

AUDIO RECORD AND PLAYBACK THROUGH A STANDARD
TELEPHONE IN A COMPUTER SYSTEM

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION:

5 The present invention pertains to the field of personal computer systems. More particularly, this invention relates to a multifunction input/output subsystem for a personal computer that enables unrestricted sound file input and output through a telephone set.

10 2. ART BACKGROUND:

Non dedicated telephone lines are often employed in small office or home office environments to provide modem and fax communication lines for a personal computer. The personal computer typically employs a non
15 dedicated telephone line for sharing modem and fax calls as well as voice calls. Such telephone line sharing enables modem and fax communication while avoiding the cost of providing separate modem telephone lines and fax telephone lines in the small business or home office.

20 In such small business or home office environments, a telephony subsystem of the personal computer typically performs the telephone line interface functions required for modem, fax and voice communication. The telephone line interface functions include line seizure and ring detect, as well as audio and digital signal processing functions.

Typically, application programs executing on the personal computer perform the higher level modem, fax and voice communication functions as well as user interface functions.

5 In a typical small business or home office telephone line topology, the telephony subsystem is coupled to an incoming telephone line or a private branch exchange (PBX) line. The telephony subsystem usually drives one or more serial telephones over a local line.

10 Typically in such systems, the telephony subsystem transfers incoming voice calls on the telephone line to the serial telephone over the local line to enable a user to receive the voice call. The telephony subsystem usually employs a switching device such as a relay to couple the telephone line to the local line and serial telephone.

15 Unfortunately in such prior systems, the telephony subsystem creates a physical connection through the relay between the telephone line and the local line to transfer incoming voice calls to the serial telephone. With such a physical connection, the originator of the incoming voice call hears a busy
20 signal over the telephone line if the serial telephone is off-hook during the incoming call. As a consequence, the serial telephone must remain on-hook to receive incoming voice calls, thereby precluding the use of the serial telephone for other input/output functions.

SUMMARY AND OBJECTS OF THE INVENTION

One object of the present invention is to employ a standard telephone for audio input and output functions in a computer system.

5

Another object of the present invention is to employ the telephone for audio input and output functions in the computer system over a local line while enabling reception of incoming calls over a telephone line.

10

A further object of the present invention is to provide a software architecture for a multifunction I/O subsystem that decouples processing of incoming calls over the telephone line from the processing of telephony signals to and from the telephone.

15

A further object of the present invention is to enable audio input and output functions through a standard analog or digital telephone by providing a standard data format for data streams transferred between driver programs in the multifunction I/O subsystem, such that the standard data format enables telephony driver programs and audio driver programs to produce

20

and consume sound data streams for audio tasks.

25

Another object of the present invention is to employ a digital telephone for audio input and output functions in the computer system by utilizing stereo-to-mono conversion and sample-rate conversion functions which operate on the standard data format.

Another object of the present invention is to enable recording of sound files for the computer system through the standard telephone, and to enable playback of sound files through the standard telephone.

5 These and other objects of the invention are provided by a method for performing audio record and playback functions through a telephone in a computer system. A user selects a normal audio mode or a telephone audio mode for the computer system through a user interface of a host audio application program executing on a host processor of the computer system. A
10 standard data format for sound data streams enables a telephony driver and an audio driver executing on a digital signal processing subsystem of the computer system to generate and consume sound samples. The telephony driver samples and transmits audio signals on a local line coupled to the telephone. The telephony driver converts the sound samples to and from a
15 data stream having a standard data format.

During audio recording, a telephony driver program generates a set of sound samples by sampling audio signals over a local line coupled to a telephone, and converts the sound samples into a sound data stream having
20 a standard data format if the telephone audio mode is selected.

An audio driver generates the sound samples by sampling audio signals over an audio line, and converting the sound samples into the sound data stream having the standard data format if the normal audio mode is
25 selected.

The sound data stream is then transferred to an audio task executing on a digital signal processing subsystem in the computer system. The audio task converts the sound data stream into a format required by a host audio application program executing on a host processor in the computer system,
5 and then transfers the sound data stream to the host audio application program.

