US 20070111796A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2007/0111796 A1

C. Giaimo, III et al. (43) Pub. Date: May 17, 2007

(54) ASSOCIATION OF PERIPHERALS COMMUNICATIVELY ATTACHED TO A CONSOLE DEVICE

(75) Inventors: **Edward C. Giaimo III**, Bellevue, WA (US); **Richard Henry Irving**, Kirkland, WA (US); **Shaheen Ashok Gandhi**, Kirkland, WA (US); **Russell Glaser**, Woodinville, WA (US); **Hugh Edward McLoone JR.**, Bellevue, WA (US); **Jon Marcus Randall Whitten**, Sammamish, WA (US)

Correspondence Address:
WOODCOCK WASHBURN LLP
(MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891 (US)

(73) Assignee: **Microsoft Corporation**, Redmond, WA

(21) Appl. No.: **11/280,736**

(22) Filed: **Nov. 16, 2005**

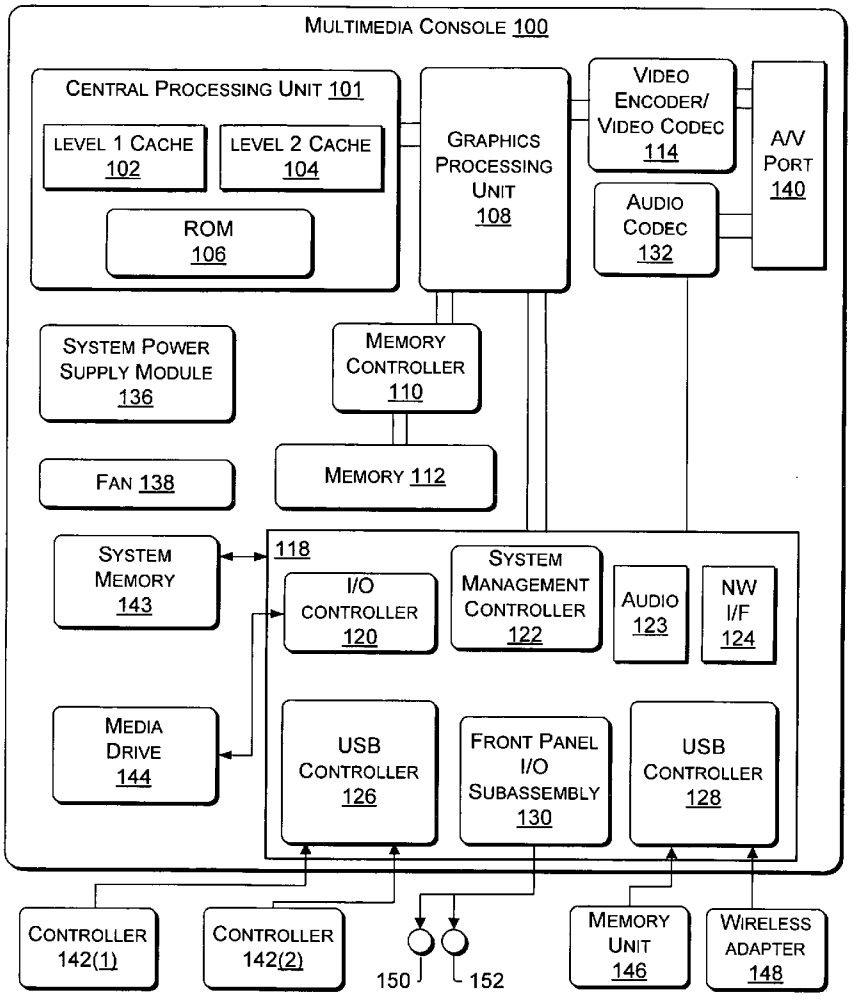
Publication Classification

(51) **Int. Cl.**
A63F 9/24 (2006.01)

(52) **U.S. Cl.** **463/42**

(57) **ABSTRACT**

Systems and methods for associating a device to a peripheral that is communicating to a game console or computing device. The peripheral is initially bound to a port of the game console or computing device. The device binds to the game console or computing device via an automatic or user-initiated sequence and then correlated to the port assigned to the peripheral. Data that is associated with the peripheral is communicated to the device after being correlated. The device and peripheral may also be associated based on a user profile. When the user signs-in to the game console or computing device, the device is associated to the peripheral via a peripheral identifier and configuration information in the profile.



