station. The docking station also encloses the flexure **232**. Unlike the embodiment shown in FIGS. **5A** and **5B**, however, the flexure **232** is attached to the top member **152** and positioned so that the detent **218** extends into the cavity **150** of the insert **104**. Any suitable attachment technique may be used, including for example, fasteners, glues, welds, snaps and/or the like. In the illustrated embodiment, screws are used. Alternatively, the flexure may be integrally formed with the top member.

It should be noted that the embodiment shown in FIG. **7** is not a limitation and that the configuration of the stand alone docking system may be widely varied. By way of example, in some cases the flexure may be attached directly or indirectly to the bottom member **276** rather than to the top member. In addition, the docking system may include additional components such as rubber foot **270** that is attached to the bottom surface of the bottom member (e.g., via glue).

FIG. **8** is a perspective view of a sound system **300** with an integrated docking station **302**, in accordance with one embodiment of the present invention. The integrated docking station **302** may for example correspond to the one described in FIG. **6**. The sound system **300**, on the other hand, is a boom box or portable music system that includes a base **304** and a pair of speakers **306**. The docking station **302** is integrated within the base **304** and allows various hand held electronic devices especially a music player such as the iPod to communicate with the internal circuitry of the base **304** so that music stored on the music player can be broadcast on the boom box via the speakers **306**. The docking station **302** can be placed anywhere on the base **304**. In the illustrated embodiment, however, the docking station **302** is placed on the top surface of the base **304**.

In each of the various embodiments described above, the components of the docking station may be formed from a variety of materials. By way of example, plastic materials such as polycarbonate (PC), ABS, or PC-ABS may be used for the housing, the insert and the latching mechanism such as the flexure and detent. Alternatively, the flexure may be formed from a metal material such as spring steel.

While this invention has been described in terms of several preferred embodiments, there are alterations, permutations, and equivalents, which fall within the scope of this invention. It should also be noted that there are many alternative ways of implementing the methods and apparatuses of the present invention. It is therefore intended that the following appended claims be interpreted as including all such alterations, permutations, and equivalents as fall within the true spirit and scope of the present invention.

What is claimed is:

1. A plurality of removable inserts to adapt a plurality of types of portable media players to a docking station, each of the plurality of types of portable media players having a varying size, each of the plurality of removable inserts comprising:

a basin having an inside surface to receive a bottom end of one of the plurality of types of portable media players, the basin further having an outside surface, the basin configured for insertion and extraction with the docking station, the basin further including:

one or more tabs located on a first side of the outside surface;

a catch located on a second side of the outside surface, the first side opposite the second side;

14

a lip extending from the inside surface of the basin, the lip configured to support and position the removable insert relative to the docking station when the removable insert is in a mounted position, wherein the lip is planar and is profiled to match a cavity in the docking station;

an opening for a recess, the opening for the recess located in the second side; and

an opening in a bottom for a connector of the docking station,

wherein the opening for the recess is located such that the opening for the recess is located in the center of the catch, wherein the recess comprises a pull for applying force about the one or more tabs on a first side of the basin, wherein the pull can be grasped by a finger.

2. The plurality of removable inserts of claim **1**, wherein the one or more tabs are configured for capture by corresponding slots in the docking station, and the catch is configured for capture by a latch in the docking station.

3. The plurality of removable inserts of claim **1**, wherein the one or more tabs includes a pair of tabs, the tabs spaced apart and located on opposite sides of a center of the removable insert, and wherein the catch is a rib that mates with a detent of the latch, the rib being located proximate the center of the removable insert, the tabs and rib working together to provide three point stability between the removable insert and the docking station.

4. A removable insert for coupling a portable media player to a docking station, the removable insert comprising:

a basin having an inside surface to receive a bottom end of the portable media player, the basin further having an outside surface, the basin configured for insertion and extraction with the docking station, the basin further including:

two tabs spaced apart and located on a first side of the outside surface and on opposite sides of a center of the removable insert;

a catch located on a second side of the outside surface, the first side opposite the second side;

a lip extending from the inside surface of the basin, the lip configured to support and position the removable insert relative to the docking station when the removable insert is in a mounted position, wherein the lip is planar;

an opening for a recess, the opening for the recess located in the second side such that the opening for the recess is located in the center of the catch; and

an opening in a bottom for a connector of the docking station,

wherein the recess comprises a pull for applying force about the one or more tabs on a first side of the basin, wherein the pull can be grasped by a finger.

5. The removable insert of claim **4**, wherein the one or more tabs are configured for capture by corresponding slots in the docking station, and the catch is configured for capture by a latch in the docking station.

6. The removable insert of claim **4**, wherein the catch is a rib that mates with a detent of the latch, the rib being located proximate the center of the removable insert, the tabs and rib working together to provide three point stability between the removable insert and the docking station.

* * * * *

EXHIBIT H



US00D588545S

(12) **United States Design Patent**
Andre et al.

(10) **Patent No.:** **US D588,545 S**
 (45) **Date of Patent:** **** Mar. 17, 2009**

(54) **CONNECTORS**

(75) Inventors: **Bartley K. Andre**, Menlo Park, CA (US); **Daniel J. Coster**, San Francisco, CA (US); **Daniele De Iuliis**, San Francisco, CA (US); **Richard P. Howarth**, San Francisco, CA (US); **Jonathan P. Ive**, San Francisco, CA (US); **Duncan Robert Kerr**, San Francisco, CA (US); **Shin Nishibori**, San Francisco, CA (US); **Matthew Dean Rohrbach**, San Francisco, CA (US); **Peter Russell-Clarke**, San Francisco, CA (US); **Douglas B. Satzger**, Menlo Park, CA (US); **Calvin Q. Seid**, Palo Alto, CA (US); **Christopher J. Stringer**, Portola Valley, CA (US); **Eugene Antony Whang**, San Francisco, CA (US); **Rico Zorkendorfer**, San Francisco, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(**) Term: **14 Years**

(21) Appl. No.: **29/270,882**

(22) Filed: **Jan. 5, 2007**

(51) **LOC (9) Cl.** **13-03**
 (52) **U.S. Cl.** **D13/154; D13/147**
 (58) **Field of Classification Search** **D13/133, D13/146, 147, 153, 154, 184, 199; 439/352, 439/358, 502, 607-610, 680**

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

1,642,064 A 9/1927 Engstrom
 (Continued)

FOREIGN PATENT DOCUMENTS

CN 3513646D 3/2006
 (Continued)

Primary Examiner—Daniel D Bui
Assistant Examiner—Thomas J Johannes
 (74) *Attorney, Agent, or Firm*—Sterne, Kessler, Goldstein & Fox P.L.L.C.

(57) **CLAIM**

We claim the ornamental design for connectors, substantially as shown and described.

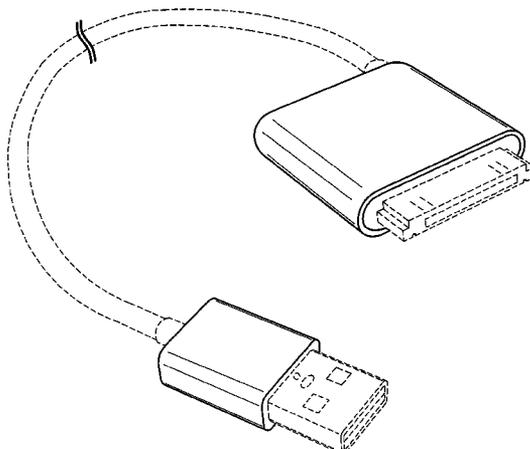
DESCRIPTION

FIG. 1 is a perspective view of connectors showing our new design, in a typical usage condition;
 FIG. 2 is a perspective view of the first of the two connectors, shown separately for clarity;
 FIG. 3 is a top view thereof;
 FIG. 4 is a bottom view thereof;
 FIG. 5 is a left view thereof;
 FIG. 6 is a right view thereof;
 FIG. 7 is a front view thereof;
 FIG. 8 is a rear side view thereof;
 FIG. 9 is a perspective view of the second of the two connectors, shown separately for clarity;
 FIG. 10 is a top view thereof;
 FIG. 11 is a bottom view thereof;
 FIG. 12 is a left view thereof;
 FIG. 13 is a right view thereof;
 FIG. 14 is a front view thereof; and,
 FIG. 15 is a rear side view thereof.

The unclaimed cable portion shown with the claimed connectors is shown broken away with a solid, wavy cut-off line in all of the figures to indicate indeterminate length.

The features shown in broken lines in the various Figures are for illustrating environmental structure and form no part of the claimed design.

1 Claim, 5 Drawing Sheets



US D588,545 S

Page 2

U.S. PATENT DOCUMENTS

1,781,958 A 1/1930 Schellenger
D102,203 S 12/1936 Comstock
3,351,892 A 11/1967 Feeser
3,727,098 A 4/1973 Crapuchettes
3,833,821 A 9/1974 Weaver
4,037,168 A 7/1977 Katoh et al.
4,084,218 A 4/1978 Kenney
D267,713 S 1/1988 Noda et al.
4,484,158 A 11/1984 Roberts et al.
4,632,486 A 12/1986 Hasircoglu
D290,458 S 6/1987 O'Leary
D298,313 S 11/1988 De Jong
D316,076 S 4/1991 Keating
5,019,767 A 5/1991 Shirai et al.
5,179,501 A * 1/1993 Ocken et al. 361/709
D337,305 S 7/1993 Ross et al.
D338,653 S 8/1993 Morgan et al.
D339,791 S 9/1993 Lucey et al.
D356,293 S 3/1995 Riley et al.
D367,257 S 2/1996 Buelow et al.
5,494,449 A 2/1996 Chioo
5,568,006 A 10/1996 Luff et al.
D380,196 S 6/1997 Aramaki
D390,828 S 2/1998 Aramaki
5,898,256 A 4/1999 Ide et al.
D418,479 S 1/2000 Chiou
D420,665 S 2/2000 Rosenberg et al.
D423,452 S 4/2000 Fujisawa et al.
D426,810 S 6/2000 Wei
6,122,526 A 9/2000 Parulski et al.
D432,523 S 10/2000 Grinkus et al.
D438,525 S 3/2001 Katayama
D441,735 S 5/2001 Katayama
D446,499 S 8/2001 Andre et al.
6,336,827 B1 1/2002 Akama et al.
D454,110 S 3/2002 Andre et al.
6,392,882 B1 5/2002 Chen et al.

D465,459 S 11/2002 Zhu et al.
D470,819 S * 2/2003 Sass et al. D13/133
D471,156 S * 3/2003 Sass et al. D13/133
D473,207 S 4/2003 Tanio
D479,200 S 9/2003 Liu et al.
6,758,685 B1 7/2004 Huang et al.
D500,737 S 1/2005 Shao
D502,146 S 2/2005 Lai et al.
D503,409 S 3/2005 Wong et al.
D506,476 S 6/2005 Andre et al.
D515,545 S 2/2006 Griffin
D516,579 S 3/2006 Nakamura
7,013,164 B2 3/2006 Lin
D521,936 S 5/2006 Novotney et al.
D527,723 S 9/2006 Andre et al.
D534,184 S 12/2006 Leija et al.
D534,545 S 1/2007 Leija et al.
D537,074 S 2/2007 Andre et al.
D537,780 S * 3/2007 Suckle D13/133
D539,814 S 4/2007 Andre et al.
D551,250 S 9/2007 Andre et al.
D559,183 S * 1/2008 Victor D13/133
D576,102 S * 9/2008 Zenri et al. D13/147

FOREIGN PATENT DOCUMENTS

CN 3543434D 7/2006
CN 3549064D 8/2006
GB 1443429 7/1921
JP 1049198 9/1997
JP 1075497 3/1998
JP D1121650 9/2001
JP 1156347 10/2001
JP 1163316 2/2002
JP 1228505 9/2002
JP 1185626 11/2002
JP 1185627 11/2002
KR 193346 12/1995

