



US009271099B1

(12) **United States Patent**
Sun

(10) **Patent No.:** **US 9,271,099 B1**

(45) **Date of Patent:** **Feb. 23, 2016**

(54) **METHOD AND APPARATUS TO USE AUDIO PORT TO PROVIDE POWER AND DATA COMMUNICATIONS FOR PERIPHERAL DEVICES**

2009/0011652	A1*	1/2009	Koh	439/638
2010/0260341	A1*	10/2010	Sander et al.	381/1
2011/0103608	A1*	5/2011	Wu et al.	381/74
2012/0052910	A1*	3/2012	Mu et al.	455/558
2012/0326736	A1*	12/2012	Chadbourne et al.	324/705
2013/0012796	A1	1/2013	Kak et al.	

(71) Applicant: **Yihuan Sun**, San Jose, CA (US)

(72) Inventor: **Yihuan Sun**, San Jose, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 254 days.

(21) Appl. No.: **13/962,739**

(22) Filed: **Aug. 8, 2013**

Related U.S. Application Data

(60) Provisional application No. 61/692,238, filed on Aug. 23, 2012.

(51) **Int. Cl.**
G06F 3/00 (2006.01)
H04S 1/00 (2006.01)

(52) **U.S. Cl.**
CPC **H04S 1/007** (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,909,555	A	6/1999	Jung
2004/0169971	A1	9/2004	Hinsberger et al.

OTHER PUBLICATIONS

Kuo, Ye-Sheng et al., "Hijacking Power and Bandwidth from the Mobile Phone's Audio Interface," International Symposium on Low Power Electronics and Design, ISLPED'10, Design Contest, Aug. 2010, Austin, Texas, 6 pages.

* cited by examiner

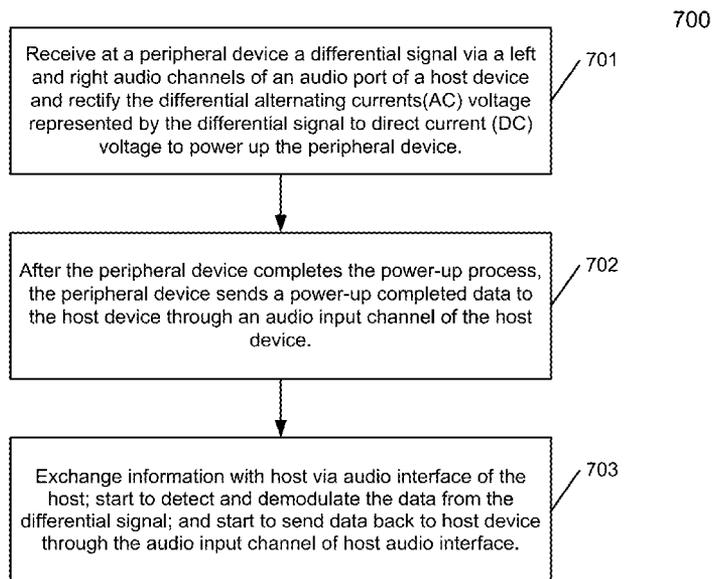
Primary Examiner — Elias Mamo

(74) *Attorney, Agent, or Firm* — Blakely, Sokoloff, Taylor & Zafman LLP

(57) **ABSTRACT**

Techniques for providing power and communicating data via an audio interface to a peripheral device are described herein. In one embodiment, a peripheral device is communicatively coupled to a host device via an audio interface of the host device, where the audio interface includes a first stereo output channel, a second stereo output channel, an audio input channel, and a ground pin. The peripheral device receives from the host device a first audio signal via the first stereo output channel and a second audio signal via the second stereo output channel. In response, the peripheral device harvests energy from both the first audio signal and the second audio signal to generate direct current (DC) power to power one or more hardware components of the peripheral device including a processor of the peripheral device.