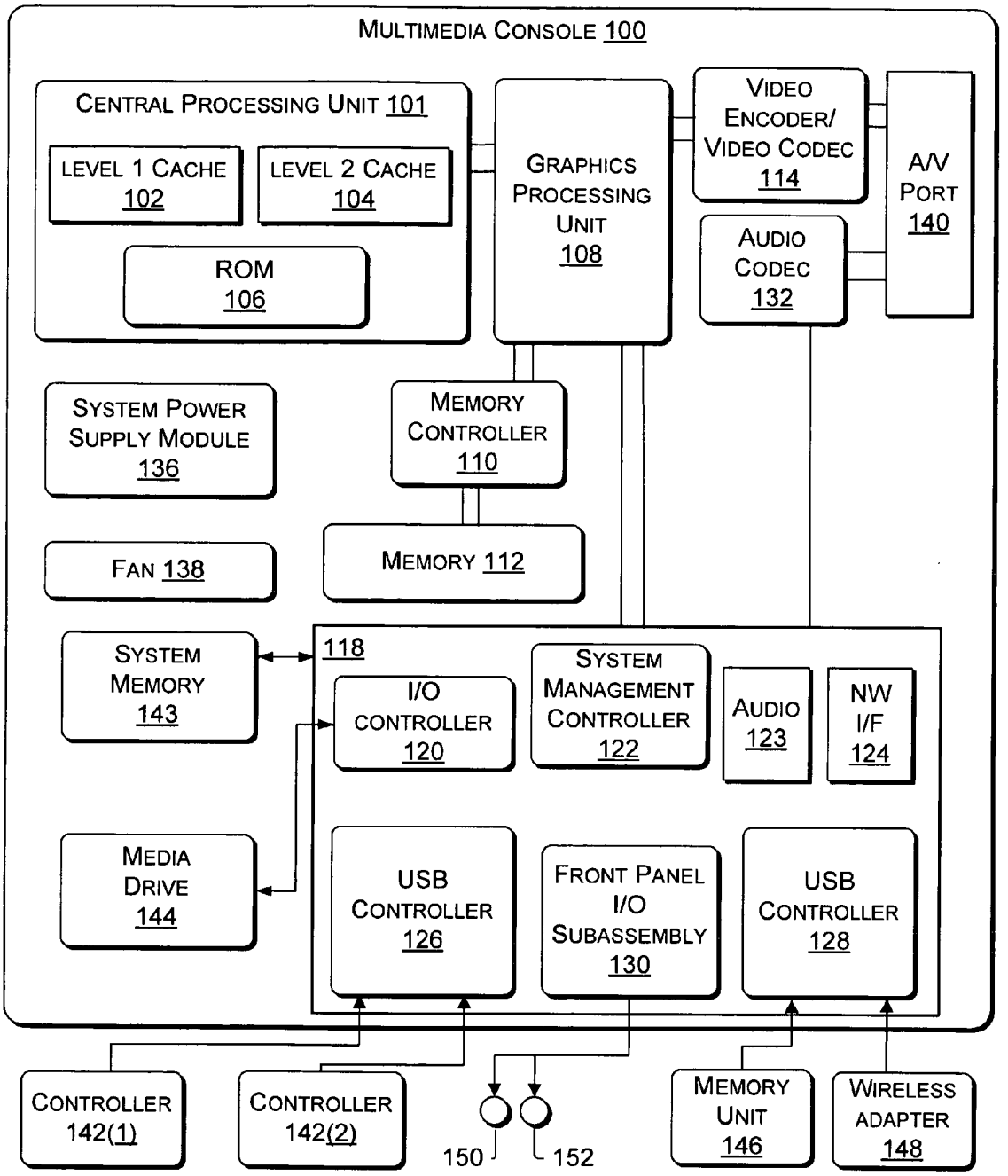

MULTIMEDIA CONSOLE 100
CENTRAL PROCESSING UNIT 101
LEVEL 1 CACHE 102
LEVEL 2 CACHE 104
ROM 106
GRAPHICS PROCESSING UNIT 108
VIDEO ENCODER/ VIDEO CODEC 114
A/V PORT 140
AUDIO CODEC 132
SYSTEM POWER SUPPLY MODULE 136
MEMORY CONTROLLER 110
MEMORY 112
FAN 138
118
SYSTEM MEMORY 143
I/O CONTROLLER 120
SYSTEM MANAGEMENT CONTROLLER 122
AUDIO 123
NW I/F 124
MEDIA DRIVE 144
USB CONTROLLER 126
FRONT PANEL I/O SUBASSEMBLY 130
USB CONTROLLER 128
CONTROLLER 142(1)
CONTROLLER 142(2)
150
152
MEMORY UNIT 146
WIRELESS ADAPTER 148

Fig. 1

Fig. 2

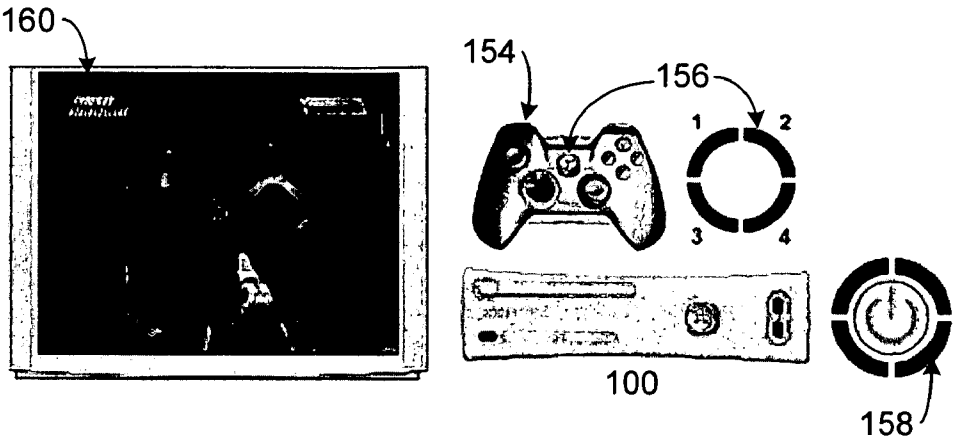

160
154
156
1
2
3
4
100
158

Fig. 3

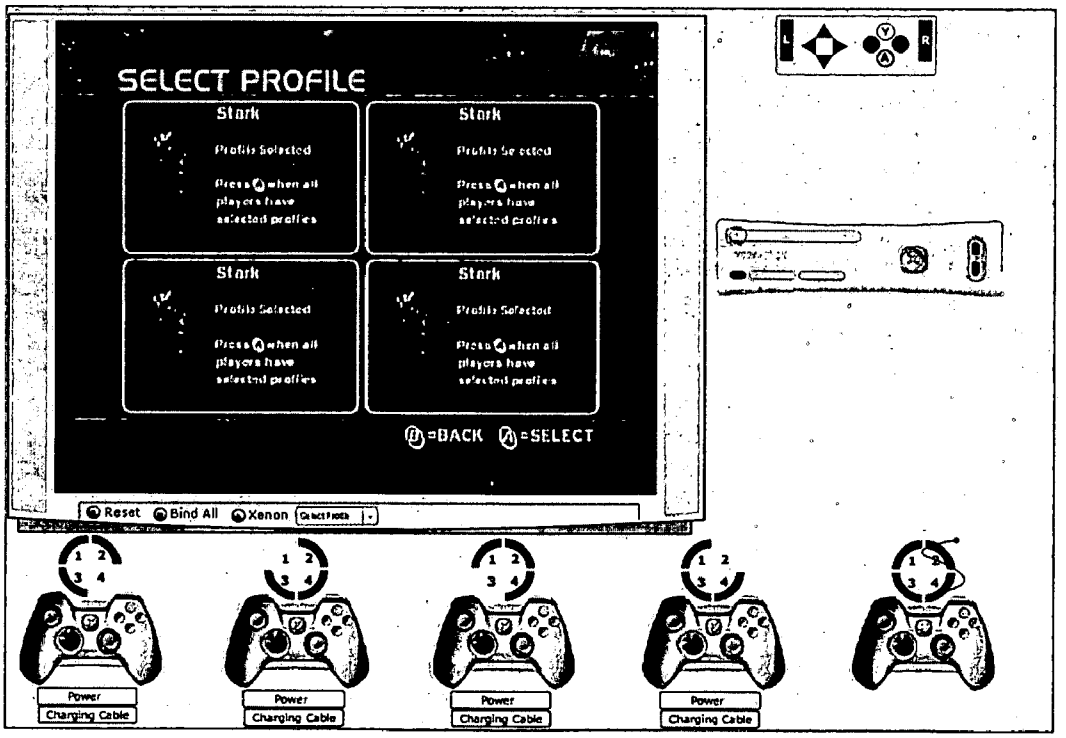

SELECT PROFILE
Stark
Stark
Stark
Stark
=BACK
=SELECT
Reset
Bind All
Xenon
Power
Charging Cable