During audio playback, the audio task receives a sound data stream from the host audio application program and converts the sound data stream
10 from a data format required by the host audio application program into the standard data format. The audio task then transfers the sound data stream having the standard data format to the audio driver if the normal audio mode is selected. The audio driver converts the sound data stream into a set of sound data values and transfers the sound data values to a stereo audio
15 codec circuit for sound generation over an audio signal line.

The audio task transfers the sound data stream to the telephony driver if the telephone audio mode is selected. The telephony driver converts the sound data stream into a set of telephony data values and transfers the
20 telephony data values to a telephony circuit, such that the telephony circuit generates telephony audio signals corresponding to the telephony data values and transfers the telephony audio signals to a telephone over a local line.

Other objects, features and advantages of the present invention will be
25 apparent from the accompanying drawings, and from the detailed description that follows below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like
5 references indicate similar elements, and in which:

Figure 1 illustrates a telephone system for one embodiment which includes a personal computer that receives incoming calls and transmits outgoing calls over a telephone line;

10

Figure 2 illustrates the personal computer which comprises a host processor, a memory subsystem, a mass storage subsystem, and a multifunction I/O subsystem;

15 Figure 3 is a diagram illustrating the multifunction I/O subsystem which comprises an analog phone line interface circuit and associated codec circuits, a digital signal processor, an stereo audio codec circuit, and a set of interface circuits;

20 Figure 4 illustrates the software architecture for the host processor and the multifunction I/O subsystem;

Figure 5 illustrates interaction between a host device driver executing on the host processor and a corresponding DSP-OS task executing on the
25 multifunction I/O subsystem;

Figure 6 illustrates the software architecture of the host processor for one embodiment;

Figure 7 illustrates the software architecture of the multifunction I/O
5 subsystem for one embodiment.

DETAILED DESCRIPTION

Figure 1 illustrates a computer system 10 for one embodiment. The computer system 10 includes a personal computer 12 that receives incoming calls and transmits outgoing calls over a telephone line 22. The telephone line 22 comprises a standard telephone line or alternatively a private branch exchange (PBX) line or ISDN connection.

A pair of telephones 16 and 17, along with a telephone answering machine 18 are coupled to the telephone line 22 in parallel with the personal computer 12. A serial telephone 15 is coupled to the personal computer 12 over a local line 20.

Figure 2 illustrates the personal computer 12. The personal computer 12 comprises a host processor 32, a memory subsystem 31, a mass storage subsystem 33, and a multifunction I/O subsystem 14. The host processor 32, the memory subsystem 31, and the multifunction I/O subsystem 14 communicate over a host bus 44.

The multifunction I/O subsystem 14 provides sound capture/playback functions, fax and data modem functions, and telephone control functions. The multifunction I/O subsystem 14 provides connections for microphones and speakers, and hardware to support telephone and telephone line connections.

The mass storage subsystem 33 represents a wide variety of mass storage devices for large scale data and program storage, including magnetic disk drives and CD ROM drives.

5 **Figure 3 is a diagram illustrating the multifunction I/O subsystem 14.**
The multifunction I/O subsystem 14 comprises an analog phone line interface circuit 60 and associated codec circuits (a μ -law codec 70 and a linear codec 72), a digital signal processor (DSP) 61, an stereo audio codec circuit 54, and a set of interface circuits 63-69.

10

A memory block 62 comprises a DSP static random access memory (SRAM) and a DSP dynamic random access memory (DRAM). Both DRAM and SRAM can be used for program or data.

15 The bus interface circuit 67 and the DMA arbiter circuit 65 enable the DSP 61 to communicate over the host bus 44.

20 The analog phone line interface circuit 60 and the μ -law and linear codec circuits 70 and 72 enable transfer of incoming and outgoing calls over the local line 20 and the telephone line 22 including full duplex connections between the μ -law codec 70 and the telephone 15 and full duplex connections between linear codec 72 and the telephone line 22. The μ -law phone and linear line codec circuits 70 and 72 transfer digital data to and from the DSP 61 through the phone interface circuit 66.