* cited by examiner

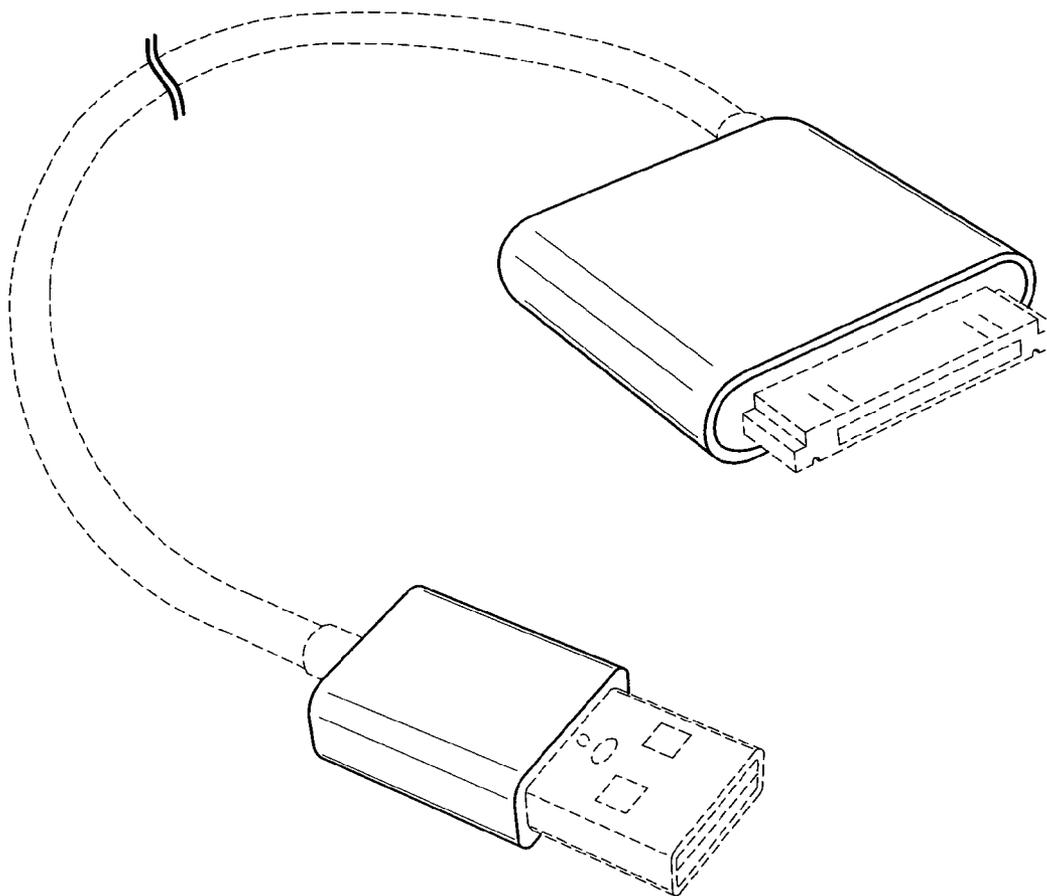


FIG. 1

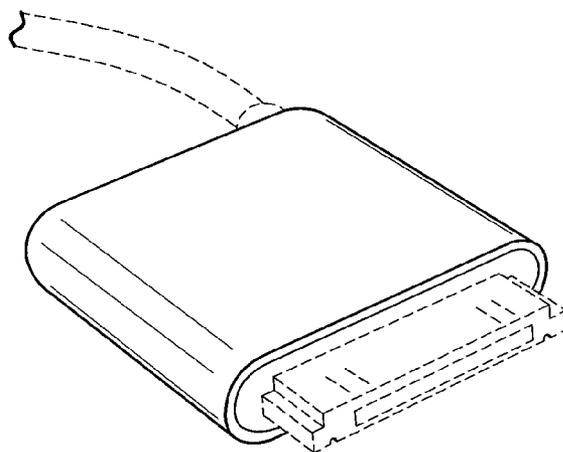


FIG. 2

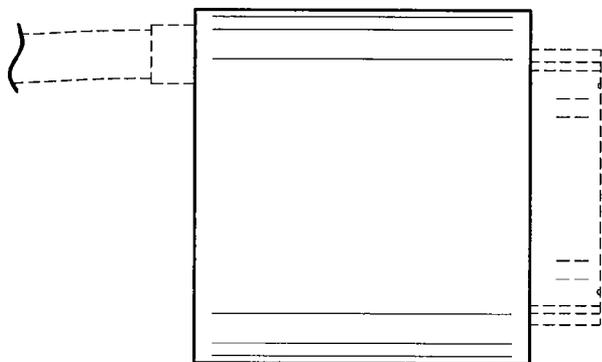


FIG. 3

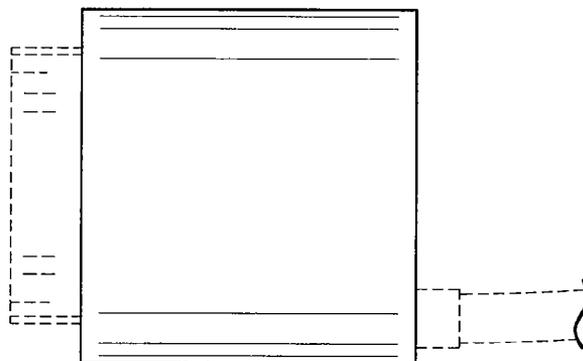


FIG. 4

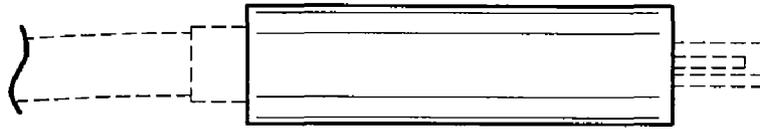


FIG. 5

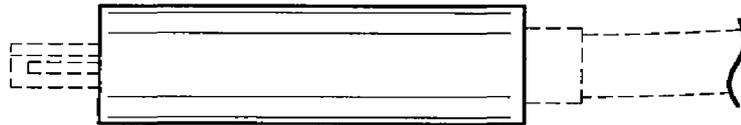


FIG. 6

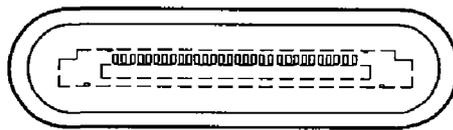


FIG. 7

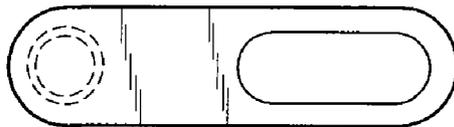


FIG. 8

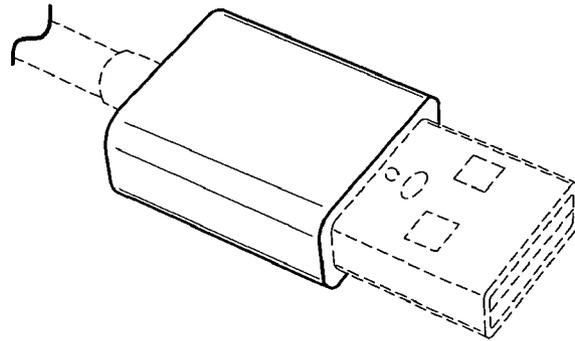


FIG. 9

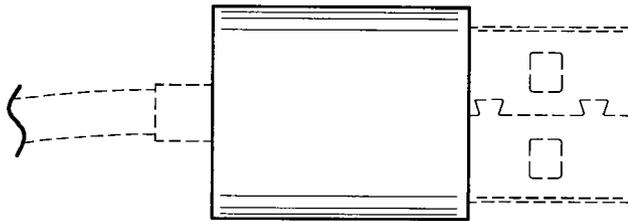


FIG. 10

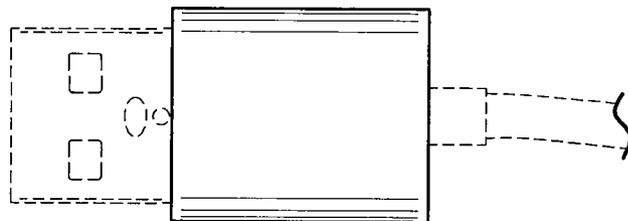


FIG. 11

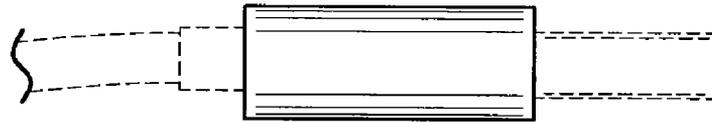


FIG. 12

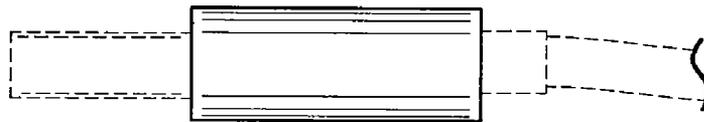


FIG. 13



FIG. 14

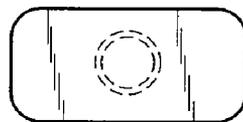


FIG. 15



US00D588545S

(12) **United States Design Patent**
Andre et al.

(10) **Patent No.:** **US D588,545 S**
(45) **Date of Patent:** **** Mar. 17, 2009**

(54) **CONNECTORS**

(75) Inventors: **Bartley K. Andre**, Menlo Park, CA (US); **Daniel J. Coster**, San Francisco, CA (US); **Daniele De Iuliis**, San Francisco, CA (US); **Richard P. Howarth**, San Francisco, CA (US); **Jonathan P. Ive**, San Francisco, CA (US); **Duncan Robert Kerr**, San Francisco, CA (US); **Shin Nishibori**, San Francisco, CA (US); **Matthew Dean Rohrbach**, San Francisco, CA (US); **Peter Russell-Clarke**, San Francisco, CA (US); **Douglas B. Satzger**, Menlo Park, CA (US); **Calvin Q. Seid**, Palo Alto, CA (US); **Christopher J. Stringer**, Portola Valley, CA (US); **Eugene Antony Whang**, San Francisco, CA (US); **Rico Zorkendorfer**, San Francisco, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(**) Term: **14 Years**

(21) Appl. No.: **29/270,882**

(22) Filed: **Jan. 5, 2007**

(51) **LOC (9) Cl.** **13-03**

(52) **U.S. Cl.** **D13/154; D13/147**

(58) **Field of Classification Search** D13/133,
D13/146, 147, 153, 154, 184, 199; 439/352,
439/358, 502, 607-610, 680

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

1,642,064 A 9/1927 Engstrom
(Continued)

FOREIGN PATENT DOCUMENTS

CN 3513646D 3/2006
(Continued)

Primary Examiner—Daniel D Bui
Assistant Examiner—Thomas J Johannes
(74) *Attorney, Agent, or Firm*—Sterne, Kessler, Goldstein & Fox P.L.L.C.

(57) **CLAIM**

We claim the ornamental design for connectors, substantially as shown and described.

DESCRIPTION

FIG. 1 is a perspective view of connectors showing our new design, in a typical usage condition;

FIG. 2 is a perspective view of the first of the two connectors, shown separately for clarity;

FIG. 3 is a top view thereof;

FIG. 4 is a bottom view thereof;

FIG. 5 is a left view thereof;

FIG. 6 is a right view thereof;

FIG. 7 is a front view thereof;

FIG. 8 is a rear side view thereof;

FIG. 9 is a perspective view of the second of the two connectors, shown separately for clarity;

FIG. 10 is a top view thereof;

FIG. 11 is a bottom view thereof;

FIG. 12 is a left view thereof;

FIG. 13 is a right view thereof;

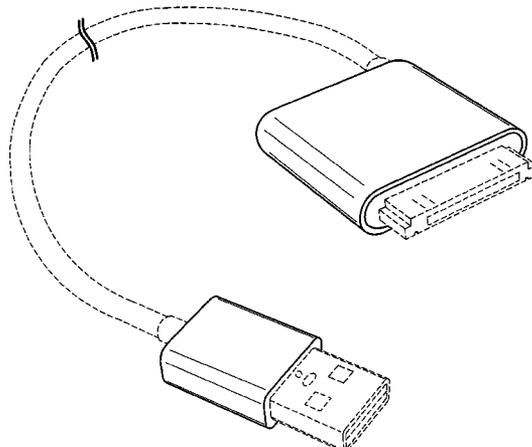
FIG. 14 is a front view thereof; and,

FIG. 15 is a rear side view thereof.

The unclaimed cable portion shown with the claimed connectors is shown broken away with a solid, wavy cut-off line in all of the figures to indicate indeterminate length.

The features shown in broken lines in the various Figures are for illustrating environmental structure and form no part of the claimed design.

1 Claim, 5 Drawing Sheets



US D588,545 S

Page 2

U.S. PATENT DOCUMENTS

1,781,958 A 1/1930 Schellenger
D102,203 S 12/1936 Comstock
3,351,892 A 11/1967 Feeser
3,727,098 A 4/1973 Crapuchettes
3,833,821 A 9/1974 Weaver
4,037,168 A 7/1977 Katoh et al.
4,084,218 A 4/1978 Kenney
D267,713 S 1/1988 Noda et al.
4,484,158 A 11/1984 Roberts et al.
4,632,486 A 12/1986 Hasircoglu
D290,458 S 6/1987 O'Leary
D298,313 S 11/1988 De Jong
D316,076 S 4/1991 Keating
5,019,767 A 5/1991 Shirai et al.
5,179,501 A * 1/1993 Ocken et al. 361/709
D337,305 S 7/1993 Ross et al.
D338,653 S 8/1993 Morgan et al.
D339,791 S 9/1993 Lucey et al.
D356,293 S 3/1995 Riley et al.
D367,257 S 2/1996 Buelow et al.
5,494,449 A 2/1996 Chioo
5,568,006 A 10/1996 Luff et al.
D380,196 S 6/1997 Aramaki
D390,828 S 2/1998 Aramaki
5,898,256 A 4/1999 Ide et al.
D418,479 S 1/2000 Chiou
D420,665 S 2/2000 Rosenberg et al.
D423,452 S 4/2000 Fujisawa et al.
D426,810 S 6/2000 Wei
6,122,526 A 9/2000 Parulski et al.
D432,523 S 10/2000 Grinkus et al.
D438,525 S 3/2001 Katayama
D441,735 S 5/2001 Katayama
D446,499 S 8/2001 Andre et al.
6,336,827 B1 1/2002 Akama et al.
D454,110 S 3/2002 Andre et al.
6,392,882 B1 5/2002 Chen et al.