Fig. 4

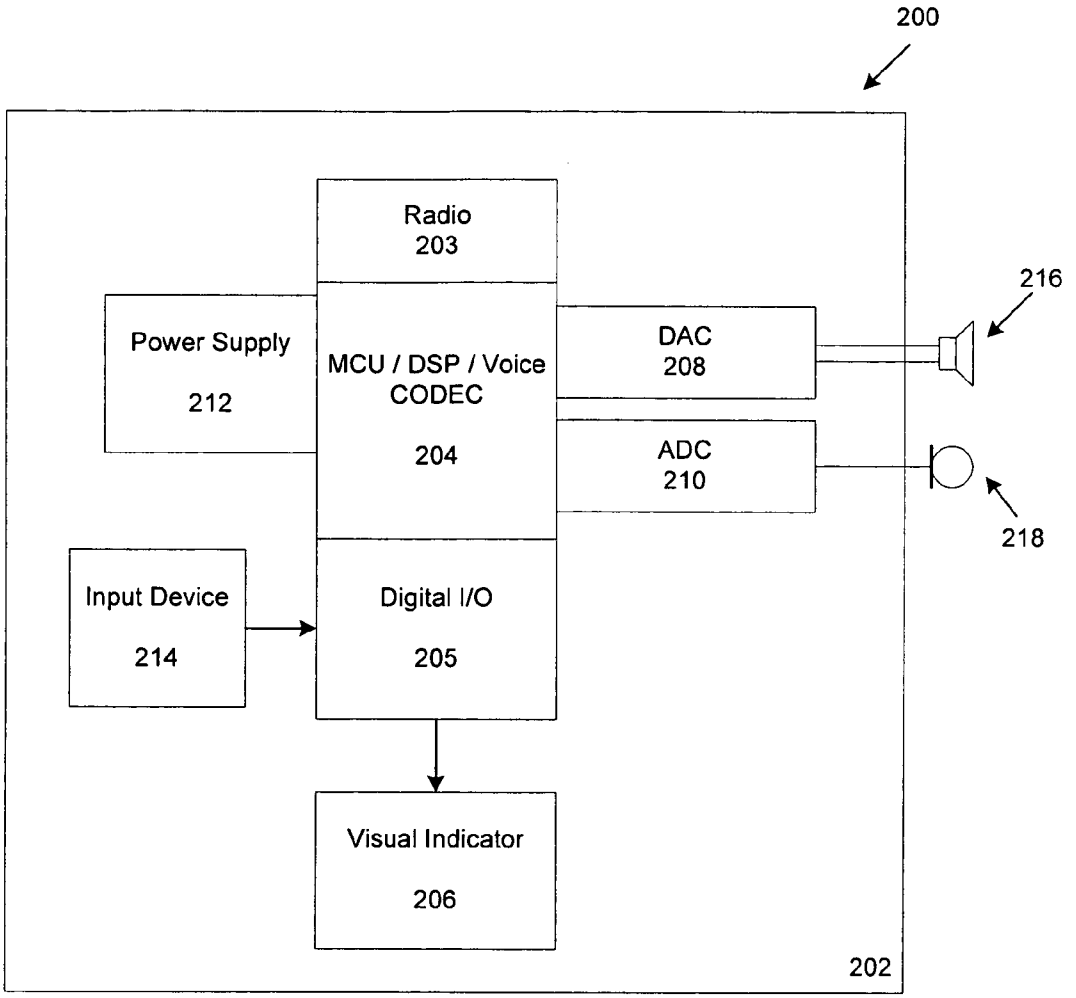

200
Radio
203
Power Supply
212
MCU / DSP / Voice
CODEC
204
DAC
208
ADC
210
216
218
Input Device
214
Digital I/O
205
Visual Indicator
206
202