25

The stereo audio codec circuit 54 transfers audio signals (mono and stereo) to the external speakers 18 and the headphone set 19 over a set of

audio signal lines 56. The stereo audio codec circuit 54 also receives audio signals (mono and stereo) from an audio source and external microphone 14 over a set of audio signal lines 58. The stereo audio codec circuit 54 performs analog-to-digital conversion on the received audio signals, and transfers the
5 digitized audio data to the audio interface (I/F) circuit 68. The stereo audio codec circuit 54 performs digital-to-analog conversion on audio data received through the audio interface (I/F) circuit 68.

For one embodiment, the multifunction I/O subsystem 14 comprises
10 an industry standard architecture (ISA) add-in card that provides hardware to support audio and telephony applications.

Figure 4 illustrates the software elements for the host processor 32 and the multifunction I/O subsystem 14. A set of application programs 100 - 102
15 executing on the host processor 32 access the hardware functions of the multifunction I/O subsystem 14 through an appropriate program interface (API) 104 - 106.

An application program interface consists of a dynamic link library
20 (DLL) layer and a corresponding host device driver. The dynamic link library layer interacts with host application programs that use the corresponding application program interface. The dynamic link library layer performs error checking and translates application program calls into messages that are passed to the appropriate device driver.

25

For example, the application program interface 104 provides dynamic link library 109 that links at runtime to a host device driver that

communicates with a hardware device in the multifunction I/O subsystem

14. The dynamic link library 109 translates the application program interface 104 calls into messages that are passed to the device driver which performs the requested function.

5

The dynamic link library layers 107-109 of the application program interfaces 104 - 106 interact with one or more host device drivers 110-114 via common device driver interfaces 115-117. The host device drivers 110-114 are implemented as dynamic link libraries. A set of device-type specific custom
10 messages are defined by the device driver for the corresponding device class.

The code for the application program interfaces 104-106 comprises the dynamic link libraries 107-109. The application programs 100-102 that require the services provided by the application program interfaces 104-106 link
15 against the library corresponding to the corresponding dynamic link library 107-109.

The dynamic link libraries 107-109 translate application program interface calls into host device driver calls or messages. The host device
20 driver calls or messages and associated data structures define the device driver interface (DDI) for the corresponding device class. A host device driver for an application program interface is provided by implementing the corresponding DDI.

25 The multifunction I/O subsystem 14 implements a digital signal processing operating system (DSP-OS) that executes on the DSP 61. The DSP-OS enables multiple concurrent execution of audio and telephony tasks (such

as a DSP-OS task 132 and a DSP-OS task 134) on the multifunction I/O subsystem 14.

5 The DSP-OS is a real time multitasking operating system that provides two programming interfaces: the application programming interface and a system programming interface (SPI). The DSP-OS application program interface provides a set of high-level processor independent functions for computations by the DSP 61, as well as stream input and output to the multifunction I/O subsystem 14.

10

 The software for the multifunction I/O subsystem 14 includes a set of DSP device drivers 130 and 135-137. The DSP-OS tasks 132-134 perform computations and I/O to devices via the DSP device drivers 135-137.

15 A typical DSP-OS task reads input data from a hardware device, processes the input data, and then writes output data to another hardware device. Normally, multiple DSP-OS tasks perform such processes concurrently. A DSP device driver enables the DSP-OS tasks to access hardware devices. Each hardware device corresponds to a DSP device driver
20 that manages the hardware device and provides I/O services to the DSP-OS tasks.

 The DSP-OS system programming interface provides a set of functions for managing DSP-OS tasks, and for managing the DSP memory 62. The DSP-
25 OS system programming interface provides functions for synchronizing DSP-OS tasks, and functions for performing device-independent I/O.