22 Claims, 9 Drawing Sheets



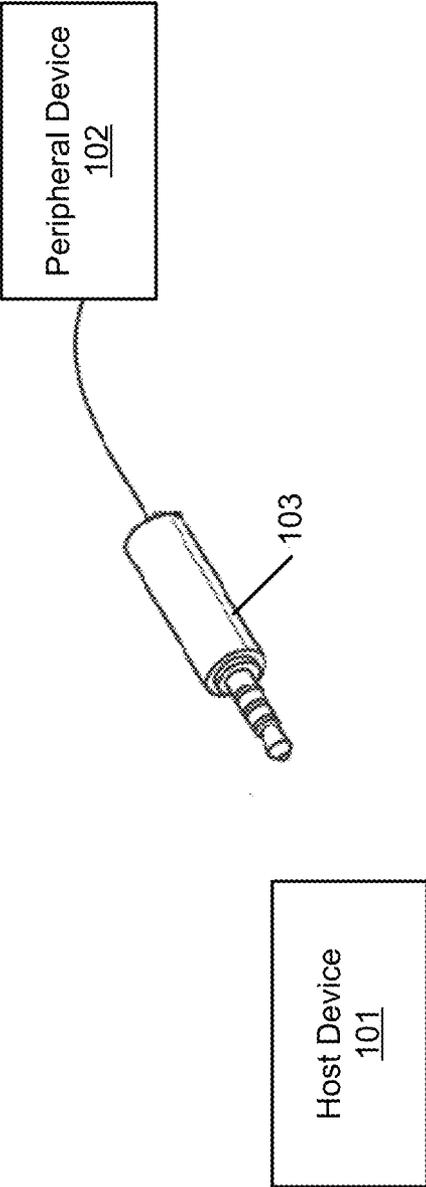


FIG. 1

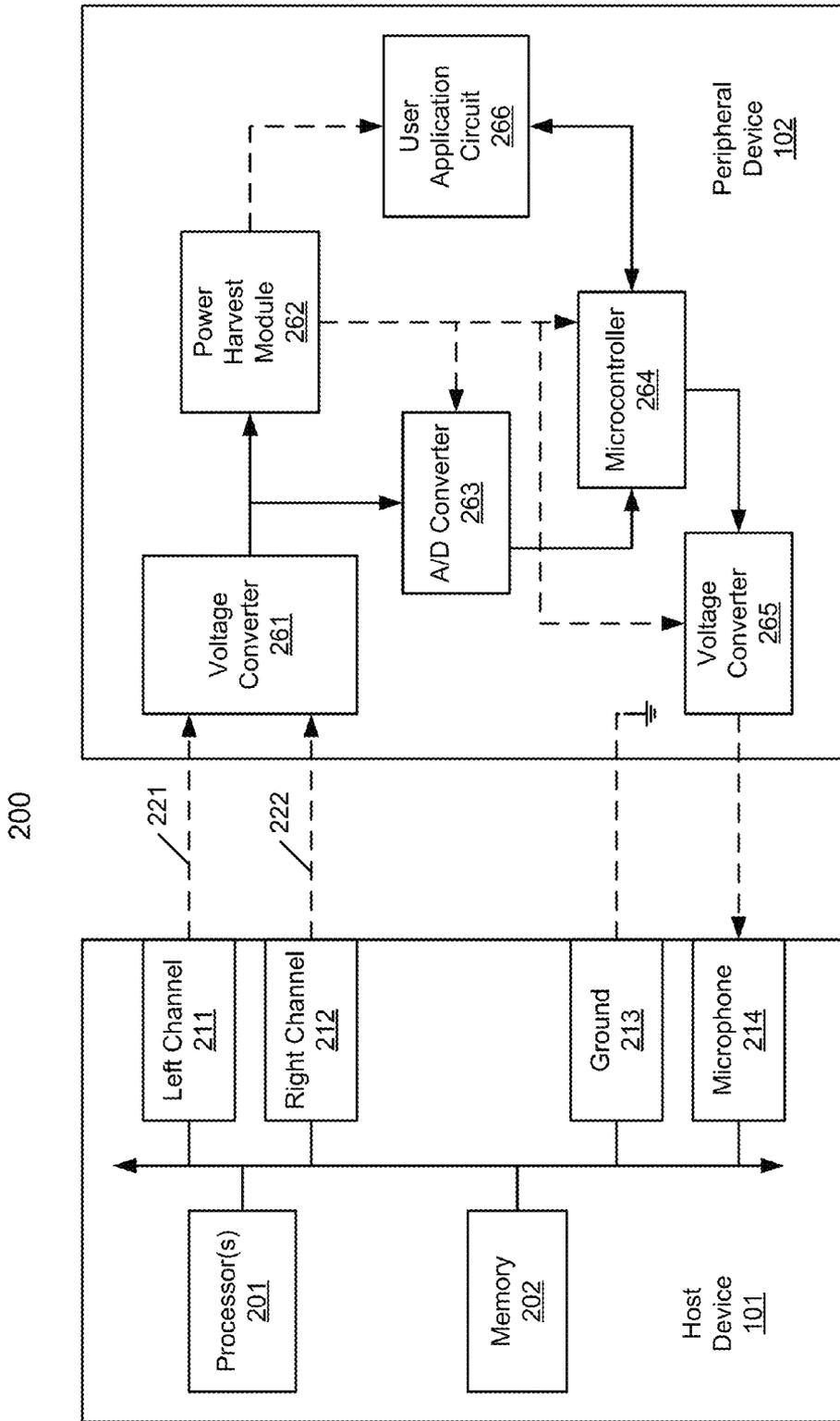


FIG. 2

FIG. 3A

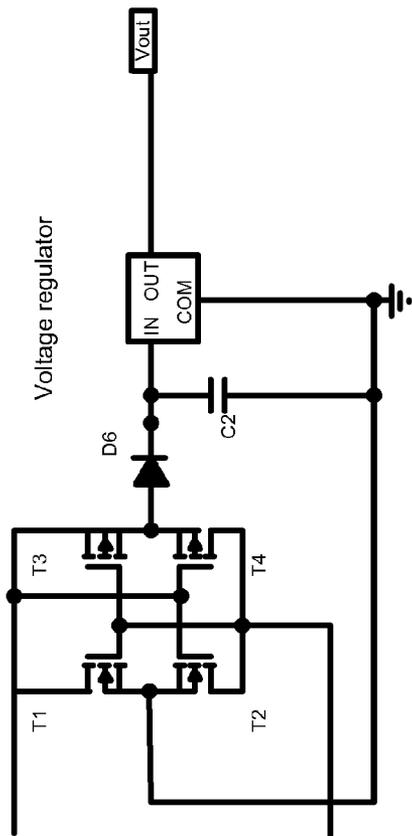
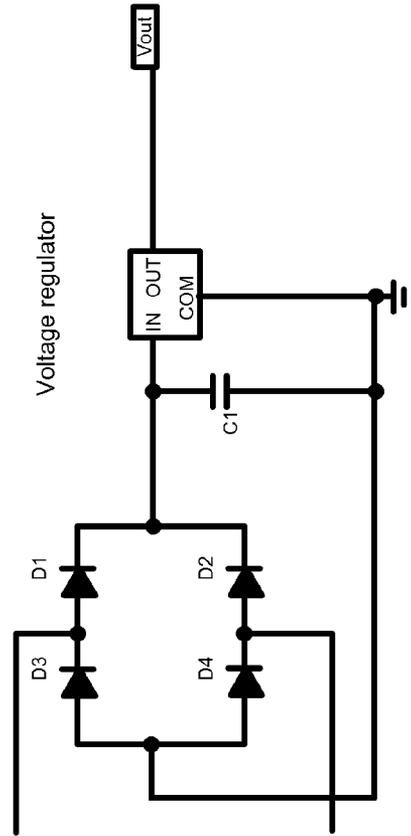