Fig. 5

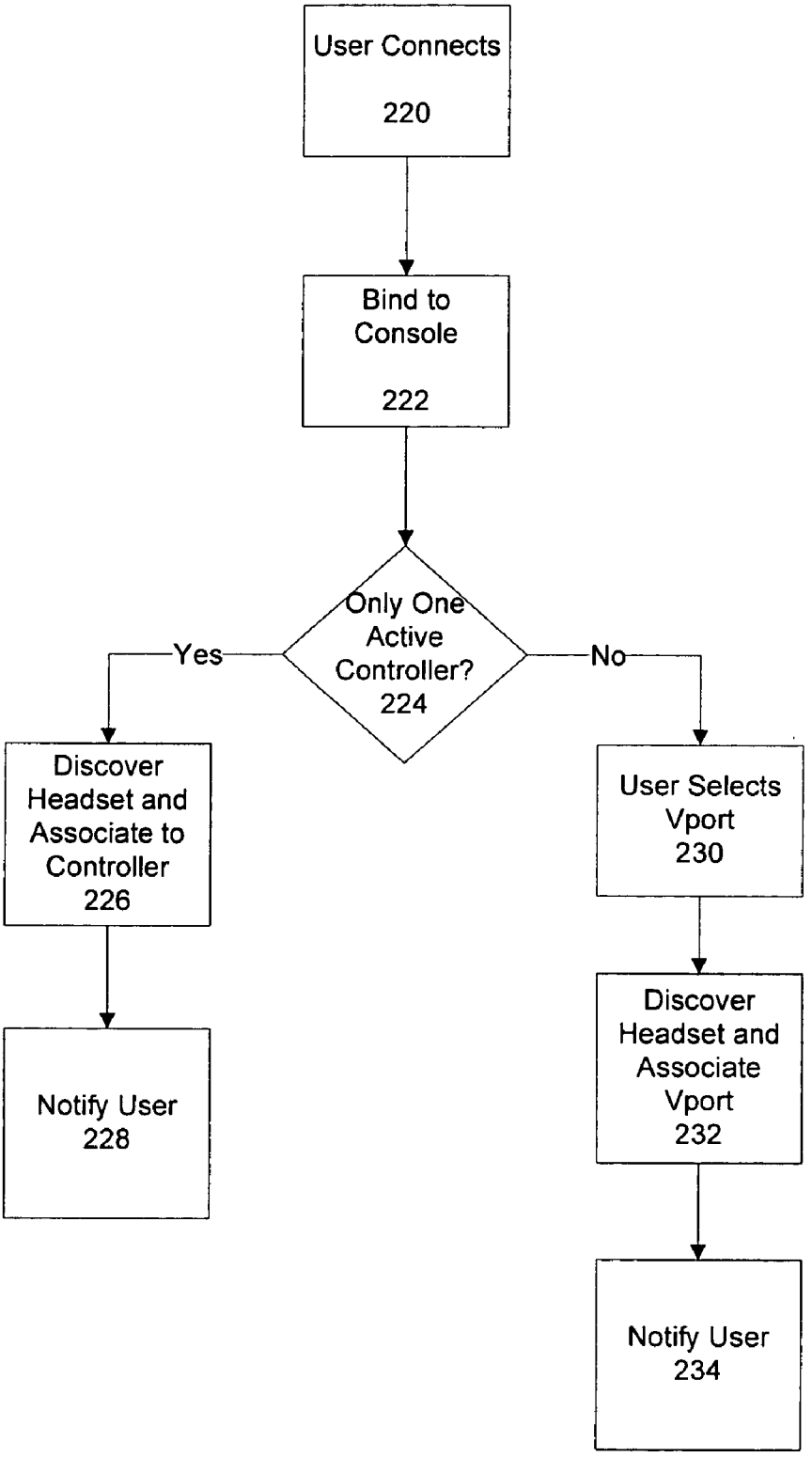

User Connects
220
Bind to Console
222
Only One Active Controller?
224
Yes
No
Discover Headset and Associate to Controller
226
Notify User
228
User Selects Vport
230
Discover Headset and Associate Vport
232
Notify User
234

Fig. 6

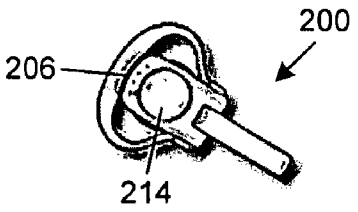


Fig. 7

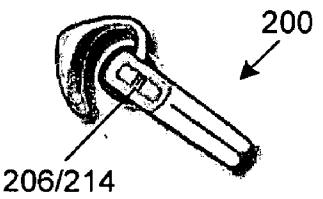


Fig. 8

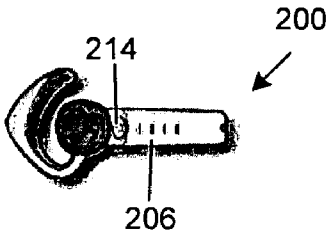


Fig. 9

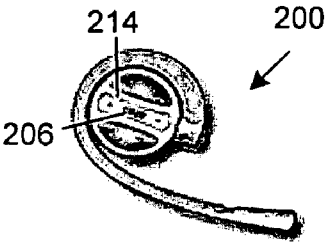


Fig. 10

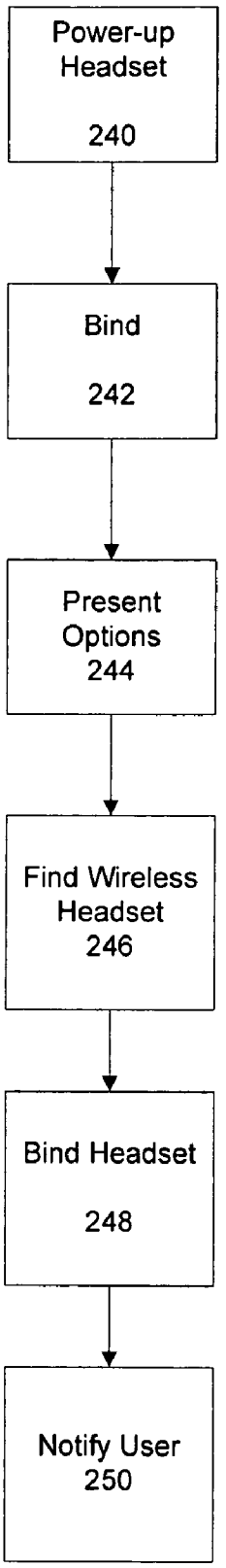
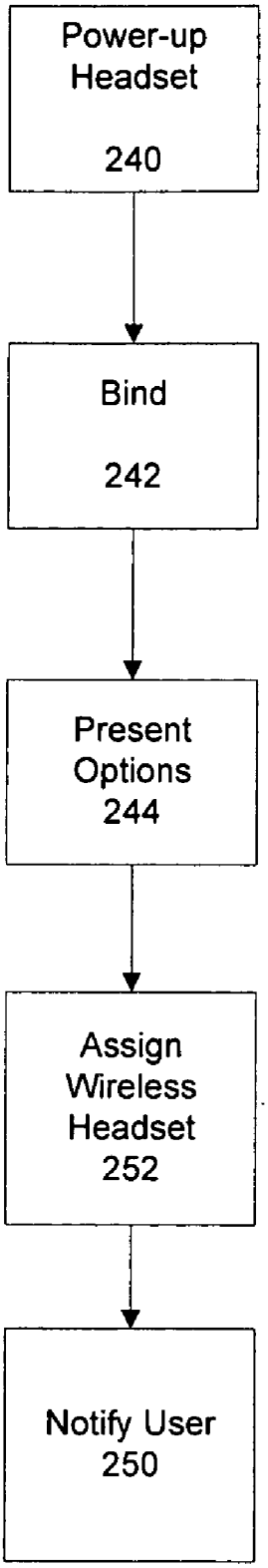

Power-up Headset
240
Bind
242
Present Options
244
Find Wireless Headset
246
Bind Headset
248
Notify User
250

Fig. 11


Power-up Headset 240
Bind 242
Present Options 244
Assign Wireless Headset 252
Notify User 250

ASSOCIATION OF PERIPHERALS COMMUNICATIVELY ATTACHED TO A CONSOLE DEVICE

COPYRIGHT NOTICE/PERMISSION

[0001] A portion of the disclosure of this patent document contains material, which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto: Copyright® 2005, Microsoft Corporation, All Rights Reserved.