D465,459 S 11/2002 Zhu et al.
D470,819 S * 2/2003 Sass et al. D13/133
D471,156 S * 3/2003 Sass et al. D13/133
D473,207 S 4/2003 Tanio
D479,200 S 9/2003 Liu et al.
6,758,685 B1 7/2004 Huang et al.
D500,737 S 1/2005 Shao
D502,146 S 2/2005 Lai et al.
D503,409 S 3/2005 Wong et al.
D506,476 S 6/2005 Andre et al.
D515,545 S 2/2006 Griffin
D516,579 S 3/2006 Nakamura
7,013,164 B2 3/2006 Lin
D521,936 S 5/2006 Novotney et al.
D527,723 S 9/2006 Andre et al.
D534,184 S 12/2006 Leija et al.
D534,545 S 1/2007 Leija et al.
D537,074 S 2/2007 Andre et al.
D537,780 S * 3/2007 Suckle D13/133
D539,814 S 4/2007 Andre et al.
D551,250 S 9/2007 Andre et al.
D559,183 S * 1/2008 Victor D13/133
D576,102 S * 9/2008 Zenri et al. D13/147

FOREIGN PATENT DOCUMENTS

CN 3543434D 7/2006
CN 3549064D 8/2006
GB 1443429 7/1921
JP 1049198 9/1997
JP 1075497 3/1998
JP D1121650 9/2001
JP 1156347 10/2001
JP 1163316 2/2002
JP 1228505 9/2002
JP 1185626 11/2002
JP 1185627 11/2002
KR 193346 12/1995

* cited by examiner

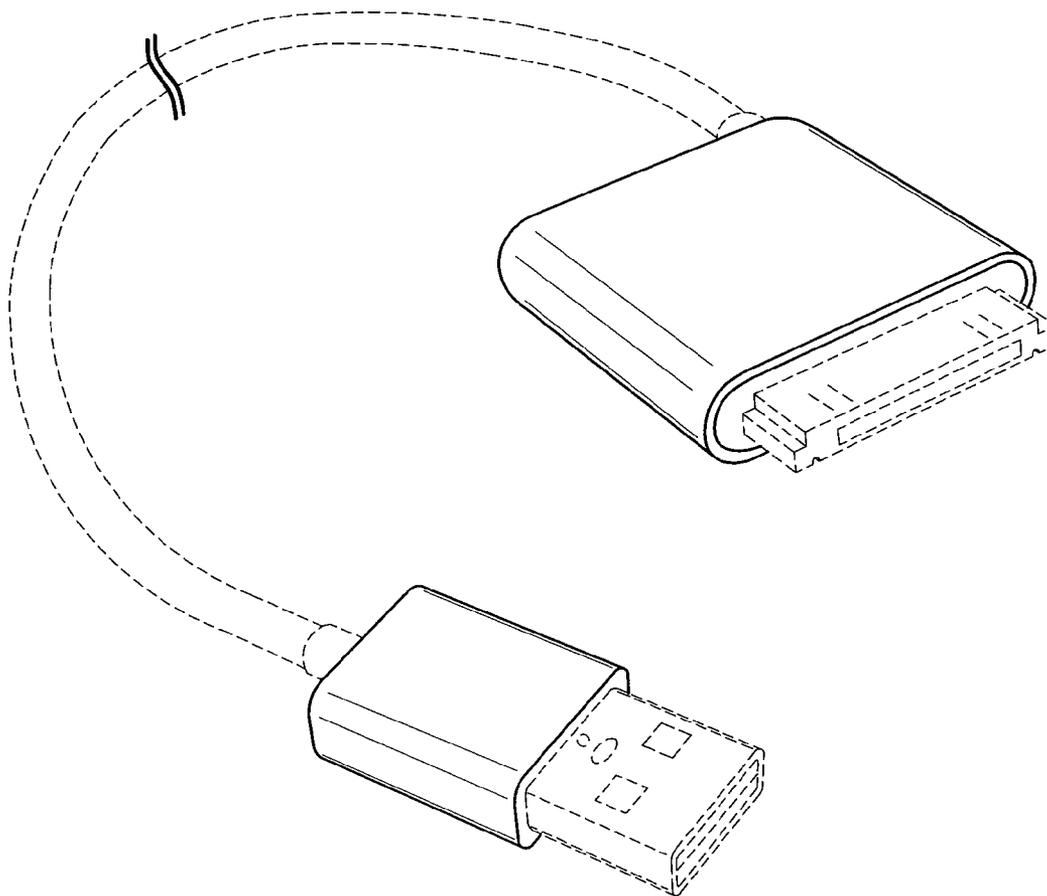


FIG. 1

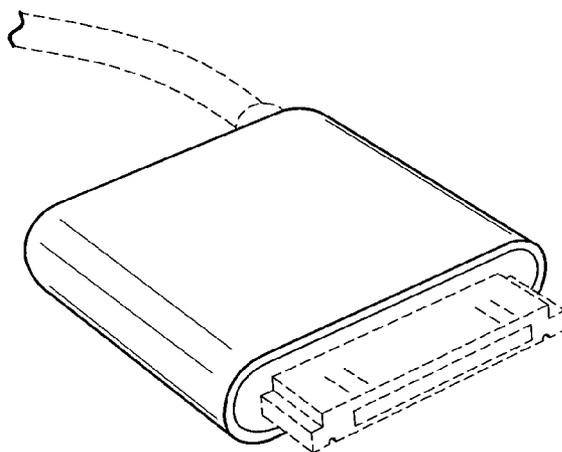


FIG. 2

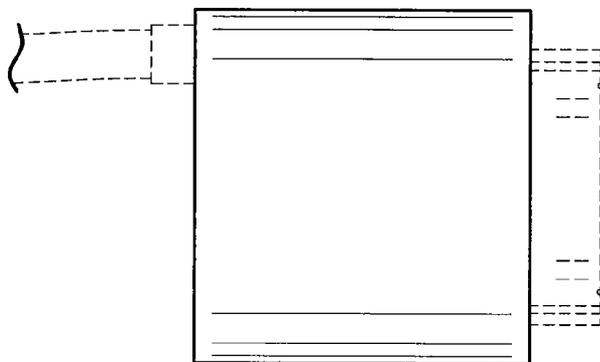


FIG. 3

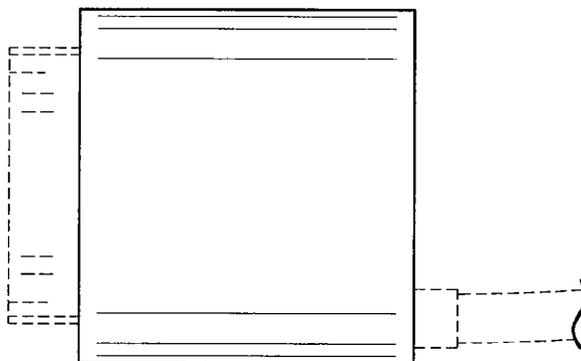


FIG. 4

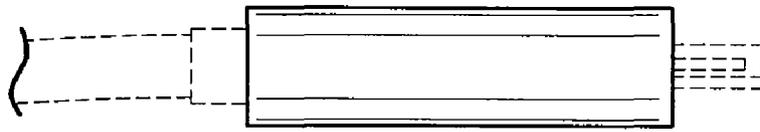


FIG. 5

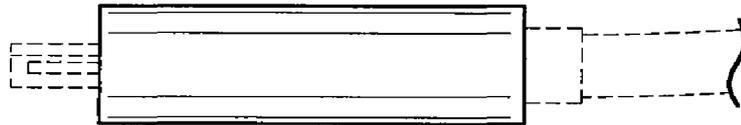


FIG. 6

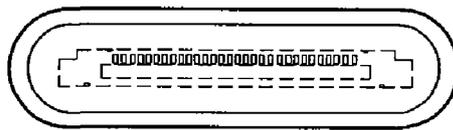


FIG. 7

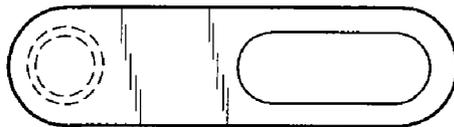


FIG. 8

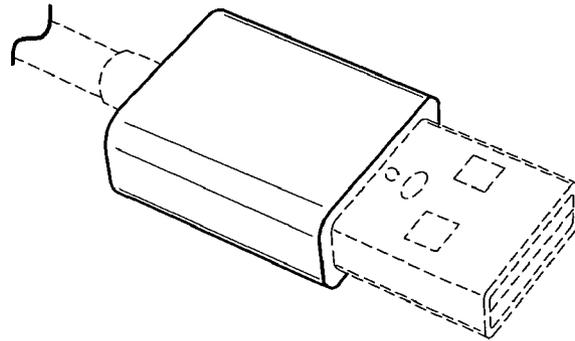


FIG. 9

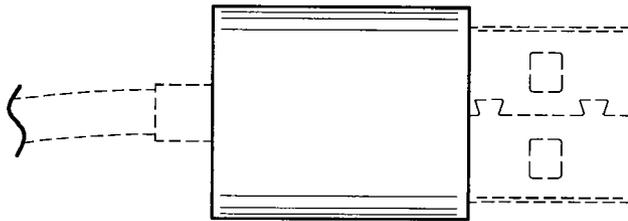


FIG. 10

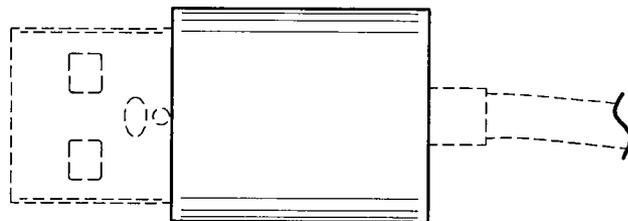


FIG. 11

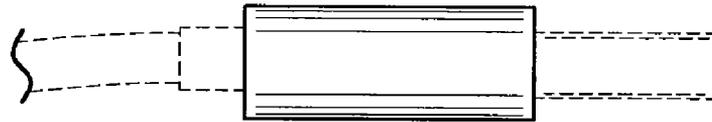


FIG. 12

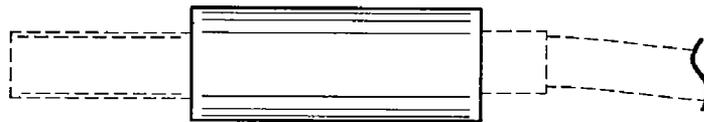


FIG. 13



FIG. 14

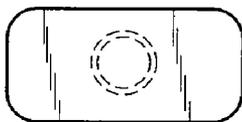


FIG. 15

EXHIBIT I



US00D596621S

(12) **United States Design Patent** (10) **Patent No.:** **US D596,621 S**
Andre et al. (45) **Date of Patent:** **** Jul. 21, 2009**

(54) **DOCKING STATION**

(75) Inventors: **Bartley K. Andre**, Menlo Park, CA (US); **Daniel J. Coster**, San Francisco, CA (US); **Daniele De Iuliis**, San Francisco, CA (US); **Richard P. Howarth**, San Francisco, CA (US); **Jonathan P. Ive**, San Francisco, CA (US); **Duncan Robert Kerr**, San Francisco, CA (US); **Shin Nishibori**, San Francisco, CA (US); **Matthew Dean Rohrbach**, San Francisco, CA (US); **Douglas B. Satzger**, Menlo Park, CA (US); **Calvin Q. Seid**, Palo Alto, CA (US); **Vincent Keane Seid**, legal representative, Los Gatos, CA (US); **Christopher J. Stringer**, Portola Valley, CA (US); **Eugene Antony Whang**, San Francisco, CA (US); **Rico Zorkendorfer**, San Francisco, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(**) Term: **14 Years**

(21) Appl. No.: **29/326,229**

(22) Filed: **Oct. 14, 2008**

Related U.S. Application Data

(63) Continuation of application No. 29/282,424, filed on Jul. 19, 2007, now Pat. No. Des. 578,512, which is a continuation of application No. 29/237,233, filed on Aug. 24, 2005, now Pat. No. Des. 558,739.