FIG. 3B



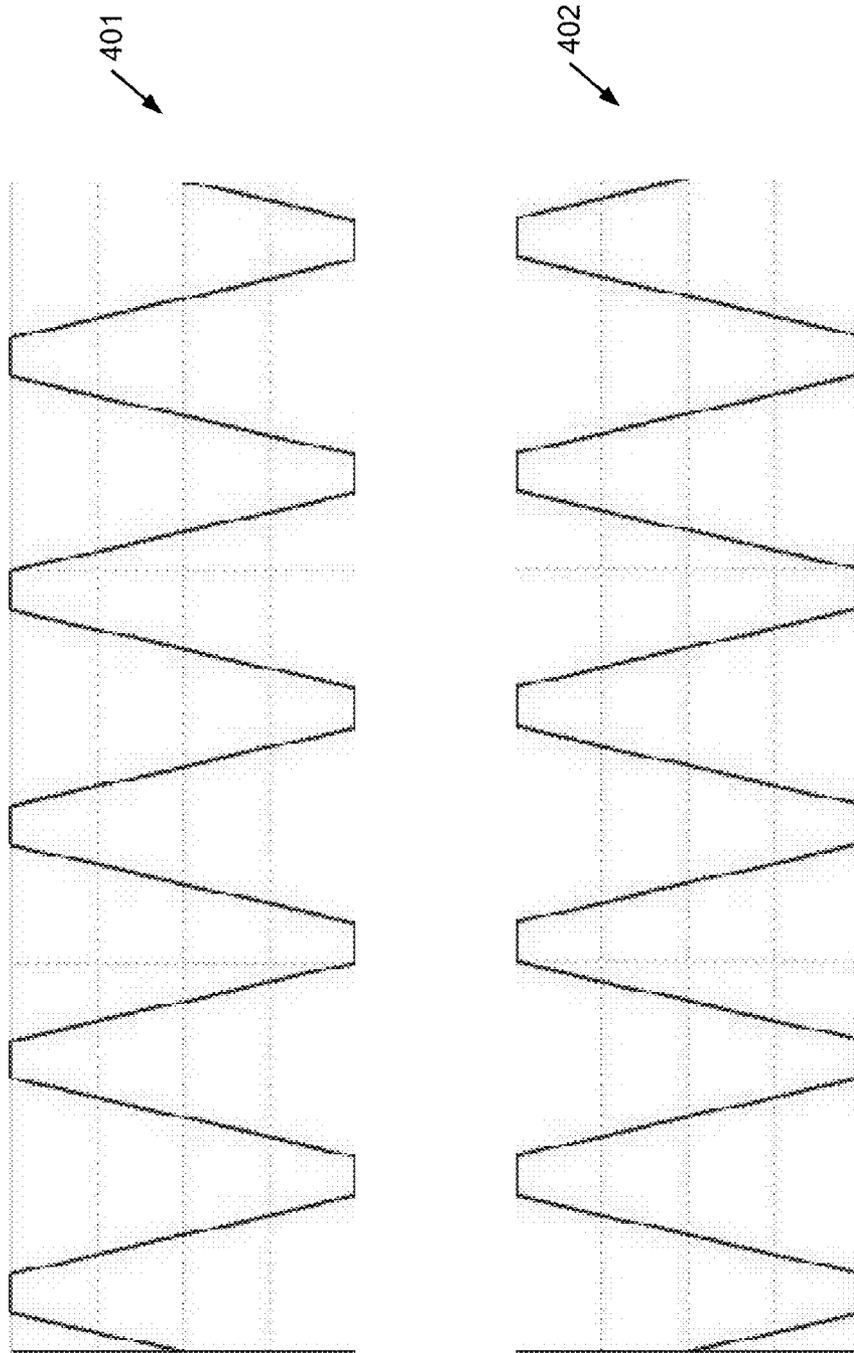
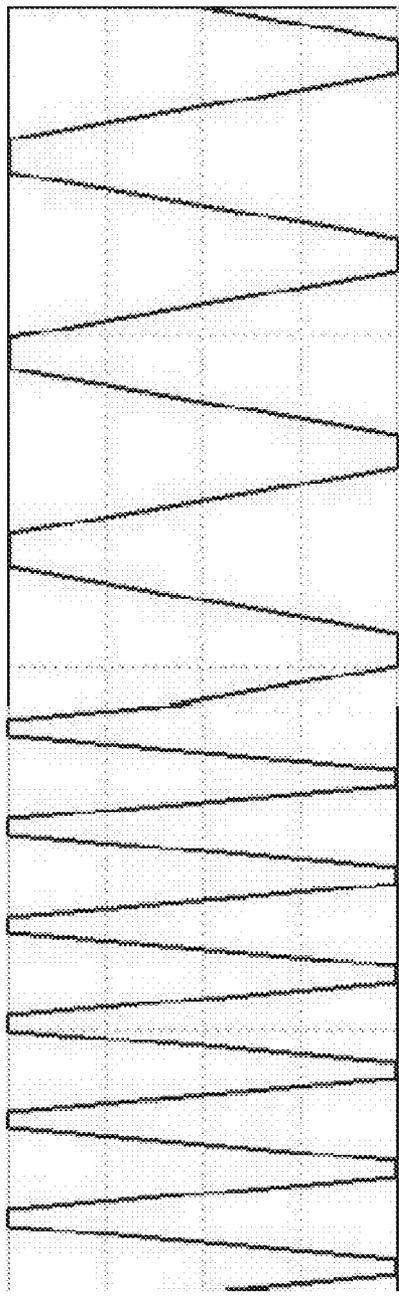
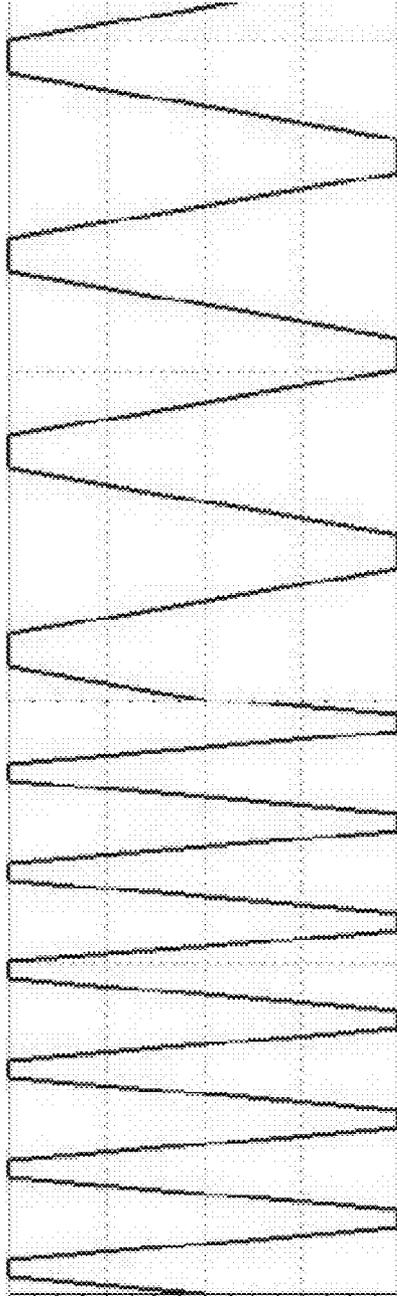