The host device drivers 110 - 114 communicate with DSP programs executing on the multifunction I/O subsystem 14 through a DSP interface 120. For example, a host audio device driver uses the DSP interface 120 to control a DSP-OS task that manages data transfer to and from the stereo audio codec circuit 54. The host audio device driver can communicate with several DSP-OS tasks. A DSP-OS task created by a host device driver may create other DSP-OS tasks to provide the appropriate hardware functions.

The DSP interface 120 is message-based. The DSP interface 120 defines a common structure for messages and a programming model for communication between host device drivers and corresponding DSP-OS tasks. Each host device driver defines an appropriate set of message formats (such as command codes, parameters, status and reply codes, etc.) for communication with the corresponding DSP-OS task.

15

A host DSP driver 122 implements the DSP interface 120. The host DSP driver 122 comprises a dynamic link library (DLL) that enables communication with the multifunction I/O subsystem 14. The host DSP driver 122 loads the DSP-OS and starts the DSP 61 to initialize the multifunction I/O subsystem 14 and run the DSP-OS. After the multifunction I/O subsystem 14 is initialized and the DSP-OS is running, the host DSP driver 122 moves buffers of data and messages between the host processor 32 and the multifunction I/O subsystem 14.

25

The host DSP driver 122 communicates with a DSP host driver 130 executing under the DSP-OS. The host DSP driver 122 and the DSP host driver 130 do not examine the content of the data and messages passed over

the hardware interface 125. The host DSP driver 122 and the DSP host driver 130 synchronize multiple data and message streams and route each stream to the appropriate host device driver and DSP-OS task.

5 A DSP-OS server 124 server executing on the host processor 32 functions as a server for the DSP-OS tasks 132-134 and 146. The DSP-OS server 124 processes requests from the multifunction I/O subsystem 14. The DSP-OS server 124 is a "daemon" on the host processor 32. The DSP-OS server 124 is a part of the host DSP driver 122. The DSP-OS server 124 server
10 services file and console I/O requests from the DSP-OS tasks 132-134 and 146 to the host processor 32. The DSP-OS server 124 assists in dynamic DSP-OS task loading, and provides a user interface for status and performance monitoring of the multifunction I/O subsystem 14.

15 The DSP-OS is a real-time multitasking operating system that provides a kernel and input/output (I/O) environment for the software executing on the multifunction I/O subsystem 14. The DSP-OS defines a set of device-independent I/O interface 140-143 for use by the DSP-OS tasks 132-134 and 146 to access hardware devices of the multifunction I/O subsystem 14. The DSP-
20 OS also defines stackable device drivers which are blocks of code that provide common (i.e., reusable) transformations on I/O streams in the standard format.

 A DSP program executing on the DSP 61 functions as a server for a
25 corresponding client host device driver on the host processor 32. A DSP program performs the commands requested by the client host device driver through the appropriate messages.

A DSP program consists of one or more DSP-OS tasks that move data from source devices to sink devices, while performing specific processing and transformations on the data. For example, an audio DSP-OS task reads audio
5 samples from the stereo audio codec circuit 54 using a device independent I/O interface, processes the samples to perform data compression, and then transfers the compressed data to the host audio device driver executing on the host processor 32.

10 The DSP-OS creates a DSP-OS dispatcher task 146 after starting up. The DSP-OS dispatcher task 146 processes requests from the host device drivers 110 - 114 to create and delete DSP-OS tasks. The DSP-OS dispatcher task 146 matches a task name to the appropriate DSP-OS tasks and then loads and starts the new DSP-OS tasks. The DSP-OS dispatcher task 146 also creates
15 mailboxes for communications between the requesting host device driver or other DSP-OS tasks and the new DSP-OS task.

The functions of a DSP-OS task are controlled by placing messages in a corresponding mailbox. The interpretation of the messages follows a device-
20 class specific protocol. Each DSP-OS task recognizes a small set of basic messages.

Figure 5 illustrates interaction between a host device driver 180 executing on the host processor 32 and a corresponding DSP-OS task 182. The
25 host device driver 180 controls and communicates with the DSP-OS task 182 via a control channel and one or more data channels.