(51) **LOC (9) Cl.** **14-03**

(52) **U.S. Cl.** **D14/217**

(58) **Field of Classification Search** D14/142, D14/149, 155, 157, 188, 191-193, 230.1-203.8, D14/217, 251, 253, 432-435.1, 447, 451, D14/496, 474, 480.1-480.7; D13/107, 108, D13/118; D3/218; 248/221.11, 309.1; 320/107, 320/110, 113-115; 361/600, 679, 683, 686; 439/529, 534; 455/556.1-556.2, 557

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

D358,821 S	5/1995	Hellier et al.	
5,602,458 A	2/1997	Dowe	
D385,530 S	10/1997	Hames et al.	
D390,824 S	2/1998	Foster et al.	
5,952,814 A	9/1999	Van Lerberghe	
6,042,414 A	3/2000	Kunert	
D442,939 S	5/2001	Goldenburg	
D463,361 S	9/2002	Ruohonen	
D477,823 S	7/2003	Lavello, Sr.	
D495,336 S	8/2004	Andre et al.	
7,075,579 B2	7/2006	Whitby et al.	
D530,717 S	10/2006	Mori et al.	
D532,005 S *	11/2006	Leith	D14/253
D535,614 S	1/2007	Majanen et al.	
D540,327 S	4/2007	Wong et al.	
D544,899 S	6/2007	Li et al.	
D551,214 S	9/2007	Hussaini et al.	
D551,658 S	9/2007	Hussaini et al.	
D553,106 S	10/2007	Griffin	
D558,208 S	12/2007	Ikeda et al.	
D558,738 S	1/2008	Andre et al.	
D558,739 S	1/2008	Andre et al.	
D575,275 S	8/2008	Andre et al.	
D578,110 S	10/2008	Andre et al.	
D578,512 S	10/2008	Andre et al.	
D580,855 S *	11/2008	Haspil et al.	D13/108
D580,856 S *	11/2008	Haspil et al.	D13/108
D582,887 S *	12/2008	Poandl	D14/172
2006/0250764 A1	11/2006	Howarth et al.	
2007/0047198 A1	3/2007	Crooijmans et al.	

OTHER PUBLICATIONS

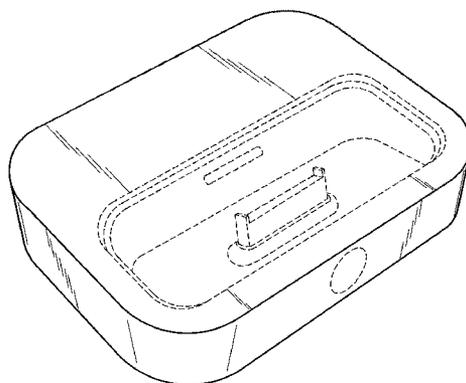
U.S. Appl. No. 29/237,687 entitled "Docking Station" filed Sep. 2, 2005.
 U.S. Appl. No. 29/296,895 entitled "Docking Station" filed Oct. 30, 2007.
 U.S. Appl. No. 29/319,154 entitled "Stand" filed Jun. 4, 2008.
 U.S. Appl. No. 29/319,238 entitled "Stand" filed Jun. 5, 2008.

* cited by examiner

Primary Examiner—Stella M Reid

Assistant Examiner—Keli L Hill

(74) *Attorney, Agent, or Firm*—Sterne Kessler Goldstein & Fox P.L.L.C.



US D596,621 S

Page 2

(57)

CLAIM

The ornamental design for a docking station, as shown and described.

DESCRIPTION

This application is related to U.S. application Ser. No. 29/296,825, filed Oct. 29, 2007, which is now U.S. Pat. No. D575,275 and to U.S. application Ser. No. 29/237,084, filed Aug. 24, 2005, which is now U.S. Pat. No. D558,738. The disclosures of application Ser. Nos. 29/282,424, 29/237,233, and 29/296,825 are herein incorporated in their entireties by reference thereto.

FIG. 1 is a front perspective view of a docking station showing our new design;

FIG. 2 is a rear perspective view thereof;

FIG. 3 is a top view thereof;

FIG. 4 is a bottom view thereof;

FIG. 5 is a front view thereof;

FIG. 6 is a rear view thereof;

FIG. 7 is a left side view thereof;

FIG. 8 is a right side view thereof;

FIG. 9 is an exemplary perspective view of how a docking system can incorporate a dock insert with the docking station; and,

FIGS. 10–13 are exemplary top views of a docking system that incorporates different dock inserts into the docking station.

Various examples of dock insert embodiments are shown and described, for example, in (i) U.S. Design patent application Ser. No. 29/237,097, filed Aug. 24, 2005, now U.S. Pat. No. D552,085; (ii) U.S. Design patent application Ser. No. 29/237,111, filed Aug. 24, 2005, now U.S. Pat. No. D551,212; and (iii) U.S. Design patent application Ser. No. 29/237,112, filed Aug. 24, 2005, now U.S. Pat. No. D551,213.

The broken lines in the Figures are for illustrating environmental structure only and form no part of the claimed design.