TECHNICAL FIELD

[0002] This invention generally relates to the field of computing and gaming devices. The present invention is directed to associating peripheral devices that communicate to a console, controller or other computing device.

BACKGROUND

[0003] Online gaming has become a large part of the gaming experience. Initially, players could communicate via text, however texting during game play is difficult. Later, games allowed players to connect to each other or a centralized server to facilitate cooperation using audio-only devices, such as headsets. Now, many gamers use specially designed headsets for these purposes. Conventional headsets connect to the gaming consoles using, e.g., a headset or microphone jack or USB connector. Because of the wired connection to the controller or console, it is relatively easy to correlate the sounds to the particular gamer's play.

[0004] Gamers also enjoy using wireless controllers, which provide players with freedom of movement by wirelessly connecting the controller to the gaming console. Typically, the wireless controllers provide features such as vibration feedback, mini-joysticks, D-pad, pressure-sensitive buttons, etc. that players would find on wired controllers. In addition, the systems that connect wireless controllers to gaming consoles often allow multiple players to play at once on the console.

[0005] Conventionally, associating headset audio with game controllers is performed by plugging-in the wired headset to a jack in the wired or wireless controller. However, when the headset is wireless, due to the nature of wireless devices, there is no physical mechanism to associate the headset to a particular gamer's play or controller. However, due to nature of wireless devices, there is no physical mechanism to associate a wireless headset to a particular gamer's play or controller.

SUMMARY

[0006] Systems and methods for associating a device to a peripheral that is communicating to a game console or computing device are disclosed herein. The peripheral is initially bound to a port of the game console or computing device. The peripheral binds to the game console or computing device via an automatic or user-initiated sequence and then correlated to the port assigned to the peripheral. Data that is associated with the peripheral is communicated to the device after being correlated. The game console (or computing device) and peripheral may also be associated based on a user profile.

[0007] A non-limiting example of the above is a wireless headset that is used by garners during game play. The headset may be associated with the gamer's controller such that game-related audio associated with the gamer's play is sent from the game console to headset. The gamer may also communicate with other garners using the headset.

[0008] Additional features and advantages will be made apparent from the following detailed description that proceeds with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The foregoing summary, as well as the following detailed description of preferred embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the drawings exemplary constructions of the invention; however, the invention is not limited to the specific methods and instrumentalities disclosed. In the drawings:

[0010] FIG. **1** is a block diagram showing a gaming console in which aspects of the present invention may be implemented;

[0011] FIG. **2** illustrates a controller and LED indicators;

[0012] FIG. **3** illustrates one or more controllers/peripherals being bound and discovered by the console;

[0013] FIG. **4** illustrates a block diagram of a wireless device;

[0014] FIG. **5** illustrates exemplary processes performed to associate the wireless device with a peripheral;

[0015] FIGS. **6-9** illustrate exemplary wireless device designs; and

[0016] FIGS. **10-11** illustrate alternative processes performed to associate the wireless device with a peripheral.

DETAILED DESCRIPTION

[0017] FIG. **1** illustrates the functional components of a multimedia/gaming console **100** in which certain aspects of the present invention may be implemented. The multimedia console **100** has a central processing unit (CPU) **101** having a level **1** cache **102**, a level **2** cache **104**, and a flash ROM (Read Only Memory) **106**. The level **1** cache **102** and a level **2** cache **104** temporarily store data and hence reduce the number of memory access cycles, thereby improving processing speed and throughput. The CPU **101** may be provided having more than one core, and thus, additional level **1** and level **2** caches **102** and **104**. The flash ROM **106** may store executable code that is loaded during an initial phase of a boot process when the multimedia console **100** is powered ON.

[0018] A graphics processing unit (GPU) **108** and a video encoder/video codec (coder/decoder) **114** form a video processing pipeline for high speed and high resolution graphics processing. Data is carried from the graphics processing unit **108** to the video encoder/video codec **114** via a bus. The video processing pipeline outputs data to an A/V (audio/

video) port **140** for transmission to a television or other display. A memory controller **110** is connected to the GPU **108** to facilitate processor access to various types of memory **112**, such as, but not limited to, a RAM (Random Access Memory).

[**0019**] The multimedia console **100** includes an I/O controller **120**, a system management controller **122**, an audio processing unit **123**, a network interface controller **124**, a first USB host controller **126**, a second USB controller **128** and a front panel I/O subassembly **130** that are preferably implemented on a module **118**. The USB controllers **126** and **128** serve as hosts for peripheral controllers **142(1)**-**142(2)**, a wireless adapter **148**, and an external memory device **146** (e.g., flash memory, external CD/DVD ROM drive, removable media, etc.). It is noted that additional USB controllers may be provided. The network interface **124** and/or wireless adapter **148** provide access to a network (e.g., the Internet, home network, etc.) and may be any of a wide variety of various wired or wireless adapter components including an Ethernet card, a modem, a Bluetooth module, a cable modem, and the like.

[**0020**] System memory **143** is provided to store application data that is loaded during the boot process. A media drive **144** is provided and may comprise a DVD/CD drive, hard drive, or other removable media drive, etc. The media drive **144** may be internal or external to the multimedia console **100**. Application data may be accessed via the media drive **144** for execution, playback, etc. by the multimedia console **100**. The media drive **144** is connected to the I/O controller **120** via a bus, such as a Serial ATA bus or other high speed connection (e.g., IEEE 1394).