The DSP interface 120 provides the control channel that enables the host device driver 180 to send messages to the DSP-OS task 182, and receive messages from the DSP-OS task 182. The control channel is implemented according to a mailbox facility. The host device driver 180 calls a
5 dspOpenTask routine to create the DSP-OS task 182, and to create a pair of mailboxes 184-185. The host device driver 180 and the DSP-OS task 182 use the mailboxes 184-185 to exchange messages.

Large data transfers between the host device driver 180 and the DSP-OS
10 task 182 are subdivided into blocks according to a buffer size specified for the host device driver 180. The control channel messages are interleaved with data blocks.

The host device driver 180 and the DSP-OS task 182 exchange data
15 through one or more data streams. A data stream is a channel that provides data flow in one direction between a host device driver and a DSP-OS task. Two data channels are employed to exchange data in both directions, i.e. from the host device driver 180 to the DSP-OS task 182 and from the DSP-OS task 182 to the host device driver 180.

20

The DSP interface 120 supplies a uniform set of services for communication with the multifunction I/O subsystem 14. For one embodiment, the DSP interface 120 supplies the communication services for the multifunction I/O subsystem 14 in a Microsoft® Windows environment.
25 The services are employed by the application programs 100-102 and the host device drivers 110-114.

Each host device driver 110-114 starts and stops at least one DSP-OS task, and transfers data streams to and from the DSP-OS task. The DSP interface 120 is optimized for efficient data streaming to the DSP-OS task. The DSP interface 120 enables command message transfer between the host device drivers 110-114 and the DSP-OS tasks.

The DSP interface 120 also provides basic control functions for the multifunction I/O subsystem 14. The control functions are employed by the DSP-OS server 124. The control functions provided by the DSP interface 120 include control functions for loading the DSP-OS program image, control functions for starting and stopping the DSP 61, as well as control functions for obtaining diagnostic and status information from the multifunction I/O subsystem 14.

An application executing on the DSP 61 consists of one or more DSP-OS tasks that typically performs computations. A DSP-OS task has an associated host device driver that functions as a server. The host device driver launches the DSP-OS tasks (via a call to dspOpenTask) and thereafter sends commands to the newly launched DSP-OS tasks. The DSP-OS tasks service requests from the corresponding host device driver. A host device driver and a corresponding DSP-OS task have a client/server relationship.

The DSP-OS tasks are created through the DSP-OS dispatcher task 146. The DSP-OS dispatcher task 146 is a special DSP-OS task created when the system starts up. The DSP-OS dispatcher task 146 communicates with the host DSP driver 122. The DSP-OS dispatcher task 146 sets up the DSP-OS environment on the multifunction I/O subsystem 14 and then assumes the

role of a central control DSP-OS task. The DSP-OS dispatcher task 146 responds to a set of command messages from the host device drivers, including requests to create and delete a DSP-OS task.

- 5 The DSP-OS tasks send data to the host device drivers, and receive data from the host device drivers. A data streaming interface enables efficient transmission of data blocks (buffers). The data streaming interface is accessed through the DSP-OS. The host application programs access the data channels between a DSP-OS task and the corresponding host device driver through the
10 host device driver. The DSP-OS tasks access files on the mass storage subsystem 33 through a corresponding host device driver.

- The DSP-OS supports device independent I/O by providing a common set of functions used to access the differing hardware devices of the
15 multifunction I/O subsystem 14 in a uniform manner. The DSP device drivers 130 and 135-137 implement the DSP-OS device-independent I/O interfaces 140-143 to enable access of the hardware devices of the multifunction I/O subsystem 14. The DSP-OS tasks 132-134 and 146 employ the device-independent I/O interfaces 140-143 to transfer data to or from any
20 hardware device of the multifunction I/O subsystem 14 as a source or a sink.