1 Claim, 6 Drawing Sheets

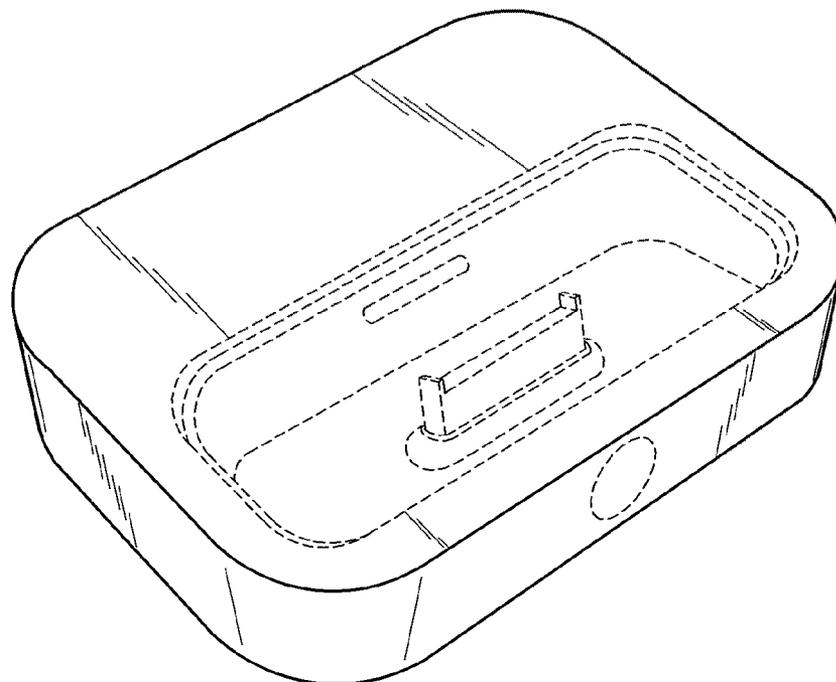


FIG. 1

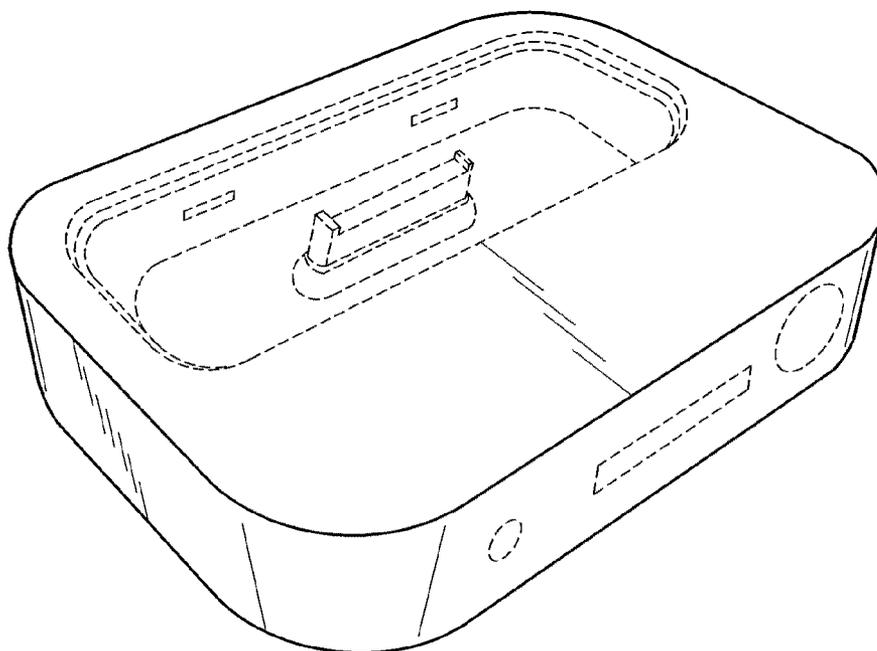


FIG. 2

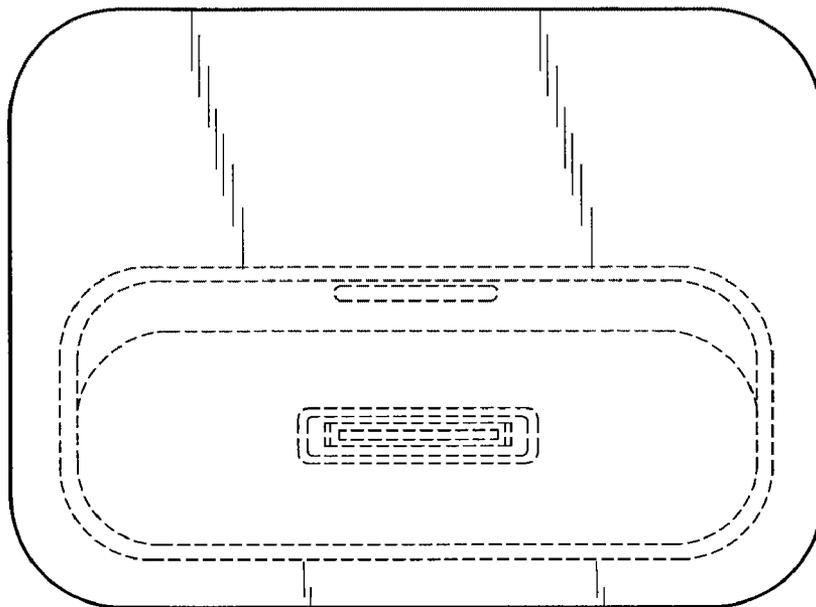


FIG. 3

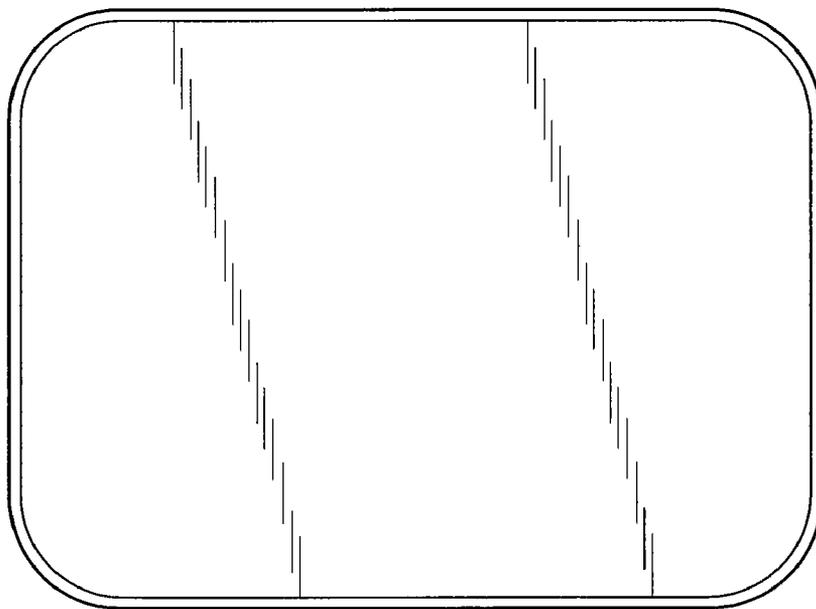


FIG. 4

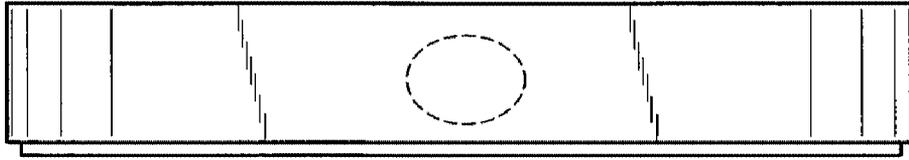


FIG. 5

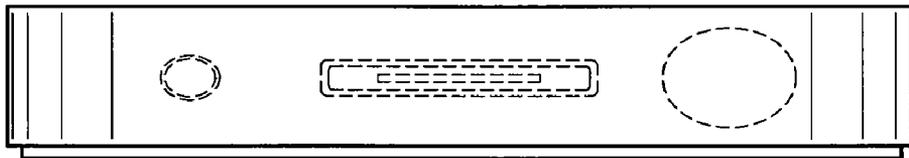


FIG. 6

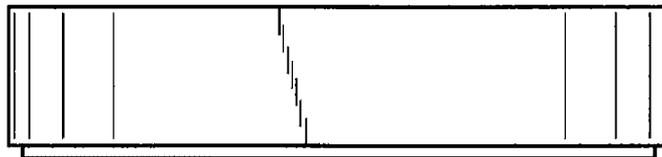


FIG. 7

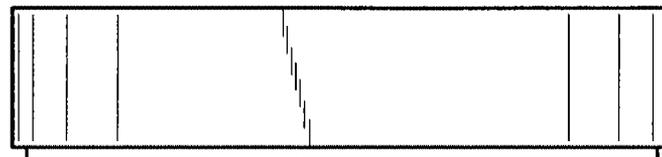


FIG. 8

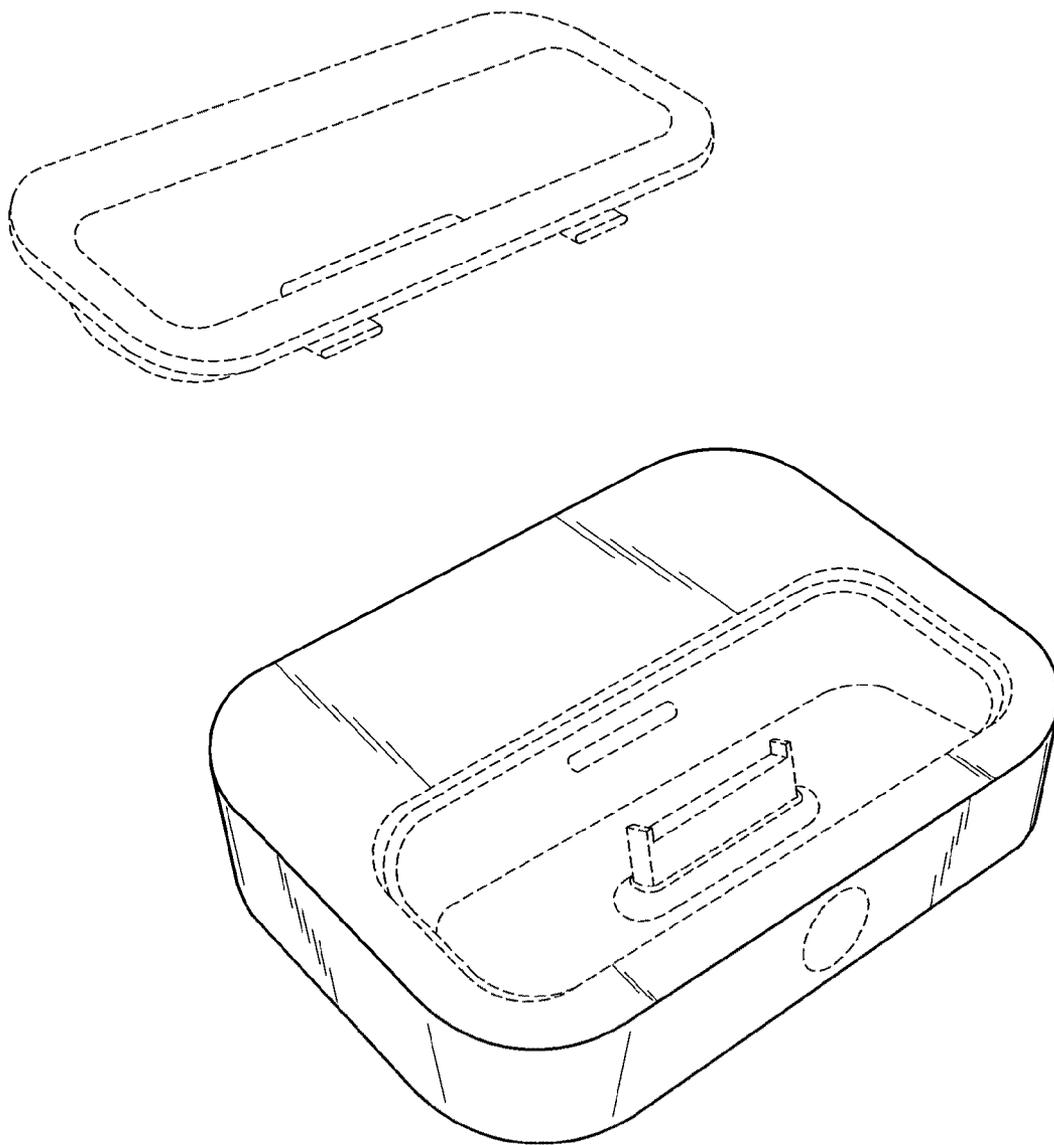


FIG. 9

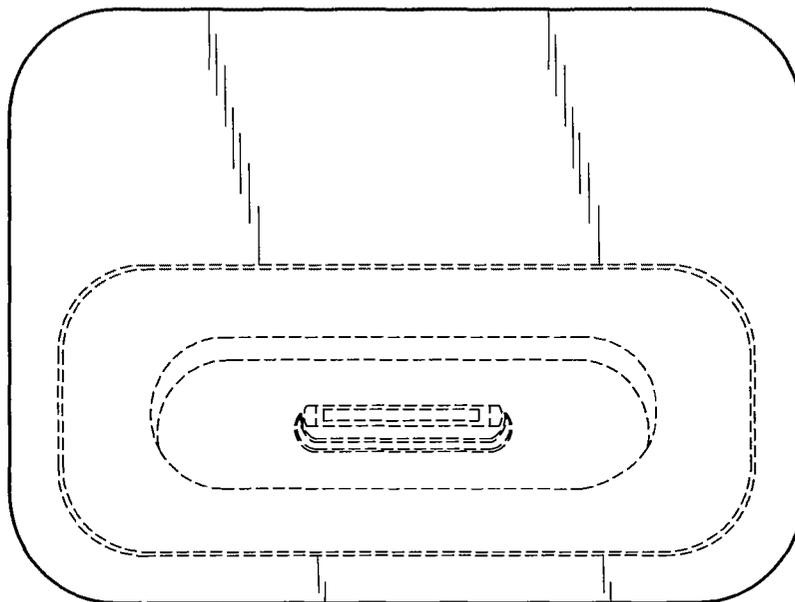


FIG. 10

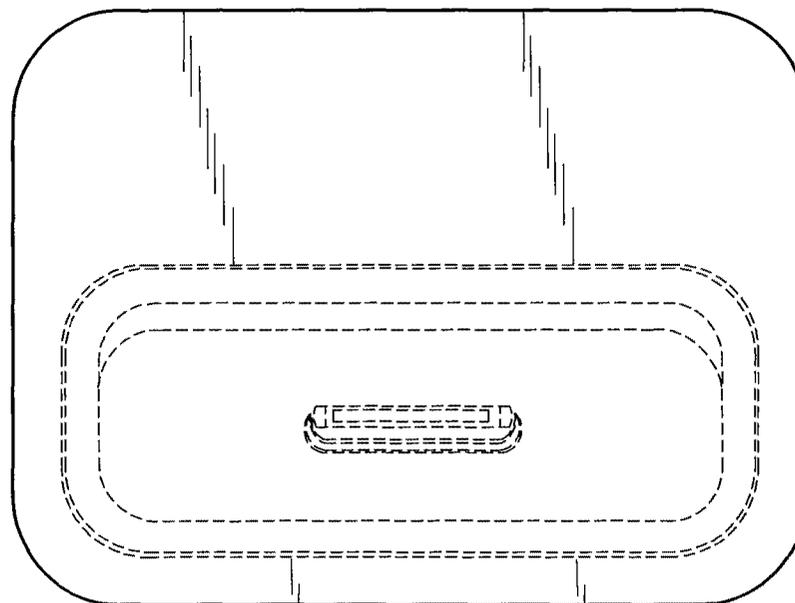


FIG. 11

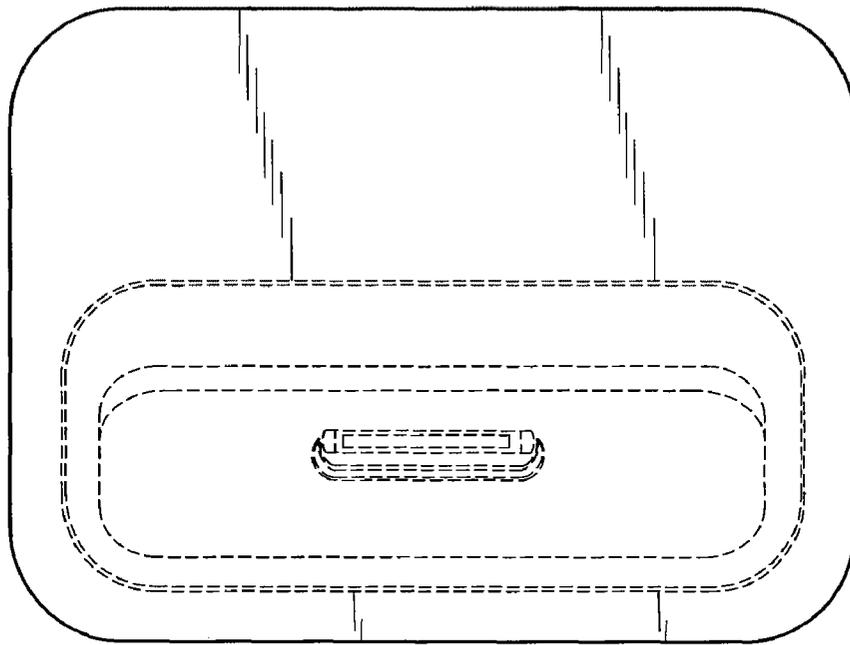


FIG. 12

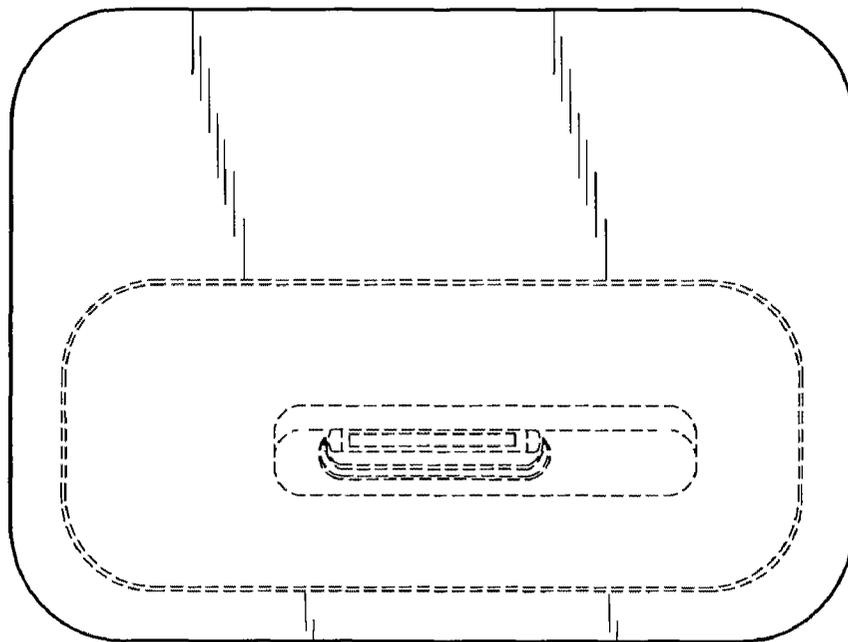


FIG. 13

EXHIBIT J

(12) **United States Design Patent** (10) **Patent No.:** **US D578,110 S**
Andre et al. (45) **Date of Patent:** **** Oct. 7, 2008**

(54) **DOCKING STATION**

(75) Inventors: **Bartley K. Andre**, Menlo Park, CA (US); **Daniel J. Coster**, San Francisco, CA (US); **Daniele De Iuliis**, San Francisco, CA (US); **Richard P. Howarth**, San Francisco, CA (US); **Jonathan P. Ive**, San Francisco, CA (US); **Duncan Robert Kerr**, San Francisco, CA (US); **Shin Nishibori**, San Francisco, CA (US); **Matthew Dean Rohrbach**, San Francisco, CA (US); **Douglas B. Satzger**, Menlo Park, CA (US); **Calvin Q. Seid**, deceased, late of Palo Alto CA (US); by **Vincent Keane Seid**, legal representative, Los Gatos, CA (US); **Christopher J. Stringer**, Woodside, CA (US); **Eugene Antony Whang**, San Francisco, CA (US); **Rico Zorkendorfer**, San Francisco, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(**) Term: **14 Years**

(21) Appl. No.: **29/296,503**

(22) Filed: **Oct. 22, 2007**

Related U.S. Application Data

(63) Continuation of application No. 29/237,084, filed on Aug. 24, 2005, now Pat. No. Des. 558,738.

(51) **LOC (8) Cl.** **14-02**

(52) **U.S. Cl.** **D14/217**

(58) **Field of Classification Search** D14/142, D14/149, 217, 251, 432, 434, 435, 447, 496; D13/107, 108, 118; 248/221.11, 309.1; 320/107, 320/110, 113-115; 361/600, 679, 683, 686; 439/529, 534; 455/556.1, 556.2, 557; D21/524; D23/270, 284, 231.1

See application file for complete search history.

(56) **References Cited****U.S. PATENT DOCUMENTS**

D328,277 S 7/1992 Leman et al.

(Continued)

OTHER PUBLICATIONS

U.S. Appl. No. 29/237,233 entitled "Docking Station" filed Aug. 24, 2005.

(Continued)

Primary Examiner—Stella M. Reid

Assistant Examiner—Keli L Acker

(74) *Attorney, Agent, or Firm*—Sterne, Kessler, Goldstein & Fox P.L.L.C.

(57) **CLAIM**

We claim the ornamental design for a docking station, substantially as shown and described.

DESCRIPTION

This application is related to: U.S. Design patent application No. 29/237,233, and entitled "Docking Station" which is hereby incorporated herein by reference.