FIG. 4



502

501

FIG. 5

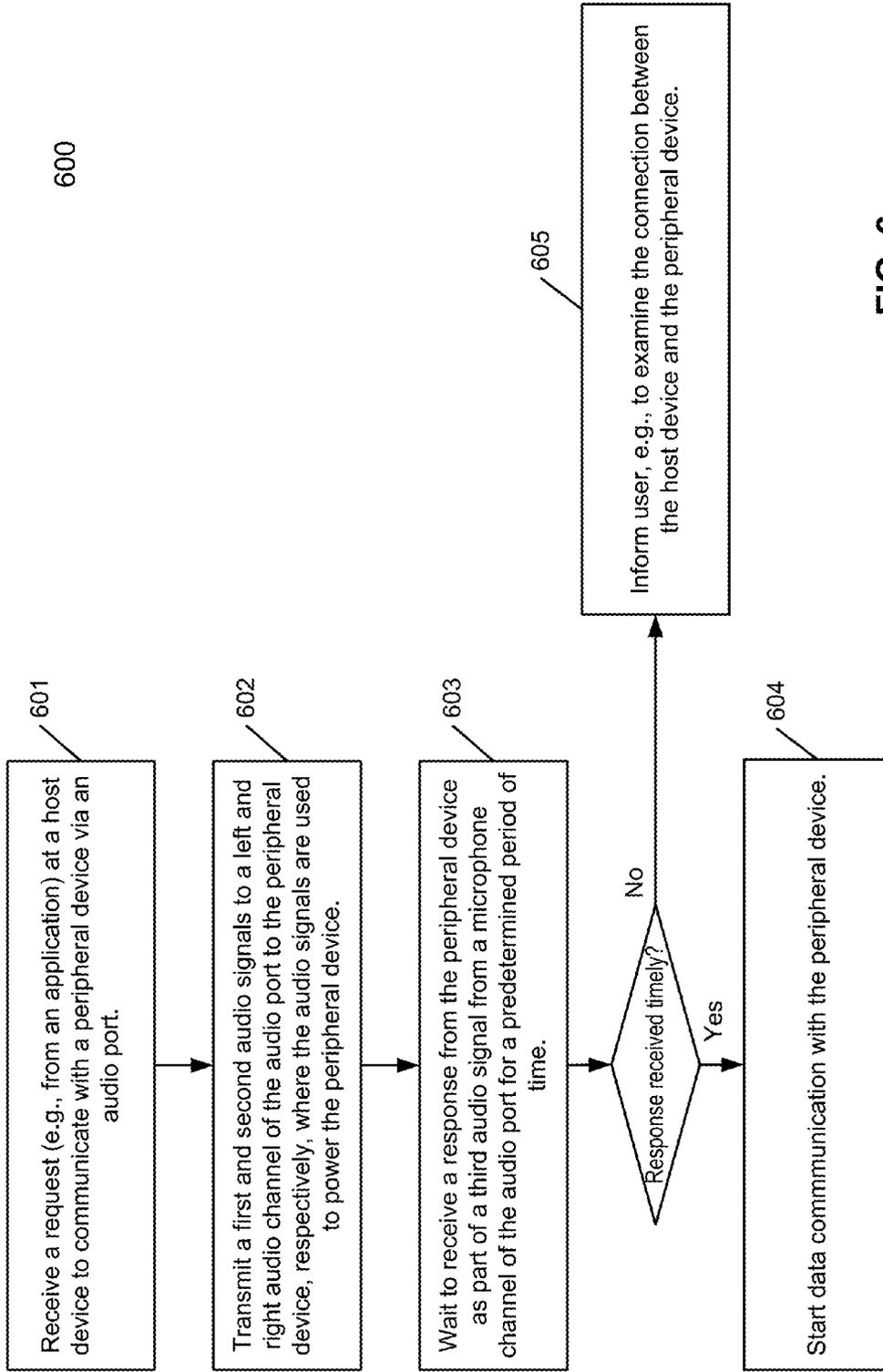


FIG. 6

700

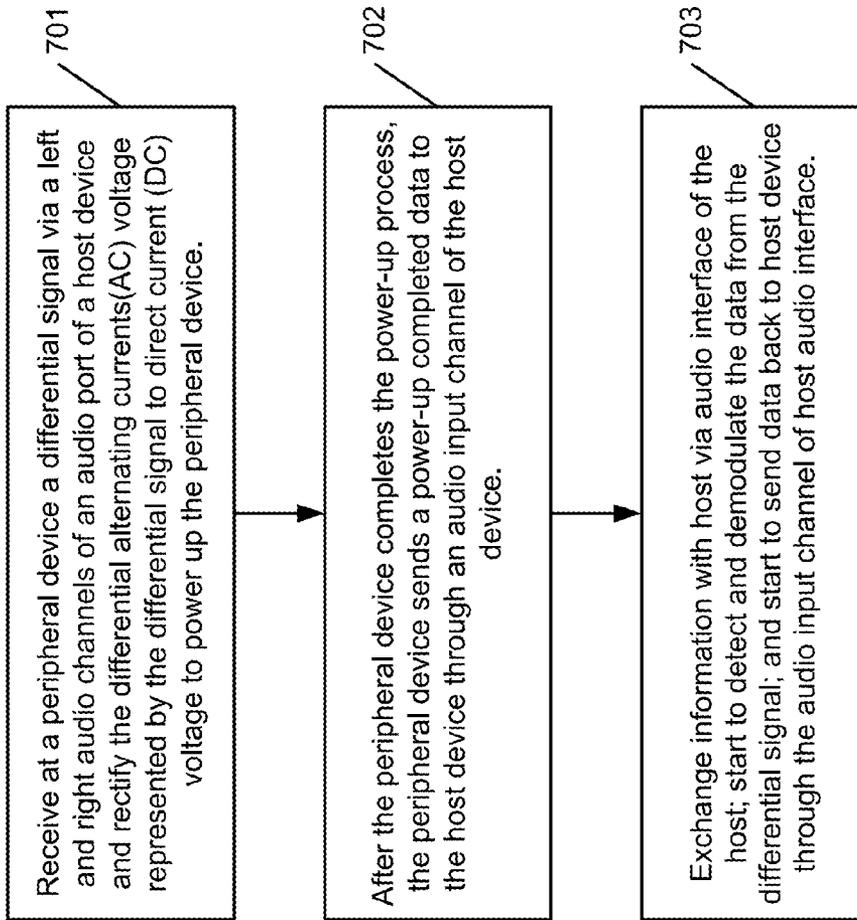


FIG. 7

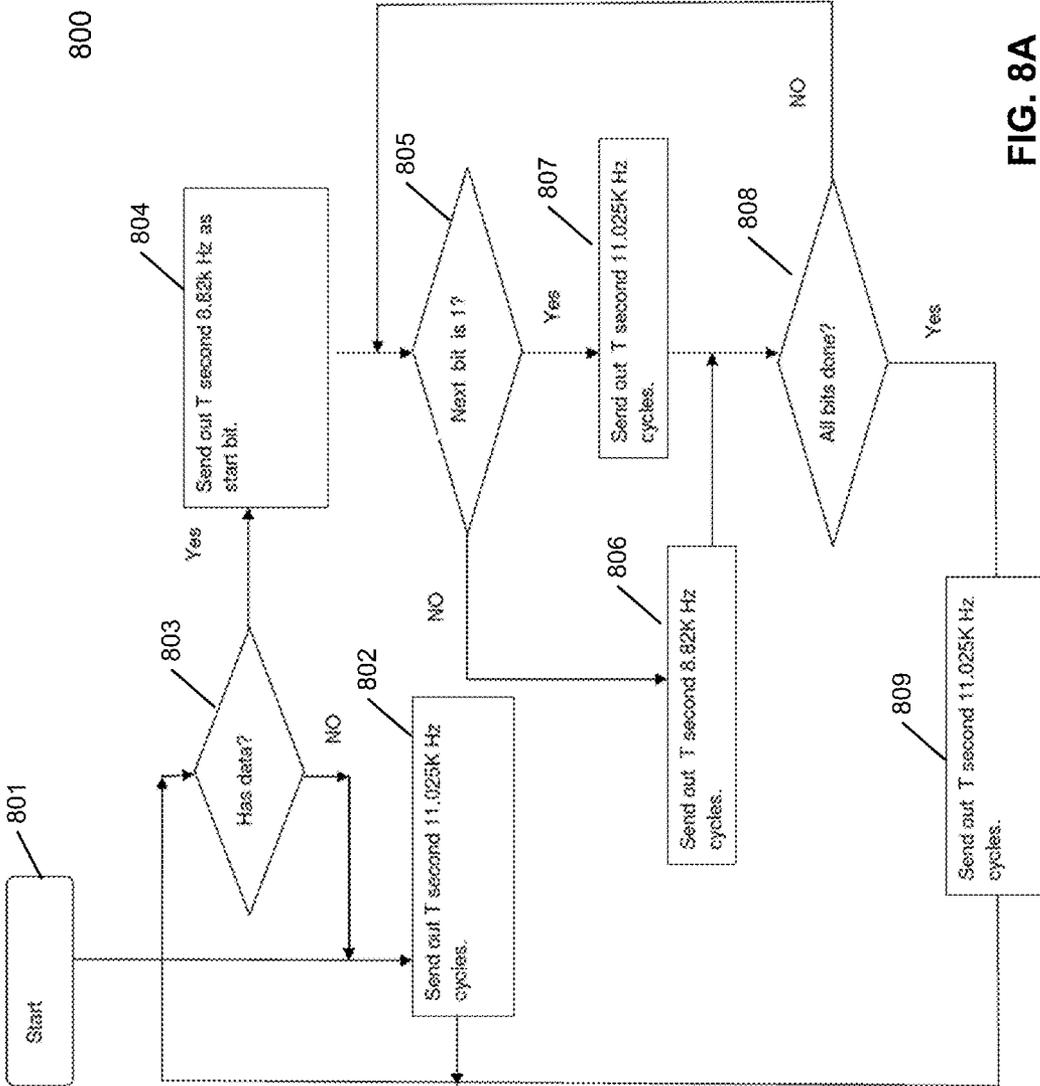


FIG. 8A

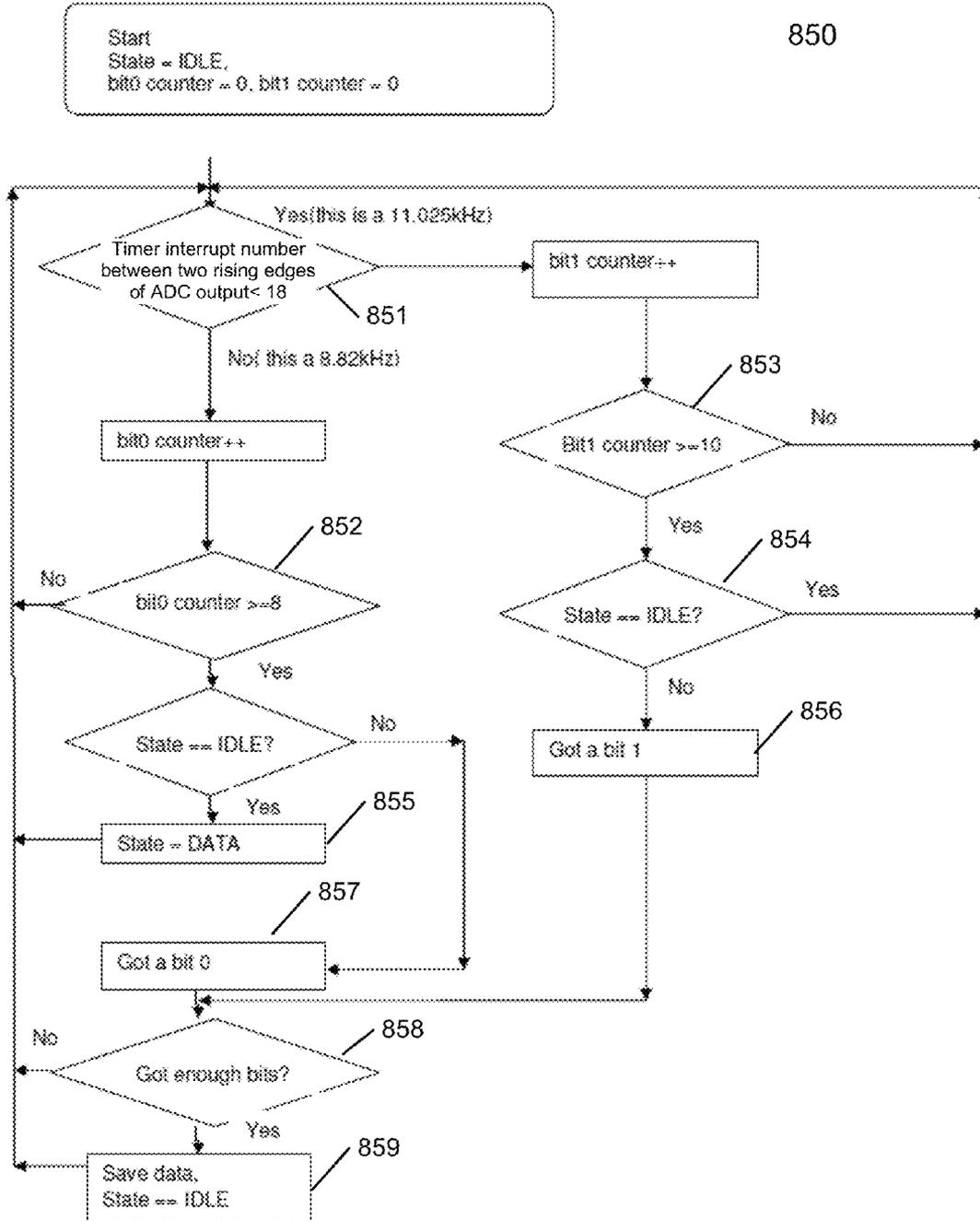


FIG. 8B

**METHOD AND APPARATUS TO USE AUDIO
PORT TO PROVIDE POWER AND DATA
COMMUNICATIONS FOR PERIPHERAL
DEVICES**

RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Patent Application No. 61/692,238, filed Aug. 23, 2012, which is incorporated by reference herein in its entirety.

FIELD OF THE INVENTION

Embodiments of the present invention relate generally to operations of electronic devices. More particularly, embodiments of the invention relate to providing power and communicating between a host device and a peripheral device via an audio interface.

BACKGROUND

An audio port in a personal computer and a mobile electronics device is very rarely used for data communication and power transferring. As a modern electronics device is getting smaller and more powerful, it can be made with more and more functions, due to the portability requirement. Most of them have very limited external interfaces to communicate with a peripheral device. Besides universal serial bus (USB) and wireless interface, an audio port or interface provides a very non-expensive approach for a host device to communicate with a peripheral device.

In the mean time, to simplify the design of both host and peripheral devices and reduce the cost; it is desirable to transfer power from the host device to the peripheral device through the audio interface. Conventional systems use one channel of stereo audio port to provide power to a peripheral device and the other channel used for data communication. Limited to the driving strength of audio port, one audio channel is not sufficient to provide enough power for many applications.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements.

FIG. 1 is a block diagram illustrating a system connection configuration according to one embodiment of the invention.

FIG. 2 is a block diagram illustrating the detail of the connection configuration according to FIG. 1.

FIGS. 3A and 3B are block diagrams illustrating certain AC/DC converters which may be used with an embodiment of the invention.

FIG. 4 shows the complementary audio carrier signals that can be used with an embodiment of the invention.

FIG. 5 shows the complementary audio signals with modulated data according to one embodiment of the invention.

FIG. 6 is a flow diagram illustrating application program starting and running on host device.

FIG. 7 is a flow diagram illustrating power on process of peripheral device and data exchange process of the application program running on microcontroller on peripheral device.

FIGS. 8A and 8B are flow diagrams illustrating data modulating process running on host device and data demodulating process running on peripheral device.

DETAILED DESCRIPTION

Various embodiments and aspects of the inventions will be described with reference to details discussed below, and the accompanying drawings will illustrate the various embodiments. The following description and drawings are illustrative of the invention and are not to be construed as limiting the invention. Numerous specific details are described to provide a thorough understanding of various embodiments of the present invention. However, in certain instances, well-known or conventional details are not described in order to provide a concise discussion of embodiments of the present inventions.