[**0021**] The system management controller **122** provides a variety of service functions related to assuring availability of the multimedia console **100**. The audio processing unit **123** and an audio codec **132** form a corresponding audio processing pipeline with high fidelity and stereo processing. Audio data is carried between the audio processing unit **123** and the audio codec **132** via a communication link. The audio processing pipeline outputs data to the A/V port **140** for reproduction by an external audio player or device having audio capabilities.

[**0022**] The front panel I/O subassembly **130** supports the functionality of the power button **150** and the eject button **152**, as well as any LEDs (light emitting diodes) or other indicators exposed on the outer surface of the multimedia console **100**. A system power supply module **136** provides power to the components of the multimedia console **100**. A fan **138** cools the circuitry within the multimedia console **100**.

[**0023**] The CPU **101**, GPU **108**, memory controller **110**, and various other components within the multimedia console **100** are interconnected via one or more buses, including serial and parallel buses, a memory bus, a peripheral bus, and a processor or local bus using any of a variety of bus architectures. By way of example, such architectures can include a Peripheral Component Interconnects (PCI) bus, PCI-Express bus, etc.

[**0024**] When the multimedia console **100** is powered ON, application data may be loaded from the system memory **143** into memory **112** and/or caches **102**, **104** and executed on the CPU **101**. The application may present a graphical user interface that provides a consistent user experience when navigating to different media types available on the multimedia console **100**. In operation, applications and/or other media contained within the media drive **144** may be launched or played from the media drive **144** to provide additional functionalities to the multimedia console **100**.

[**0025**] The multimedia console **100** may be operated as a standalone system by simply connecting the system to a television or other display. In this standalone mode, the multimedia console **100** allows one or more users to interact with the system, watch movies, or listen to music. However, with the integration of broadband connectivity made available through the network interface **124** or the wireless adapter **148**, the multimedia console **100** may further be operated as a participant in a larger network community.

[**0026**] When the multimedia console **100** is powered ON, a set amount of hardware resources are reserved for system use by the multimedia console operating system. These resources may include a reservation of memory (e.g., 16 MB), CPU and GPU cycles (e.g., 5%), networking bandwidth (e.g., 8 kbs), etc. Because these resources are reserved at system boot time, the reserved resources do not exist from the application's view.

[**0027**] In particular, the memory reservation preferably is large enough to contain the launch kernel, concurrent system applications and drivers. The CPU reservation is preferably constant such that if the reserved CPU usage is not used by the system applications, an idle thread will consume any unused cycles.

[**0028**] With regard to the GPU reservation, lightweight messages generated by system applications (e.g., popups) are displayed by using a GPU interrupt to schedule code to render popup into an overlay. The amount of memory required for an overlay depends on the overlay area size and the overlay preferably scales with screen resolution. Where a full user interface is used by the concurrent system application, it is preferable to use a resolution independent of application resolution. A scaler may be used to set this resolution such that the need to change frequency and cause a TV resynch is eliminated.

[**0029**] After the multimedia console **100** boots and system resources are reserved, concurrent system applications execute to provide system functionalities. The system functionalities are encapsulated in a set of system applications that execute within the reserved system resources described above. The operating system kernel identifies threads that are system application threads versus gaming application threads. The system applications are preferably scheduled to run on the CPU **101** at predetermined times and intervals in order to provide a consistent system resource view to the application. The scheduling is to minimize cache disruption for the gaming application running on the console.

[**0030**] When a concurrent system application requires audio, audio processing is scheduled asynchronously to the gaming application due to time sensitivity. A multimedia console application manager (described below) controls the gaming application audio level (e.g., mute, attenuate) when system applications are active.

[**0031**] Input devices (e.g., controllers **142(1)** and **142(2)**) are shared by gaming applications and system applications. The input devices are not reserved resources, but are to be

switched between system applications and the gaming application such that each will have a focus of the device. The application manager preferably controls the switching of input stream, without the gaming application's knowledge, and a driver maintains state information regarding focus switches.

[0032] Referring to FIG. **2**, there is illustrated an exemplary wireless controller **154** having a four quadrant LED indicator **156** (and enlarged view) and console **100** having a four quadrant indicator **158**. The controller **154** also includes vibration feedback, mini-joysticks, pressure-sensitive buttons, etc. A game is shown on the screen **160**. The console indicator **158** is shown surrounding a power button, however, other configurations may be implemented. Each quadrant of the ring may be illuminated by an LED, which may be either a single color or bi-colored to illuminate in plural colors. As will be described below, the quadrants may be illuminated in patterns indicating the notifications, system status, binding and discovery.

[0033] To support an environment where multiple consoles **100** and wireless controllers **154** may coexist, each controller is logically "bound" to a single console **100** so that a link is established with only that console **100**. A controller **154** can not be bound to more than one console **100** at a time. Binding is the process by which a console **100** transmits information to a controller **154** that will enable that controller to establish a link with the console **100**. Once "bound" to a console **100**, the controller **154** attempts to establish a link with the console **100** to which it is bound whenever the controller **154** is turned on.

[0034] It is preferable that binding information is retained only in the controller. Binding is one to one with respect to the controller **154**, but it is one to many with respect to the console **100**. Binding, thus, persists on the controller **154** across battery discharge/charge cycles, until a new binding relationship is established. Establishing a binding relationship is attempted when a BIND button on the console and a BIND button on a wireless controller **154** are pressed within a predetermined period of time of each other. Successfully establishing a binding relationship is dependent on successfully establishing a radio communication link and executing a mutual verification algorithm.

[0035] The console is preferably powered up before pressing its BIND button. If a user initiates binding on a controller **154** that is currently connected to a console **100**, the controller **154** drops the connection to the console **100** prior to attempting the binding process. As the binding process operates, a status notification screen may display binding and discovery process (e.g., binding . . . bound . . . discovered). Binding is a one to one event. In other words, pressing the binding button on the console **100** will bind one controller **154** at a time. To bind a second controller **154**, the BIND button on the console **100** is pressed a second time. If binding is not successful within a predetermined time, the console **100** or controller **154** will automatically time out and return to a previous state such that the previous binding relationship is not lost.

[0036] There are four (or other) virtual controller ports on the console **100**, referred to herein as "Vports." The Vports represent the active game controllers connected to the console **100**, either wired or wirelessly. The numbered Vports are automatically assigned to controllers in the order they are connected to the console **100**. Each Vport is represented by a quadrant of the LED indicator **156** and the console indicator **158**. "Discovery" is the process during which a wired or wireless game device is recognized by the console **100**, assigned a Vport, and made available for game play.