Various examples of dock insert embodiments are shown and described in in (i) U.S. Design patent application No. 29/237,097, entitled "Dock Insert"; (ii) U.S. Design patent application No. 29/237,111 entitled "Dock Insert"; and (iii) U.S. Design patent application No. 29/237,112 entitled "Dock Insert", each of which is hereby incorporated herein by reference.

FIG. 1 is a front perspective view of a docking station embodying our new design.

FIG. 2 is a rear perspective view of the docking station.

FIG. 3 is a top view for the docking station.

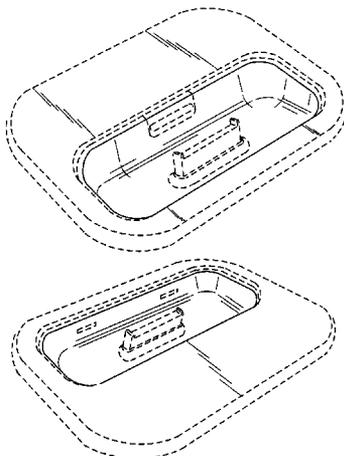
FIG. 4 is an exemplary perspective view of how a docking system can incorporate the docking station with a dock insert.

FIGS. 5-8 are exemplary top views of a docking system that incorporates different dock inserts into the docking station; and,

FIG. 9 is an exemplary perspective view of the docking station integrated into an electronic device such as a portable music player (e.g., boom box).

The broken lines immediately adjacent the shaded areas represent the bounds of the claimed design while all other broken lines are directed to environment and are for illustrative purposes only; the broken lines form no part of the claimed design.

1 Claim, 6 Drawing Sheets



US D578,110 S

Page 2

U.S. PATENT DOCUMENTS

D336,631 S	6/1993	Ivester	D526,321 S	8/2006	Hayes et al.
D350,330 S	9/1994	Peterson	D530,717 S	10/2006	Mori et al.
D374,858 S	10/1996	Chu	D532,005 S	11/2006	Leith
5,600,225 A	2/1997	Goto	7,145,603 B2	12/2006	Whitby et al.
5,627,727 A	5/1997	Aguilera et al.	D535,614 S	1/2007	Majanen et al.
D385,530 S	10/1997	Hames et al.	D537,925 S *	3/2007	Mahon D23/293.1
D388,764 S	1/1998	Bartling et al.	D551,212 S	9/2007	Andre et al.
D397,996 S	9/1998	Smith	D551,213 S	9/2007	Andre et al.
D409,987 S	5/1999	Okumura	D551,214 S *	9/2007	Hussaini et al. D14/217
D411,166 S	6/1999	Uemura et al.	D551,658 S *	9/2007	Hussaini et al. D14/217
5,952,814 A	9/1999	Van	D552,085 S	10/2007	Andre et al.
D416,536 S	11/1999	Ross et al.	D553,106 S *	10/2007	Griffin D14/434
D419,160 S	1/2000	Davidson et al.	D558,207 S *	12/2007	Ikeda et al. D14/434
6,042,414 A	3/2000	Kunert	D558,208 S *	12/2007	Ikeda et al. D14/434
D422,556 S	4/2000	Okura et al.	D558,209 S *	12/2007	Ikeda et al. D14/434
D439,908 S	4/2001	Gozani	2002/0103008 A1	8/2002	Rahn et al.
D440,542 S	4/2001	Hawkins et al.	2003/0198015 A1	10/2003	Vogt
D442,939 S	5/2001	Goldenburg	2006/0250764 A1	11/2006	Howarth et al.
D444,121 S	6/2001	Chun et al.	2007/0047198 A1	3/2007	Crooijmans et al.
D444,124 S	6/2001	Casey et al.			
6,307,348 B1	10/2001	Green			
D450,707 S	11/2001	Francavilla et al.			
D454,332 S	3/2002	Yokoyama			
D459,299 S	6/2002	Hughes et al.			
D461,476 S	8/2002	Evers et al.			
D463,361 S	9/2002	Ruohonen			
D466,122 S	11/2002	Moody			
D468,305 S	1/2003	Ma et al.			
6,524,240 B1	2/2003	Thede			
D472,900 S	4/2003	Matsumoto			
D477,823 S	7/2003	Lavello			
6,711,005 B2	3/2004	Martin			
D495,336 S	8/2004	Andre et al.			
6,798,647 B2	9/2004	Dickie			
6,813,528 B1	11/2004	Yang			
6,898,080 B2	5/2005	Yin et al.			
D510,091 S	9/2005	Mori et al.			
D525,616 S	7/2006	Andre et al.			
7,075,579 B2	7/2006	Whitby et al.			

OTHER PUBLICATIONS

“iPod Dock Cradle”, downloaded Apr. 11, 2003, www.bookendzdocks.com/bookendz/dock_cradel.htm.

“BookEndz Home Page”, downloaded Apr. 11, 2003, www.bookendzdocks.com/bookendz/index.htm.

“Atech Flash Technology Iduo Ipod Dock and card Reader”, www.the-gadgeteer.com, downloaded Nov. 21, 2005.

“Adaptec xHub for Notebooks”, http://www.everythingusb.com/hardware/index/Adaptec_xHub_for_Notebooks.htm, downloaded Nov. 22, 2005.

“Belkin’s Top Loading USB Hub for iPod shuffle”, http://www.everythingusb.com/belkin_hispeed_usb_2.0_4-port_hub.html, downloaded Nov. 22, 2005.

“Belkin Hi-Speed USB 2.07_Port Hub”, http://www.everythingusb.com/hardware/index/Belkin_Hi-Speed_USB_2.0_7-Port_Hub..., downloaded Nov. 22, 2005.

U.S. Appl. No. 29/319,055 entitled “Dock Insert” Filed Jun. 3, 2008.

U.S. Appl. No. 29/319,155 entitled “Dock Insert” filed Jun. 4, 2008.