Reference in the specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in conjunction with the embodiment can be included in at least one embodiment of the invention. The appearances of the phrase “in one embodiment” in various places in the specification do not necessarily all refer to the same embodiment.

According to some embodiments, an improved method is provided to use an audio port or interface to connect a host device with a peripheral device. Through this connection the host device delivers power to the peripheral device, and duplex data communication is established between the host device and its peripheral device. Referring to FIG. 1, host device **101** is communicatively coupled to peripheral device **102** via audio interface **103**, which in this example is an audio cable.

Host device **101** can be any of the computing or electronics device, such as, for example, a desktop, a laptop, a tablet, a personal digital assistant (PDA), a gaming device, a media player, a mobile phone, a set top box, or any other devices that can run user application program to do data processing and have a compatible audio interface or port. In one embodiment, audio interface **103** can be any of audio plug or jack that have four contact pins representing, with respect to host device **101**, a left stereo output channel, right stereo output channel, an audio input channel (e.g., microphone), and a ground. In a particular embodiment, audio interface **103** is a 3.5 millimeter (mm) audio plug having four contacts as described above.

The term peripheral device is generally defined as any data acquisition devices such as an electrocardiography (ECG) sensor, a temperature sensor, a humidity sensor; auxiliary devices, such as a mouse, keyboard, etc. that connects to and communicates with the host device through this host device’s audio port.

Unlike a conventional method which uses one of the stereo output channels to deliver power, and uses the other channel to transmit data; in one embodiment of the invention, host device **101** utilizes two audio output channels and audio interface **103** as a differential pair to deliver power to peripheral device **102**. In addition, host device **101** will use a differential audio signal which delivers power to peripheral device **102** as a carrier, modulate data onto this differential audio carrier signal and transmit data to peripheral device **102** via this differential pair. Peripheral device **102** is configured to harvest power from the audio signals received from the differential pair and also to demodulate the data from the audio signals. When peripheral device **102** attempts to respond or communicate data back to host device **101**, peripheral device **102** may transmit the data, which may be encoded to a format can be transferred direct in audio interface or may also be modulated onto a separate audio carrier signal, via an audio input channel of audio interface **103**.