[0037] Thus, the acts of "binding" and "discovery" are preferably two different acts. The act of binding is initiated by pressing the BIND buttons on the controller and console. Once bound, the controller will begin the discovery process, and if successful, will be assigned the first available Vport, which in this case is Vport **1** as described. If one to three controllers had previously been bound and discovered, then the next controller discovered would be assigned Vport **2**, **3**, **4**, etc., respectively. If, a total number controllers equaling the total number of Vports were already discovered, then the binding process could still be performed, however no Vport would be available to assign, so the controller would not be assigned a Vport. However it would still be bound to the console and available to be discovered if one of the other controllers were either turned off or bound to a new console.

[0038] Referring to FIG. **3** there is a visualization of the binding and discovery processes and how the LED indicator **156** and the console indicator **158** visually convey the processes to players. As shown in FIG. **3**, the controller has been powered on and the BIND button on the console **100** and the controller have been pressed. After the binding process has completed, the discovery process takes place. Because this is the first controller to be discovered by the console **100**, it is associated with Vport **1** and the top left quadrant of the indicators **156** and **158** will illuminate to signal the connection. If more than one controller is discovered by the console **100**, the other quadrants of indicator **158** are illuminated in succession. Thus, if two controllers are connected, two quadrants of the indicator **158** will illuminate, and so on up to four controllers and four quadrants. It is noted that while additional quadrants are successively illuminated on the console, only a single quadrant is illuminated on any single controller at a time except in error conditions or other status displayed to the user.

[0039] Referring to FIG. **4**, there is illustrated a block diagram of an exemplary wireless device (e.g., headset) **200**. The headset **200** may include an electronics module **202** that houses a radio **203**, a microcontroller (MCU)/digital signal processor (DSP), Voice CODEC **204**, an I/O device **205**, a digital to analog converter (DAC) **208**, an analog to digital to converter (ADC) **210**, a power supply **212**, an input device **214**, and visual indicator **216**. The components within the electronics module **203** connect to a speaker **216** and a microphone **218**.

[0040] The radio **203** may be a Frequency Hopping Spread Spectrum (FHSS) radio operating at the 2.4 GHz frequency band that communicates data (e.g., audio, configuration, etc.) to the console **100**. The MCU/DSP/CODEC **204** processes audio communication to and from the headset **200**. Output audio is communicated via the DAC **208** to the speaker **216**. Input audio is received by the microphone **218** and converted by the ADC **210** to digital information and passed to the MCU/DSP/CODEC **204** to the radio **203** for communication to the console **100**.

[0041] The input device **214**, which will be described in greater detail below with reference to FIGS. **6-9**, is used to associate the headset **200** with a particular Vport. The visual

indicator **206** provides a user with an indication or confirmation of the associated Vport. As will be described below, the audio related to user's game play is communicated to/from the headset **200** based on an association of the headset **200** to a Vport assigned to the user's controller **154** or other discovered peripheral.

[**0042**] As noted above, the wireless headset **200** communicates directly to the console **100**, rather than the controller **154**. As such, the headset performs a binding/discovery similar to the controller **154**. To accomplish this, the wireless headset **200** is associated with console and assigned a Vport. There are separate Vports for controller and voice device ports.

[**0043**] FIG. **5** illustrates the process of binding and discovery for a wireless headset **200**. At step **220**, a user connects (binds) to the console by, e.g., pressing a button on the console **100**, wireless controller **154** or headset **200** (via input device **214**). Preferably, the headset **200** and/or the console **100** is powered-up after the bind button is pressed, which advantageously reduces RF emissions. However, in the case of the headset **200**, the user may first power up the headset **200** or the act of pressing the button may power up the headset **200**. Optionally, step **220** may be accomplished by bringing the headset **200** into communication range of the console **100**. At step **222**, the console **100** initiates and completes bind process with the headset **200**. At step **224**, it is determined if there is one active controller **154**. If so, then at step **226**, the headset **222** is assigned the same Vport as the active controller **154**.

[**0044**] The user is then notified of the success of the association at step **228**. The notification may be audible or visual. An audible confirmation of Vport assignment may be made by sending a sending a chime or tone to the headset. Personalized settings may be ignored/overridden in order to ensure the chime or tone is played on the headset. Visual notification may be made by a flashing LED (or other visual indicator **206**) on the headset. If a single LED is provided on the headset, it may be flashed at a predetermined rate. If several LEDs are provided, an LED pattern may be flashed. It is preferable in the later scenario that that pattern be the same as the associated controller **154**. Further, an on-screen display notification may be used. If the association is unsuccessful, a different set of notifications may be used than for successful associations.

[**0045**] However, if at step **224**, there is more than one active controller **154**, then the user is notified to selects a Vport to which the headset **200** is to be associated by the console **100**. This notification may be performed through a visual on-screen display, audibly or visually on the headset **200** via the indicator **206**. Referring now to FIGS. **6-9**, the user selection of a Vport may be accomplished through an input device **214** such as a dial or jog shuttle (FIG. **6**), slider (FIG. **7**), button (FIG. **8**) or toggle switch (FIG. **9**). One of ordinary skill in the art would recognize that other mechanical input devices may be provided to make a selection of the Vport.

[**0046**] The visual indicator **206** may include LEDs, a seven segment display, etc. to convey to the user which Vport is selected on the headset **200**. The use of the input device aids in resolving/avoiding conflicts between multiple headsets and a particular controller **154**. Further, if multiple headsets are set to the same Vport, it is preferable that the first one to be discovered is assigned to the Vport. At step **232** the headset **200** is discovered and associated by the console **100** to the selected Vport. A notification of the successful association is provided at step **234**. The notification may be similar to that noted above.

[**0047**] Alternatively, the process may be performed using an on-screen display. This may be desirable because the user may find using the on-screen display more familiar and convenient for configuring devices as the on-screen display is used to configure many options for game play. Referring now to FIG. **10**, the on-screen process may begin when the user powers-up the wireless headset (step **240**). Next, a button is pressed on the controller and the user navigates to an Options screen (steps **242** and **244**). A "Find wireless headset" option may be presented, and if selected, the console initiates a bind process with headset (step **246**). An on-screen display is presented indicated to the user to press a bind button on the headset (step **248**). The user presses the headset's bind button and the console completes the bind process to associate the headset with controller. The headset may then notify the user of a successful bind and association (step **250**).