* cited by examiner

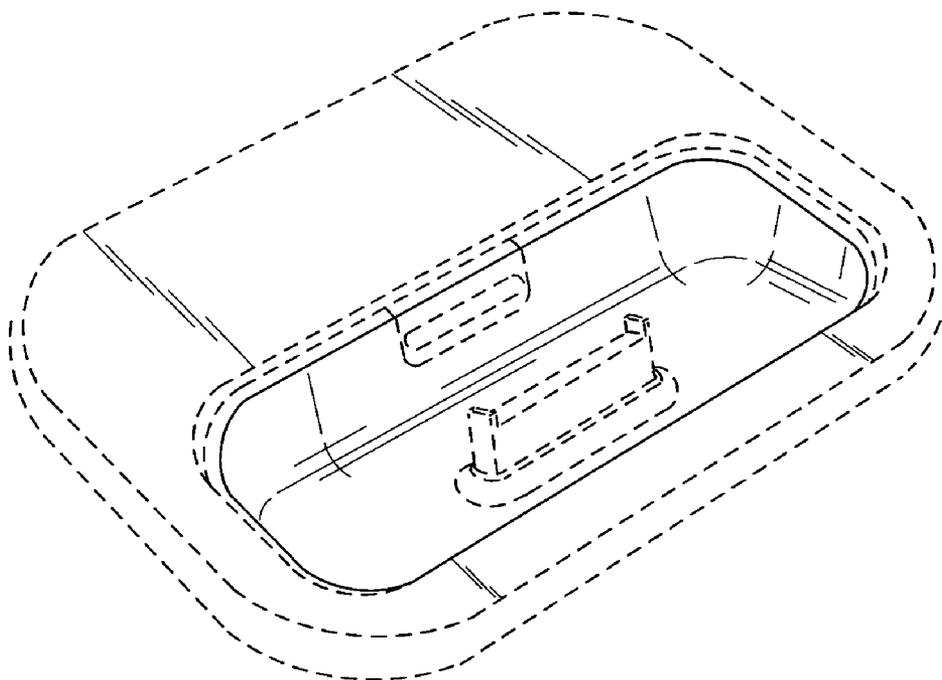


FIG. 1

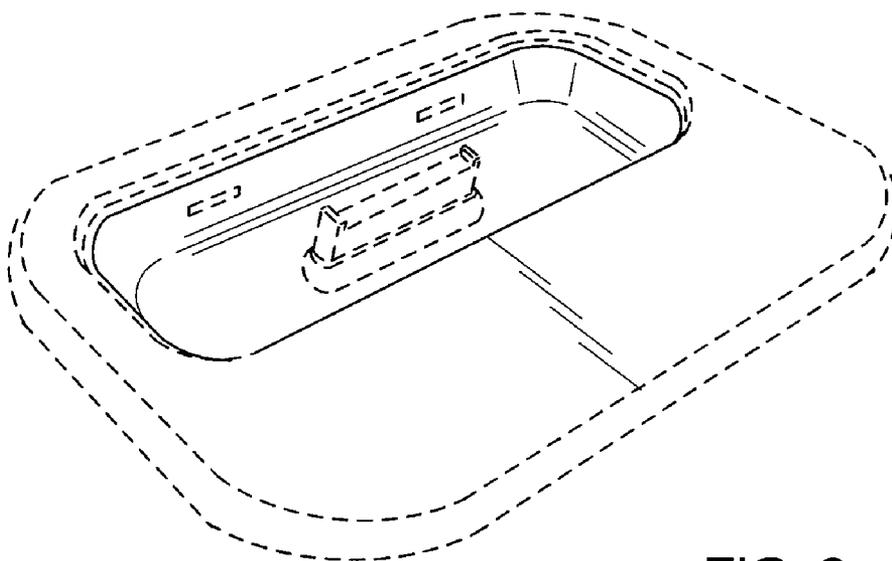


FIG. 2

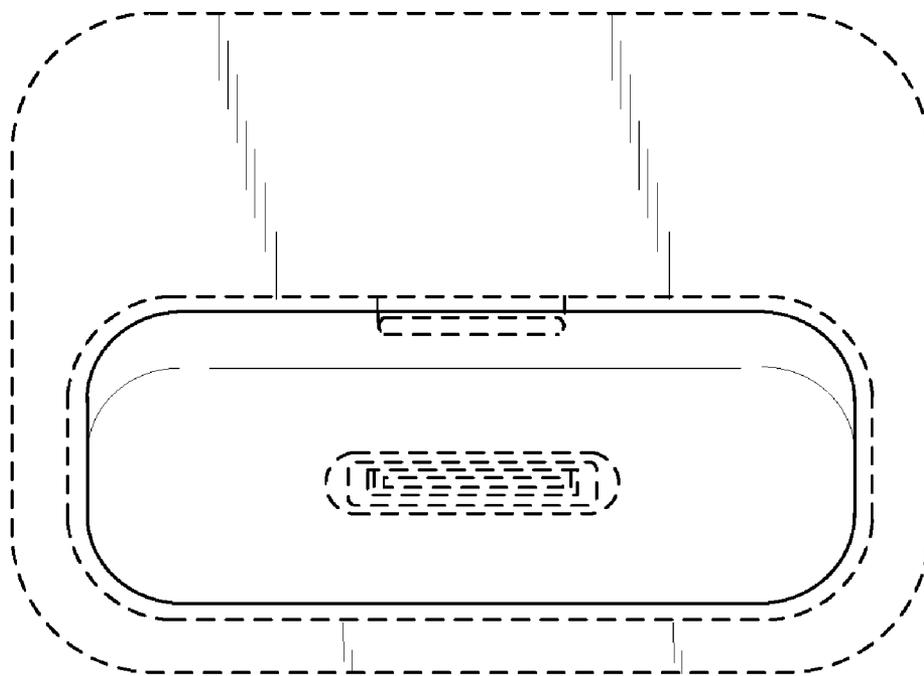


FIG. 3

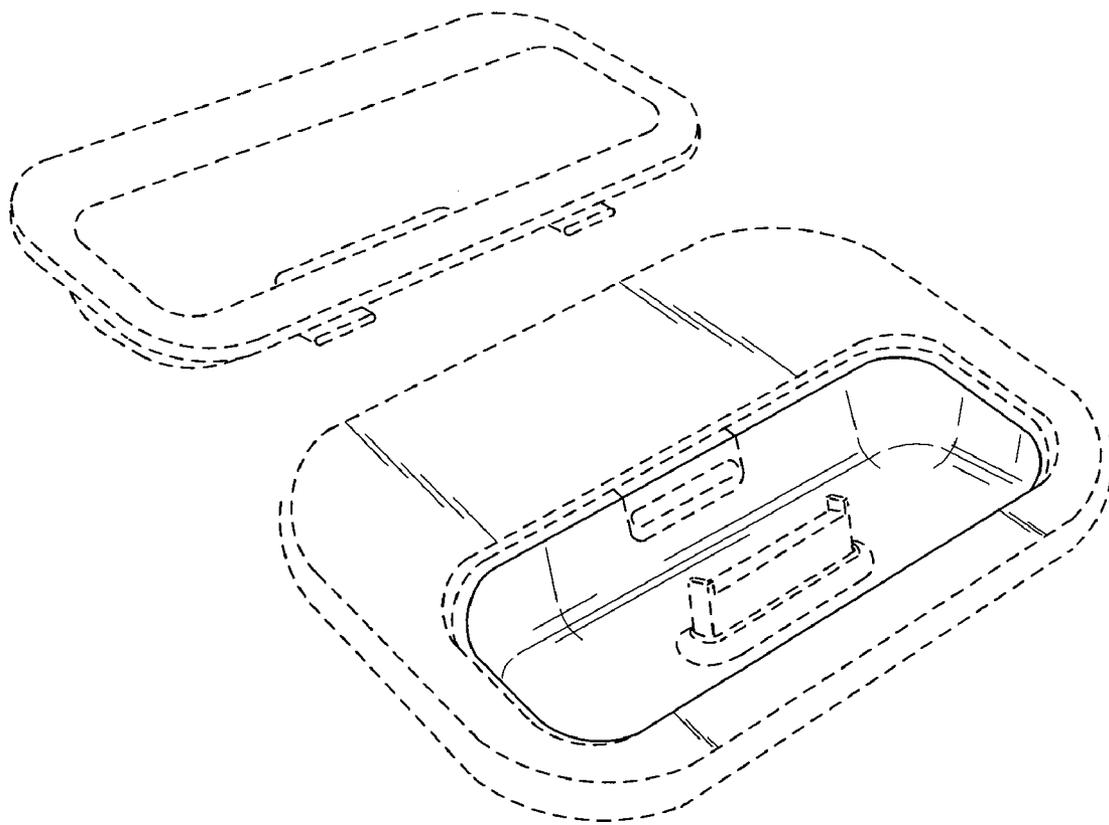


FIG. 4

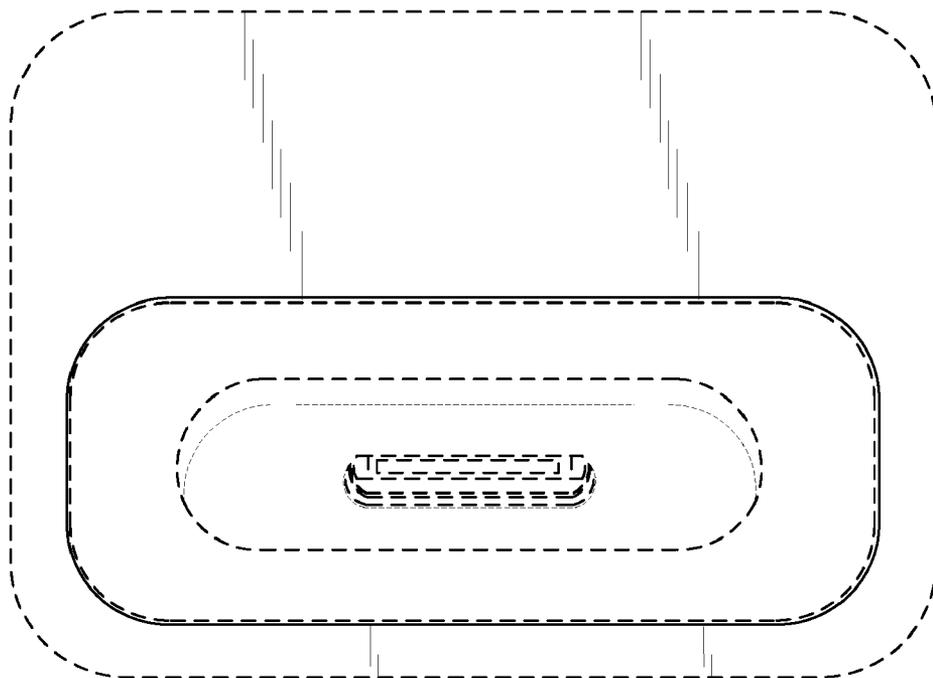


FIG. 5

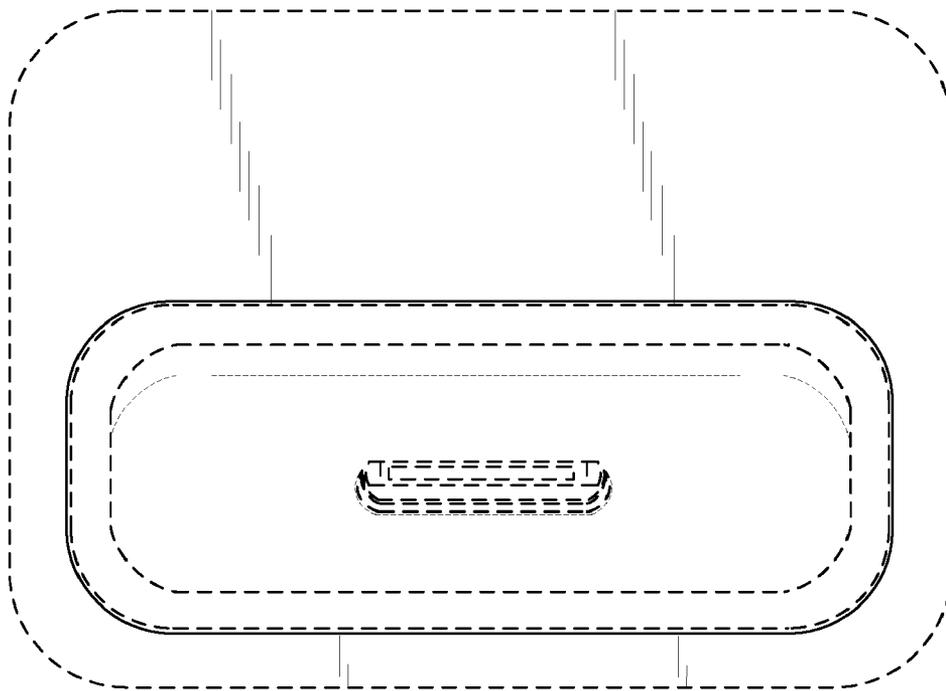


FIG. 6

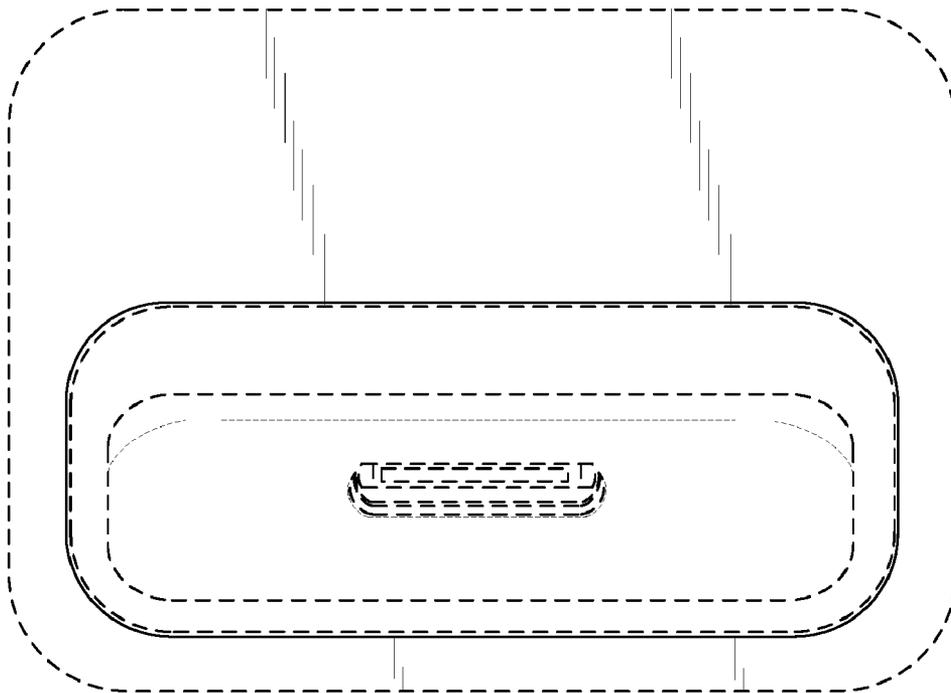


FIG. 7

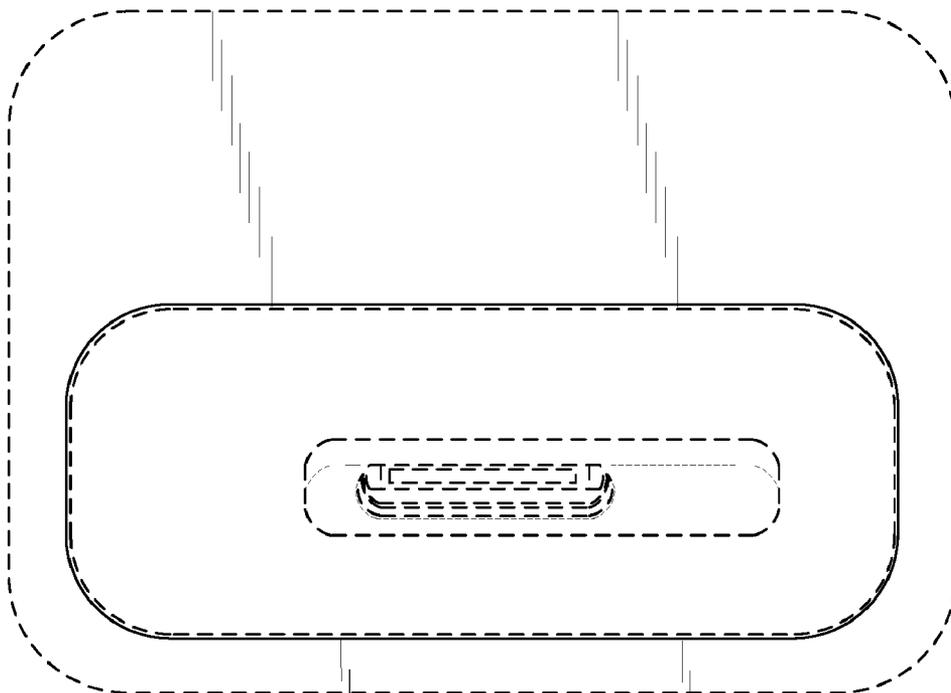


FIG. 8

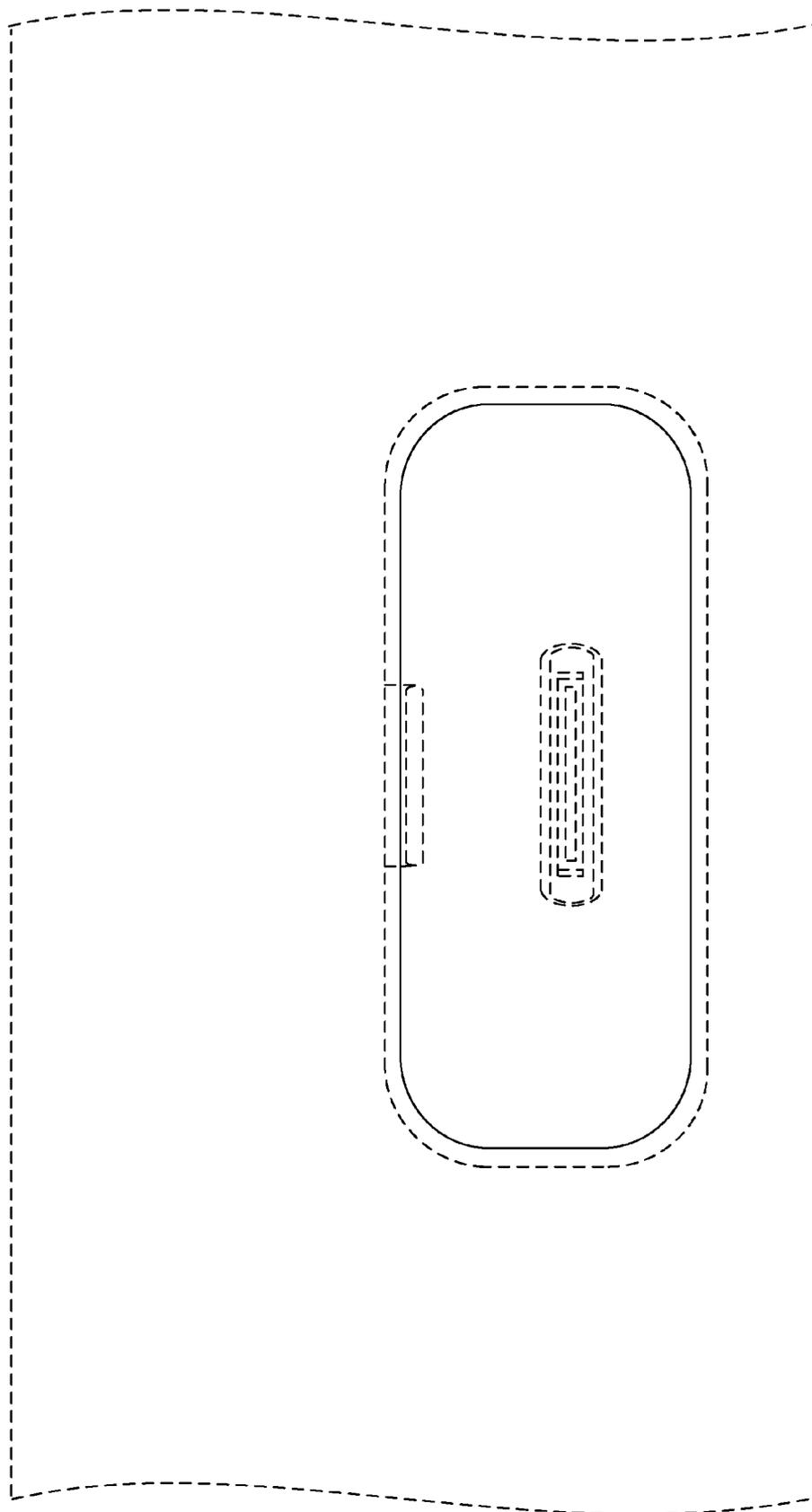


FIG. 9

EXHIBIT K

Int. Cl.: 9

Prior U.S. Cls.: 21, 23, 26, 36 and 38

Reg. No. 3,679,056

United States Patent and Trademark Office

Registered Sep. 8, 2009

TRADEMARK
PRINCIPAL REGISTER



APPLE INC. (CALIFORNIA CORPORATION)
1 INFINITE LOOP
CUPERTINO, CA 95014

FOR: COMPUTERS; COMPUTER HARDWARE; COMPUTER PERIPHERALS; HANDHELD COMPUTERS; HANDHELD MOBILE DIGITAL ELECTRONIC DEVICES FOR RECORDING, ORGANIZING, TRANSMITTING, MANIPULATING, AND REVIEWING TEXT, DATA, IMAGE, AUDIO, AND AUDIOVISUAL FILES, FOR THE SENDING AND RECEIVING OF TELEPHONE CALLS, ELECTRONIC MAIL, AND OTHER DIGITAL DATA, FOR USE AS A DIGITAL FORMAT AUDIO PLAYER, HANDHELD COMPUTER, PERSONAL DIGITAL ASSISTANT, ELECTRONIC ORGANIZER, ELECTRONIC NOTEPAD, CAMERA, AND GLOBAL POSITIONING SYSTEM (GPS) ELECTRONIC NAVIGATION DEVICE; DIGITAL AUDIO AND VIDEO RECORDERS AND PLAYERS; PERSONAL DIGITAL ASSISTANTS; ELECTRONIC ORGANIZERS; CAMERAS; TELEPHONES; MOBILE PHONES; SATELLITE NAVIGATIONAL SYSTEMS, NAMELY, GLOBAL POSITIONING SYSTEMS (GPS); ELECTRONIC NAVIGATIONAL DEVICES, NAMELY, GLOBAL POSITIONING SATELLITE (GPS) BASED NAVIGATION RECEIVERS; COMPUTER GAME MACHINES FOR USE WITH EXTERNAL DISPLAY SCREENS, MONITORS, OR TELEVISIONS; A FULL LINE OF ACCESSORIES AND PARTS FOR THE AFOREMENTIONED GOODS; STANDS, COVERS, CASES, HOLSTERS, POWER ADAPTORS, AND WIRED AND WIRELESS REMOTE CONTROLS FOR THE AFOREMENTIONED GOODS; COMPUTER MEMORY HARDWARE; COMPUTER DISC DRIVES; OPTICAL DISC DRIVES; COMPUTER NETWORKING HARD-

WARE; COMPUTER MONITORS; FLAT PANEL DISPLAY MONITORS; COMPUTER KEYBOARDS; COMPUTER CABLES; MODEMS; COMPUTER MICE; ELECTRONIC DOCKING STATIONS; SET TOP BOXES; BATTERIES; BATTERY CHARGERS; ELECTRICAL CONNECTORS, WIRES, CABLES, AND ADAPTORS; DEVICES FOR HANDS-FREE USE; HEADPHONES; EARPHONES; EAR BUDS; AUDIO SPEAKERS; MICROPHONES; AND HEADSETS; A FULL LINE OF COMPUTER SOFTWARE FOR BUSINESS, HOME, EDUCATION, AND DEVELOPER USE; USER MANUALS FOR USE WITH, AND SOLD AS A UNIT WITH, THE AFOREMENTIONED GOODS; DOWNLOADABLE AUDIO AND VIDEO FILES, MOVIES, RING TONES, VIDEO GAMES, TELEVISION PROGRAMS, POD CASTS AND AUDIO BOOKS VIA THE INTERNET AND WIRELESS DEVICES FEATURING MUSIC, MOVIES, VIDEOS, TELEVISION, CELEBRITIES, SPORTS, NEWS, HISTORY, SCIENCE, POLITICS, COMEDY, CHILDREN'S ENTERTAINMENT, ANIMATION, CULTURE, CURRENT EVENTS AND TOPICS OF GENERAL INTEREST, IN CLASS 9 (U.S. CLS. 21, 23, 26, 36 AND 38).

FIRST USE 1-31-1977; IN COMMERCE 1-31-1977.

OWNER OF U.S. REG. NOS. 1,114,431, 2,753,069 AND OTHERS.

THE MARK CONSISTS OF THE DESIGN OF AN APPLE WITH A BITE REMOVED.

SER. NO. 77-648,705, FILED 1-13-2009.

JERI J. FICKES, EXAMINING ATTORNEY

EXHIBIT L

Int. Cl.: 9

Prior U.S. Cls.: 21, 23, 26, 36 and 38

Reg. No. 2,715,578

United States Patent and Trademark Office

Registered May 13, 2003

**TRADEMARK
PRINCIPAL REGISTER**



APPLE COMPUTER, INC. (CALIFORNIA CORPORATION)
1 INFINITE LOOP
CUPERTINO, CA 95014

FOR: COMPUTERS HARDWARE; COMPUTER HARDWARE, NAMELY, SERVER, DESKTOP, LAPTOP, NOTEBOOK AND SUBNOTEBOOK COMPUTERS; HAND HELD AND MOBILE COMPUTERS; COMPUTER TERMINALS AND MONITORS; PERSONAL DIGITAL ASSISTANTS; PORTABLE DIGITAL AUDIO PLAYERS; ELECTRONIC ORGANIZERS; COMPUTER KEYBOARDS, CABLES, MODEMS; AUDIO SPEAKERS; COMPUTER VIDEO CONTROL DEVICES, NAMELY, COMPUTER MICE, TRACKBALLS, JOYSTICKS AND GAMEPADS; A FULL LINE OF COMPUTER SOFTWARE FOR BUSINESS, HOME, EDUCATION, AND DEVELOPER USE; COMPUTER PROGRAMS FOR PERSONAL INFORMATION MANAGEMENT; DATABASE MANAGEMENT SOFTWARE; CHARACTER RECOGNITION SOFTWARE; TELEPHONY MANAGEMENT SOFTWARE; ELECTRONIC MAIL AND MESSAGING SOFTWARE; TELECOMMUNICATIONS SOFTWARE, NAMELY FOR PAGING; DATABASE SYNCHRONIZATION SOFTWARE; COMPUTER PROGRAMS FOR ACCESSING, BROWSING AND SEARCHING ONLINE DATABASES; OPERATING SYSTEM SOFTWARE; APPLICATION DEVELOPMENT TOOL PROGRAMS; BLANK COMPUTER STORAGE MEDIA; FONTS, TYPEFACES, TYPE DESIGNS AND SYMBOLS RECORDED ON MAGNETIC MEDIA; COMPUTER SOFTWARE FOR USE IN PROVIDING MULTIPLE USER ACCESS TO A GLOBAL COMPUTER INFORMATION NETWORK FOR SEARCHING, RETRIEVING,