In this example, by driving two stereo output channels with complementary AC signals, host device **101** can deliver as

much as twice power to peripheral device **102** compared to using only one channel by a conventional system. By modulating data onto the differential AC signal, data also transferred to peripheral using same audio port. Assuming the maximum output peak to peak voltage delivered to one audio channel is V_{pp} , if we drive the other channel $-V_{pp}$ (phase inverted), we will obtain $2V_{pp}$ on the peripheral side. Assuming that output impedance of each audio channel of host device **101** equals to Z_o , and input impedance of peripheral device **102** optimized to $2Z_o$, the maximum power received by peripheral device **102** is

$$\left(\frac{2 * V_{pp}}{4 * Z_o}\right)^2 * 2Z_o = \frac{1}{2} * \frac{V_{pp}^2}{Z_o}.$$

Comparing to one channel approach maximum power that can be delivered is

$$\frac{1}{4} * \frac{V_{pp}^2}{Z_o}.$$

This approach doubles the output power. Because both stereo output channels are used to deliver power, the data will be transferred to peripheral device **102** by modulating the differential audio signal as a carrier.

FIG. 2 is a block diagram illustrating a detail configuration according to FIG. 1. Host device **101** may include other hardware components such as one or more processors **201**, memory controller **203**, memory **202**, peripherals interface **204**, audio circuitry **206**, other peripherals may include USB, Ethernet, RF Circuit, Display, Touch panel, Keyboard, Mouse etc. These components may communicate over one or more buses or signal lines. Host device **101** is only one example of the host device; it may have more or fewer components than shown in FIG. 2, may have components combined into one, or may have a different arrangement of the components. All these components and functions may be implemented in hardware, software or a combination of both.

An operating system (e.g., Andriod™, iOS™, Windows™) will run on the host device processor(s), includes various software components and/or drivers to make all components work.

A software application (not shown) may be executed from memory **202** by processor(s) **201** which will request services from the peripheral device **102** through the audio port. For example, host device **101** may be a mobile device having a mobile application running therein to communicate with peripheral device **102** via the audio interface. In one embodiment, when the application is launched within the host device, such as, a mobile device, the application attempts to access an audio stack or framework of an operating system of the host device. For example, the application may attempt to drive predetermined audio data or waveforms to the audio interface via the audio framework provided by the operating system.

According to one embodiment, when an application is activated or executed by processor **201** from memory **202**, the application communicates with an operating system (e.g., Andriod™, iOS™, Windows™) to drive audio data onto audio output channels **211-212**. The application may communicate with an audio subsystem of the operating system, for example, via an application programming interface (API) provided by the operating system. The audio data is handled by an audio driver and converted to audio signals by the audio circuitry **206**. The audio signals may be delivered via paths

221-222 and received by peripheral device **102**. The audio signals transmitted on paths **221-222** are complementary signals to form differential audio signals. The differential audio signals are received by voltage converter **261**. Voltage converter **261** is to convert the signals received from host device **101** to a predetermined range optimized for circuit in power harvest module **262**. One embodiment of voltage converter **261** may be a transformer, if the power harvest circuit can work with differential audio signal voltage from host device directly, then voltage converter **261** can be just two wires pass through the original signal. The output of voltage converter is connected to a power harvest module **262** and a downstream data path to be processed by analog-to-digital (A/D) converter **263**.

In one embodiment, power harvest module **262** is to harvest energy from the differential audio signals and convert the energy into DC voltage to power up other components, such as microcontroller and user application circuit of peripheral device **102**. In one embodiment, power harvest module **262** may include a variety of rectifier, such as, for example, circuits as shown in FIG. 3A or 3B. Referring to FIG. 3A, the circuit includes a rectifying bridge having four metal-oxide-semiconductor field-effect transistors (MOSFETs) to convert the AC signals into DC voltage. The output of the rectifying bridge is coupled to a diode to prevent the current from going backward. The capacitor serves a low-pass filter to filter certain high frequency noise. In one embodiment, the circuit further includes a voltage regulator to regulate the output DC voltage. Voltage regulator can be linear regulator or switching regulator. Alternative implementation can use other rectifier as shown in FIG. 3B, FIGS. 3A and 3B are just two examples of the rectifier circuits, the rectifier has many varieties, for example half-wave rectifier and voltage-multiplying rectifier are not shown.

Referring back to FIG. 2, in one embodiment, microcontroller **264** and the program running on it will manage the operation of peripheral device. Microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Microcontroller may integrate other function like ADC, DAC, Comparator, timer, etc. Program running on the microcontroller will: 1) receive information from host device (explained in FIG. 8B); 2) control the user application circuit (e.g. temperature sensors), and exchange information with it; and 3) convert data to a format that can be transmitted in audio signal and transmit to host device through the microphone input port of host device.

Referring to FIG. 2, A/D converter **263** includes a comparator to convert differential analog audio signal (e.g., audio signal with modulated data) to a two-level digital signal. The digital signals are used by microcontroller **264** for demodulating and processing. In one embodiment, as shown in FIG. 8B, a program running on the microcontroller **264** demodulates and decodes the raw data from the comparator and sends the result data to other processing logic (for example, user application circuit **266**) as an intended recipient of the data. In one embodiment, this comparator is inside the microcontroller.

Referring to FIG. 2, when microcontroller **264** transfers data to host device **101** (through voltage converter **265**), a program running on microcontroller will convert the data to a format that can be transferred through an audio channel. One embodiment is to modulate a signal to a carrier, for example using Bell **202** signaling method. Another embodiment is to use line code technology to eliminate the DC component of the raw data and transfer digital signal in base band. Both

methods will be referred as a modulate method in this data transmission from a peripheral device to a host device.

Voltage converter **265** is used to convert the voltage level of output of microcontroller **264** to a proper range required by host device audio input channel. In one embodiment voltage converter **265** is a voltage divider consists of two resistors.

FIG. **4** shows the complementary audio carrier signals that are transmitted from host device **101** to peripheral device **102** via the audio output channels **211-212**. In this example, referring to FIG. **4**, waveform **401** represents a waveform on the left output channel and waveform **402** represents a waveform on the right output channel.

The data may be modulated onto the audio carrier signals via a phase-shift keying (PSK) method or a frequency-shift keying (FSK) modulation method. FIG. **5** shows the complementary audio signals with FSK modulation according to one embodiment, **501** is binary one data, which is represented by a first predetermined number of consecutive cycles (e.g., 10 cycles) of a first predetermined frequency (e.g., 11.025 kilohertz or KHz). **502** is binary zero data which is represented by a second predetermined number of consecutive cycles (e.g., 8 cycles) of a second predetermined frequency (e.g., 8.82 KHz).

Referring back to FIG. **2**, according to one embodiment, microcontroller **264** is programmed to set timer interrupt clock cycle to $T_{int}=1/(44.1 \text{ k} * 4)$ seconds. The number of this timer interrupt will be counted between two rising edges of the A/D converter **263** output. Each 11.025K signal cycle will have 16 T_{int} clock count and each 8.82 KHz signal cycle will have 20 T_{int} clock count. If **18** is selected as a threshold and the bit rate is 1.1025 KHz baud, each bit will take 907.029 micro second (us). Every bit **0** will last eight 8.82 k cycles and every bit **1** will last ten 11.025 KHz cycles. Microcontroller **264** will count the T_{int} cycles between two rising edge of comparator output. If the count is smaller than 18, then this is an 11.025 KHz signal, otherwise it is an 8.82 KHz signal.