[**0048**] As another alternative, the user may power-up the wireless headset and presses the headset's bind button (steps **240** and **242**). These two steps may be accomplished by a single press of a power button. The user presses a button on the controller and navigates to an Options screen (step **244**). Next the user selects an "Assign wireless headset" button from the Options screen (step **252**). The console initiates bind process with headset and associates headset to the Vport of the controller that initiated the process. The headset and/or on-screen display notify the user of a successful bind and association (step **250**).

[**0049**] The above processes may be repeated for every user that wants to associate a wireless headset with their controller.

[**0050**] Thus, as described above, the console **100** manages the association of headsets to controllers. Headsets are bound to consoles, which allows the management of headsets and other peripherals to evolve over time as the console **100** can be upgraded with new features with relative ease. In addition, it is preferable that a one-to-one association is made between headsets and controllers.

[**0051**] In addition to the binding/discover notifications, a user may be notified if the Vport assignment is lost and/or re-associated. For example, the controller **100** association preferably should be persisted between power/sleep cycles of the console **100**. However, the wireless controller discovery process may reassign Vports after a power cycle. This means a controller **154** associated with Vport**3** before the power cycle may be associated with Vport**1** after. The headset **200** will follow the controller **154** to Vport**1**.

[**0052**] The headset **200** originally associated with the controller **154** should follow the controller **154** to its newly assigned Vport. This may be accomplished through some headset identifier or preferably to a gamer profile used to associate a headset **200** to a controller **154**. In later instance, when a gamer signs-in, the profile determines which headset **200** should be associated to the controller **154** via a unique identifier of the headset **200** associated with the gamer profile. The controller **154** is automatically assigned to a

Vport at time of discovery and the headset **200** is then associated to that Vport, as noted above.

[**0053**] It is possible that a user may wish to use a wired headset during game play. When a wired headset is inserted to the headset jack on a controller, the wireless headset association will be dropped in favor of the wired headset. This is because it is assumed that the wired headset is being used because of a user preference and an accidental insertion of a wired headset is unlikely.

[**0054**] In addition to a control to select a Vport, the headset may include controls for power on/off, microphone mute, volume up/down, connect to console, etc. Indicators may be provided for battery level (e.g., LEDs or audible alert), headset ON, etc. A buzzer or other audible mechanism may be provided to locate a missing headset **200**.

[**0055**] While the processes above have been described with reference to a wireless controller, it is noted they may equally apply to a wired controller to which a wireless headset is to be associated.

[**0056**] While the present invention has been described in connection with the preferred embodiments of the various Figs., it is to be understood that other similar embodiments may be used or modifications and additions may be made to the described embodiment for performing the same function of the present invention without deviating therefrom.

What is claimed:

1. A wireless communication device, comprising:

a radio that communicates to a console;

an selector for selecting a port setting used by said wireless communication device which correlates to a port assigned by said console to a peripheral; and

an indicator that indicates said port setting,

wherein communications associated with said peripheral are communicated to said wireless communication device.

2. The wireless communication device of claim 1, wherein said port setting is automatically assigned to said wireless communication device if said peripheral is the only peripheral assigned a port by said console.

3. The wireless communication device of claim 1, said selector comprising a bind input device, wherein when said bind input device is actuated, said wireless communication device will bind with said console, and wherein said console will provide an on-screen display option to discover said wireless communication device.

4. The wireless communication device of claim 1, wherein said wireless communication device is set to said port setting based on a user profile and an identifier associated with said peripheral.

5. The wireless communication device of claim 1, wherein said notification comprises an audible component.

6. The wireless communication device of claim 1, wherein said indicator is illuminated in a predetermined sequence convey a notification that said wireless communication device has successfully been associated with said peripheral.

7. The wireless communication device of claim 1, wherein communications associated with said peripheral are communicated between said wireless communication device and said console.

8. A method of establishing an association between a wireless device and a peripheral that communicate with a computing device, comprising:

initiating a binding sequence;

recognizing said wireless device and binding it to said computing device;

setting said wireless device to a port setting correlated to a port assigned to peripheral; and

communicating data between said wireless device and said computing device, said data being associated with said peripheral.

9. The method of claim 8, further comprising automatically assigning said port setting if said peripheral is the only assigned peripheral communicating with said computing device.

10. The method of claim 8, said initiating further comprising:

receiving an input via bind input device; and

providing an on-screen display option to discover said wireless device.

11. The method of claim 10, further comprising setting said wireless communication device to said port setting based on a selected on-screen display option.

12. The method of claim 8, further comprising providing a notification that said wireless device has been associated with said peripheral.

13. The method of claim 12, further comprising illuminating a visual indicator in a predetermined sequence to convey said notification.

14. The method of claim 8, said setting further comprising receiving a user input via a mechanical input device to set said wireless device to said port setting.

15. The method of claim 8, further comprising:

establishing communication between a wired device and said computing device;

removing said association of said wireless device and said peripheral; and

establishing said associated between said wired device and said peripheral.

16. The method of claim 7, said setting further comprising:

accessing a user profile;

determining peripheral settings in said user profile; and

setting said port setting based on said peripheral settings and an identifier of said peripheral.

17. A method of establishing an association between a wireless headset and a game controller that communicate with a gaming console, comprising:

initiating a binding sequence to bind said wireless headset to said gaming console;

setting said wireless headset to a port setting correlated to a port assigned to game controller; and

communicating audio between said wireless headset and said gaming console, said audio being associated with said game controller and port.

18. The method of claim 17, said initiating further comprising:

providing an on-screen display option to discover said wireless headset; and

setting said wireless headset to said port setting based on a selected on-screen display option.

19. The method of claim 17, further comprising maintaining a radio in said gaming console in an OFF condition until said binding sequence is initiated.

20. The method of claim 17, said setting further comprising:

accessing a gamer profile;

determining peripheral settings in said gamer profile; and

setting said port setting based on said game controller settings and an identifier of said game controller.

* * * * *