TRANSFERRING, MANIPULATING AND DISSEMINATING A WIDE RANGE OF INFORMATION; COMPUTER SOFTWARE FOR USE AS A PROGRAMMING INTERFACE; COMPUTER SOFTWARE FOR USE IN NETWORK SERVER SHARING; LOCAL AND WIDE AREA NETWORKING SOFTWARE; COMPUTER SOFTWARE FOR MATCHING, CORRECTION, AND REPRODUCTION OF COLOR; COMPUTER SOFTWARE FOR USE IN DIGITAL VIDEO AND AUDIO EDITING; COMPUTER SOFTWARE FOR USE IN ENHANCING TEXT AND GRAPHICS; COMPUTER SOFTWARE FOR USE IN FONT JUSTIFICATION AND FONT QUALITY; COMPUTER SOFTWARE FOR USE TO NAVIGATE AND SEARCH A GLOBAL COMPUTER INFORMATION NETWORK, AS WELL AS TO ORGANIZE AND SUMMARIZE THE INFORMATION RETRIEVED; COMPUTER SOFTWARE FOR USE IN WORD PROCESSING AND DATABASE MANAGEMENT; WORD PROCESSING SOFTWARE INCORPORATING TEXT, SPREADSHEETS, STILL AND MOVING IMAGES, SOUNDS AND CLIP ART; COMPUTER SOFTWARE FOR USE IN AUTHORIZING, DOWNLOADING, TRANSMITTING, RECEIVING, EDITING, EXTRACTING, ENCODING, DECODING, PLAYING, STORING AND ORGANIZING AUDIO, VIDEO, STILL IMAGES AND OTHER DIGITAL DATA; COMPUTER SOFTWARE FOR ANALYZING AND TROUBLESHOOTING OTHER COMPUTER SOFTWARE; CHILDREN'S EDUCATIONAL SOFTWARE; COMPUTER GAME SOFTWARE; COMPUTER GRAPHICS SOFTWARE; COMPUTER SEARCH ENGINE SOFTWARE; WEB SITE DEVELOPMENT SOFTWARE; COMPUTER PROGRAM WHICH PROVIDES REMOTE VIEWING, REMOTE CONTROL, COMMUNICATIONS

AND SOFTWARE DISTRIBUTION WITHIN PERSONAL COMPUTER SYSTEMS AND ACROSS COMPUTER NETWORK; COMPUTER PROGRAMS FOR FILE MAINTENANCE AND DATA RECOVERY; COMPUTER PERIPHERALS; INSTRUCTIONAL MANUALS PACKAGED IN ASSOCIATION WITH THE ABOVE , IN CLASS 9 (U.S. CLS. 21, 23, 26, 36 AND 38).

FIRST USE 1-1-1977; IN COMMERCE 1-1-1977.

OWNER OF U.S. REG. NOS. 1,078,312, 2,180,949 AND OTHERS.

SER. NO. 76-426,501, FILED 7-1-2002.

JAMES A. RAUEN, EXAMINING ATTORNEY

EXHIBIT M

Int. Cl.: 9

Prior U.S. Cls.: 21, 23, 26, 36, and 38

Reg. No. 3,341,286

United States Patent and Trademark Office

Registered Nov. 20, 2007

TRADEMARK
PRINCIPAL REGISTER



APPLE INC. (CALIFORNIA CORPORATION)
1 INFINITE LOOP
CUPERTINO, CA 95014

FOR: FULL LINE OF ELECTRONIC AND MECHANICAL PARTS AND FITTINGS FOR PORTABLE AND HANDHELD DIGITAL ELECTRONIC DEVICES FOR RECORDING, ORGANIZING, TRANSMITTING, MANIPULATING, AND REVIEWING TEXT, DATA, AUDIO AND VIDEO FILES; ELECTRONIC DOCKING STATIONS; STANDS SPECIALLY DESIGNED FOR HOLDING PORTABLE AND HANDHELD DIGITAL ELECTRONIC DEVICES; BATTERY CHARGERS; BATTERY PACKS; ELECTRICAL CONNECTORS, WIRES, CABLES, AND ADAPTORS; WIRED AND WIRELESS REMOTE CONTROLS FOR PORTABLE AND HANDHELD DIGITAL ELECTRONIC DEVICES; HEADPHONES AND EARPHONES; STEREO AMPLIFIER AND SPEAKER BASE STATIONS; AUTO-

MOBILE STEREO ADAPTORS; AUDIO RECORDERS; RADIO RECEIVERS; RADIO TRANSMITTERS; VIDEO VIEWERS, NAMELY, VIDEO MONITORS FOR PORTABLE AND HANDHELD DIGITAL ELECTRONIC DEVICES; AND CARRYING CASES, ALL FOR USE WITH PORTABLE AND HANDHELD DIGITAL ELECTRONIC DEVICES FOR RECORDING, ORGANIZING, TRANSMITTING, MANIPULATING, AND REVIEWING TEXT, DATA, AUDIO, IMAGE, AND VIDEO FILES, IN CLASS 9 (U.S. CLS. 21, 23, 26, 36 AND 38).

FIRST USE 3-0-2005; IN COMMERCE 3-0-2005.

OWNER OF U.S. REG. NOS. 2,781,793 AND 2,835,698.

SN 78-689,534, FILED 8-10-2005.

CAROLINE WOOD, EXAMINING ATTORNEY

Court Name: U.S. District Court, NDCA
Division: 5
Receipt Number: 54611007757
Cashier ID: harwellt
Transaction Date: 07/22/2010
Payer Name: San Francisco Legal Support

CIVIL FILING FEE
For: Apple Inc.
Case/Party: D-CAN-3-10-CV-003216-001
Amount: \$350.00

CHECK
Check/Money Order Num: 66244
Amt Tendered: \$350.00

Total Due: \$350.00
Total Tendered: \$350.00
Change Amt: \$0.00

Case # 10-cv-3216-EDL

Checks and drafts are accepted
subject to collections and full
credit will only be given when the
check or draft has been accepted by
the financial institution on which
it was drawn.