FIG. **6** is a flow diagram illustrating how host device power up the peripheral device and establish data host device power up the peripheral device and establish data host communication with peripheral device. At block **601**, user starts an application on a host device, such as host device **101** of FIG. **2**, requesting a service from a peripheral device. This application starts a process to power up and to establish communication with a peripheral device via an audio interface.

At block **602**, the host device's processing logic (e.g., application and/or operating system) transmits a differential audio signal on the audio interface (e.g., audio socket or jack) of the host device, which is communicatively coupled to and received by the peripheral device. The transmitted differential audio signals are used to provide power to the peripheral device. At block **603**, host device's processing logic waits for a response from the peripheral device via the audio interface of the host device input channel (e.g., microphone) for a predetermined time period. If the host device application receives a response from the peripheral device which confirms the peripheral power on process completes successfully, and then host device application starts a process at block **604**, otherwise, the application will inform a user to check connection of the peripheral device and keeps executing block **603** until a confirmation is received. At block **604**, processing logic starts duplex data communication by encoding and modulating data onto the differential audio signals (as carrier signals) on output channels to be sent to the peripheral device. At the same time, the application will receive data from the input audio channel.

FIG. **7** is a flow diagram illustrating how a peripheral device receives power from a host device and establishes communication with the host device, according to one embodiment of the invention. Referring to FIG. **7**, at block **701**, a peripheral device receives a differential signal via an audio cable which is communicatively coupled to an audio interface of a host device. The peripheral device is powered up by harvesting energy from a differential signal received. The energy harvesting process is composed of all or part of 3 hardware function blocks, including 1) converting the differential signal to a predetermined voltage; 2) rectifying the signal to a DC voltage; and 3) regulating the DC voltage to a predetermined stable voltage. At block **702**, after receiving stable power supply, microcontroller completes initialization process and sends power-up complete information to the host device through an audio input channel of the host device. At block **703**, the peripheral device's processing logic starts receiving information from the host device, controls user application circuit, and sends status and data back to the host device.

FIG. **8A** is a flow diagram illustrating a method of modulating data to two frequencies according to one embodiment of the invention. Method **800** may be performed by processing logic, which may include software, hardware, or a combination thereof. For example, method **800** may be performed by host device **101** of FIG. **2**. Referring to FIG. **8A**, at block **801**, an application is launched at a host device (e.g., by a user), and this application start transfer power and data. At block **802**, the application starts driving a differential signal at an audio interface of the host device. In one embodiment, the differential signal has a first predetermined frequency of 11.025 KHz, if the application does not have any data to be sent to the peripheral device, the application may keep driving the same signal having the first predetermined frequency, we call this is IDLE state.

At block **803**, when it is determined that the application has data to be sent to the peripheral device, the application drives a second predetermined frequency and number of cycles of pulses onto the audio carrier signal at block **804**. In one embodiment, the application drives T seconds (e.g., $\frac{1}{1102.5}$ seconds or 907.029 microseconds or μs) of a differential signal at a second predetermined frequency. The second predetermined frequency may be 8.82 KHz. The first T seconds of the second predetermined frequency after IDLE state may be used as a start bit for receiver to synchronize with the transmitter. After the start bit, at block **805**, application will send out data bit, including parity check bit or cyclic redundancy check (CRC) bits depend on the communication protocol (e.g., universal asynchronous receiver/transmitter or UART protocol). In one embodiment, for each bit having a logical value of one, processing logic sends out T seconds of pulses having the first predetermined frequency at block **807**. For each bit having a logical value of zero, processing logic will send out T seconds of pulses having the second predetermined frequency at block **806**. After all data bits are completed, processing logic completes the transaction with T seconds of pulses having the first predetermined frequency at block **809**, which serve as a stop bit and keep sending until next data are ready.

The following table is an example sending a byte **5A** at 1102.5 baud rate, it sends out 907.029 us of 8.82K signal as start bit. Then it sends 907.029 μs of 11.025K as binary 1, and so on until all data bits and stop bit are sent and return to idle state.

Bus state	Code name	Interval (μs)	Frequency (Hz)	Cycles
Idle	1	907.029	11025	10
Start bit	0	907.029	8820	8
D7	0	907.029	8820	8
D6	1	907.029	11025	10
D5	0	907.029	8820	8
D4	1	907.029	11025	10
D3	1	907.029	11025	10
D2	0	907.029	8820	8
D1	1	907.029	11025	10
D0	0	907.029	8820	8
Parity bit	0 (even)	907.029	8820	8
Stop	1	907.029	11025	10
Idle	1	907.029	11025	10

FIG. 8B is a flow diagram illustrating a method of demodulating data from 2FSK signal. Method 850 may be performed by processing logic, which may include software, hardware, or a combination thereof. For example, method 850 may be performed by peripheral device 102 of FIG. 2. In one embodiment, a microcontroller of a peripheral device (e.g., microcontroller 264 of FIG. 2) may be programmed to configure a timer interrupt clock cycle to $T_{int}=1/(44.1 \text{ KHz} * 4)$ seconds. The number of this timer interrupt between two rising edges of A/D converter 263 output may be counted by a counter. In one embodiment, each 11.025 KHz signal cycle will have 16 T_{int} clock counts and each 8.82 KHz signal cycle will have 20 T_{int} clock counts.

In one embodiment, at block 851, a predetermined number of cycles, in this example, 18, is selected as a threshold to determine types of signals received. Since the bit rate is 1.1025 KHz baud, each bit will take 907.029 μs. Every bit 0 will last eight 8.82 KHz cycles (block 852) and every bit 1 will last ten 11.025 KHz cycles (block 853). After application starts, the host will keep sending the 11.025 KHz signal to the peripheral device, which represents an IDLE state (block 854). The microcontroller will count the T_{int} cycles between two rising edges of comparator output. If the count is smaller than 18, then the signal is an 11.025 KHz signal; otherwise it is 8.82 KHz signal cycle.

After the peripheral receives first 8 consecutive 8.82 KHz signal cycles, which represents a start bit in many communication protocols, the peripheral device enters a DATA state (block 855). In the DATA state, for 10 consecutive 11.025 KHz cycles, a bit 1 is received (block 856). For 8 consecutive 8.82 KHz cycles, a bit 0 is received (block 857). The microcontroller keeps receiving data until all data bits in one transfer session are received (block 858). After one transfer session is done, the host will keep sending 11.025 KHz signal cycles, which put the transfer in an IDLE state (block 859).

Note that while FIG. 2 illustrates various components of a data processing system, it is not intended to represent any particular architecture or manner of interconnecting the components; as such details are not germane to embodiments of the present invention. It will also be appreciated that network computers, handheld computers, mobile phones, and other data processing systems which have fewer components or perhaps more components may also be used with embodiments of the invention.

Some portions of the preceding detailed descriptions have been presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the ways used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally,

conceived to be a self-consistent sequence of operations leading to a desired result. The operations are those requiring physical manipulations of physical quantities.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the above discussion, it is appreciated that throughout the description, discussions utilizing terms such as those set forth in the claims below, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

The techniques shown in the figures can be implemented using code and data stored and executed on one or more electronic devices. Such electronic devices store and communicate (internally and/or with other electronic devices over a network) code and data using computer-readable media, such as non-transitory computer-readable storage media (e.g., magnetic disks; optical disks; random access memory; read only memory; flash memory devices; phase-change memory) and transitory computer-readable transmission media (e.g., electrical, optical, acoustical or other form of propagated signals—such as carrier waves, infrared signals, digital signals).

The processes or methods depicted in the preceding figures may be performed by processing logic that comprises hardware (e.g. circuitry, dedicated logic, etc.), firmware, software (e.g., embodied on a non-transitory computer readable medium), or a combination of both. Although the processes or methods are described above in terms of some sequential operations, it should be appreciated that some of the operations described may be performed in a different order. Moreover, some operations may be performed in parallel rather than sequentially.

In the foregoing specification, embodiments of the invention have been described with reference to specific exemplary embodiments thereof. It will be evident that various modifications may be made thereto without departing from the broader spirit and scope of the invention as set forth in the following claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

What is claimed is:

1. A computer-implemented method, comprising:

activating a first application at a host device that requires a service from a peripheral device, wherein the first application drives a first audio signal into a first stereo output channel of an audio interface circuit of the host device and drives a second audio signal into a second stereo output channel of the audio interface circuit, and wherein the first audio signal and the second audio signal are complementary signals;

harvesting energy by a power harvest module of the peripheral device from both the first audio signal and the second audio signal to generate a direct current (DC) voltage to power up one or more hardware components of the peripheral device including a microcontroller of the peripheral device;

powering up and completing an initializing process at the peripheral device, wherein the microcontroller on the peripheral device generates peripheral device ready data, modulates the peripheral device ready data to a

9

third signal, and transmits the third signal to an input channel of the audio interface circuit of the host device; confirming by the first application of the host device readiness of the peripheral device by receiving the third signal from the input channel of the audio interface circuit of the host device and demodulating the peripheral device ready data sent from the third signal received from the peripheral device;

establishing by the first application of the host device duplex data communication with the peripheral device, by transferring data through the first and the second audio output channels of the audio interface circuit, and receiving data from the input channel of the audio interface circuit; and

continuously receiving power by the peripheral device by harvesting energy from the first and the second audio channels of the audio interface circuit of the host device, while demodulating the data from the first and the second audio channels, wherein the peripheral device receives data sent from the host device, and wherein the peripheral device modulates data received from a second application on the peripheral device, and transmits the data to the host device via the audio input channel of the audio interface circuit of the host device.

2. The method of claim 1, wherein the audio interface circuit of the host device includes a first stereo output channel, a second stereo output channel, an audio input channel, and a ground pin.

3. The method of claim 2, wherein the first stereo output channel is a left stereo output channel, and wherein the second stereo output channel is a right stereo output channel.

4. The method of claim 1, wherein harvesting energy from both the first audio signal and the second audio signal comprises rectifying the first and the second audio signals as differential alternating current (AC) signals into a DC voltage to generate the DC power.

5. The method of claim 1, wherein peripheral device is configured to performing the following operations:

detecting a fourth signal that has been modulated onto the first audio signal and the second audio signal;

extracting the fourth signal from the first audio signal and the second audio signal; and

processing the extracted fourth signal as part of data communicated from the host device to the peripheral device, including delivering the extracted data to the second application running within the peripheral device as an intended recipient.

6. The method of claim 5, wherein detecting and extracting the fourth signal comprise:

recognizing a first portion of the fourth signal that has been modulated in a first predetermined frequency;

translating the first portion into a first binary data value;

recognizing a second portion of the fourth signal that has been modulated in a second predetermined frequency; and

translating the second portion into a second binary data value.

7. The method of claim 6, wherein recognizing a first portion of the fourth signal comprises counting a number of consecutive pulses of the first predetermined frequency that is above a first predetermined threshold.

8. The method of claim 7, wherein recognizing a second portion of the fourth signal comprises counting a number of consecutive pulses of the second predetermined frequency that is above a second predetermined threshold.

10

9. A peripheral device, comprising:

a microcontroller;

an interface circuit capable of communicatively coupling to a host device via an audio interface of the host device, the audio interface having a first stereo output channel, a second stereo output channel, an audio input channel, and a ground pin, wherein the interface circuit is to receive from the host device a first audio signal via the first stereo output channel and a second audio signal via the second stereo output channel;

an energy harvesting circuit coupled to the interface circuit to harvest energy from both the first audio signal and the second audio signal to generate a direct current (DC) voltage to power up the peripheral device; and

a program executed by the microcontroller to receive the first audio signal and the second audio signal from the interface circuit and to demodulate data from the first and the second audio signals, wherein the data is transmitted from an application running within the host device.

10. The device of claim 9, wherein the first stereo output channel is a left stereo output channel, and the second stereo output channel is a right stereo output channel.

11. The device of claim 10, wherein the program is to receive data from application circuit and peripheral device status, modulate the data into a third audio signal, and transmit the third audio signal to the host device via the audio input channel of the audio interface.

12. The device of claim 11, wherein the energy harvesting circuit is to rectify the first and the second audio signals as differential alternating current (AC) signals into a DC voltage to generate the DC power.

13. The device of claim 12, wherein the first audio signal and the second audio signal are complementary signals.

14. The device of claim 10, wherein the program is configured to

detect a fourth signal that has been modulated onto the first and the second audio signals,

extract the fourth signal from the first and the second audio signals, and

process the extracted fourth signal as part of data communicated from the host device to the peripheral device, including delivering extracted data to an application as an intended recipient.

15. The device of claim 14, wherein the microcontroller includes a comparator to convert complementary first audio signal and second audio signal to a binary signal.

16. The device of claim 15, wherein the microcontroller is further configured to

recognize a first portion of the fourth signal that has been modulated in a first predetermined frequency,

translate the first portion into a first binary data value,

recognize a second portion of the fourth signal that has been modulated in a second predetermined frequency, and

translate the second portion into a second binary data value.

17. The device of claim 16, wherein recognizing a first portion of the fourth signal comprises counting a number of consecutive pulses of the first predetermined frequency that is above a first predetermined threshold.

18. The device of claim 16, wherein recognizing a second portion of the fourth signal comprises counting a number of consecutive pulses of the second predetermined frequency that is above a second predetermined threshold.

11

19. A computer-implemented method, comprising:
 receiving, at a peripheral device communicatively coupled
 to a host device via an audio interface of the host device,
 the audio interface having a first stereo output channel, a
 second stereo output channel, an audio input channel, and a ground pin, a first audio signal via the first stereo
 output channel and a second audio signal via the second
 stereo output channel;
 harvesting energy from both the first audio signal and the
 second audio signal to generate direct current (DC)
 power to power one or more hardware components of
 the peripheral device including a processor of the
 peripheral device; and
 receiving data from the first stereo output channel and the
 second stereo output channel by demodulating data
 which has been modulated on the first and the second
 audio signals transmitted via the first and the second
 stereo output channels by host device, wherein the

12

modulated data on the first and the second stereo output
 channels are complementary signals.
 20. The method of claim 19, wherein the first stereo output
 channel is a left stereo output channel, and wherein the sec-
 ond stereo output channel is a right stereo output channel.
 21. The method of claim 19, further comprising:
 receiving data from an application running within the
 peripheral device;
 modulating the data into a third audio signal; and
 transmitting the third audio signal with the modulated data
 from the peripheral device to the host device via the
 audio input channel of the audio interface.
 22. The method of claim 19, wherein harvesting energy
 from both the first audio signal and the second audio signal
 comprises rectifying the first and the second audio signals as
 differential alternating current (AC) signals into a DC Voltage
 to generate the DC power.

* * * * *