- The DSP-OS tasks 132-134 and 146 read input data from a source device, processes the input data, and then write the processed data to a sink device. An SS_get and an SS_put operation are used for such processing. The local
25 line 20, the telephone line 22, the stereo audio codec circuit 54, and the hardware devices of the host processor 32 are available as sources and sinks. The mass storage subsystem 33 functions as multiple sinks and multiple

sources simultaneously, such that multiple logical data channels are concurrently active.

The multifunction I/O subsystem 14 provides hardware devices that support multiple application program interfaces. For example, the stereo audio codec circuit 54 is employed for sound file input and output through an audio application program interface, and for telephone conversations through a telephony application program interface. The telephone line 22 is employed for telephone conversations through the telephony application program interface, for sending and receiving faxes through a fax application program interface, and for data modem processing. In all such cases, the particular hardware device is accessed via the DSP-OS device-independent I/O interface.

The hardware devices of the multifunction I/O subsystem 14 that generate and consume sampled sound data streams represent the sampled sound data in differing data formats. A hardware device or DSP device driver may represent sound data in amplitude normalized format, wherein the sound samples comprise 32-bit floating point left justified data values that are scaled to range from -2^{31} to $2^{31}-1$ for conversion to 32-bit signed integers.

A hardware device or DSP device driver may represent sound data as 32-bit floating point right justified data values, wherein 8-bit sound samples range from -256 to 255 and 16-bit sound samples range from -32768 to 32767.

A hardware device or device driver may represent sound data as 16-bit signed integer right justified data values with one sample per 32-bit word, wherein sound samples range from -32768 to 32767.

A hardware device or DSP device driver may represent sound data as packed 16-bit integer data values, wherein a 32-bit word either represents left and right samples (stereo) or two successive samples (mono). In addition, 8-bit μ -law samples are packed four to a 32-bit word.

DSP-OS tasks employ a standard data format to represent sampled sound data streams for the multifunction I/O subsystem 14. The standard data format is 32-bit floating-point left-justified. The sound data stream samples are scaled to range between -2^{31} and $2^{31}-1$. Such scaling enables conversion between floating point and integer data formats.

The standard data format supports a range of $\pm(2)^{127}$. However, sound sample values having a magnitude greater than 2^{31} generate erroneous results if sent to a hardware device of the multifunction I/O subsystem 14. Also, sound sample values having a magnitude less than 2^{15} are equivalent to 0.

For example, the DSP device driver for the stereo audio codec 54 receives standard data format samples from a corresponding DSP-OS task, converts the standard data format samples into 32-bit integers, then right-shifts the 32-bit integers by 16 bits to extract the high-order 16 bits, and then writes the high-order 16 bits to the stereo audio codec 54.

For another example, a DSP device driver for telephony converts standard data format sound samples into 8-bit μ -law samples by converting

the standard data format sound samples into 32-bit integers, then extracting the high-order 14 bits and converting the high-order 14 bits to 8-bit μ law.

The standard data format for sampled sound data streams enables the
5 DSP-OS tasks to open and read sound samples from multiple hardware devices that require differing data formats. Similarly, the standard data format for sampled sound data streams enables the DSP-OS tasks to write sound samples to multiple hardware devices that require differing data formats.

10

The standard data format for sampled sound data streams also enables the DSP-OS tasks to communicate with devices that generate the same data format with differing scaling. For example, the stereo audio codec 54 generates true 16-bit integer samples that range from -32768 to 32767, while
15 the DSP device driver for the telephone line 22 converts 8-bit μ law sound samples into a linear integer format having a 14-bit range (-4096 to 4095). The standard data format for sampled sound data streams enables compatibility if the sound samples for the telephone line 22 are left justified and the extra least significant bits set to zero.

20

Figure 6 illustrates the software architecture of the host processor 32 for one embodiment. The host application programs on the host processor 32 comprise a host telephony application program 200 and a host audio application program 202.

25

The host telephony application program 200 enables origination and reception of voice calls, data modem calls, and fax calls over the telephone

line 22. The host telephony application program 200 performs high-level user interface functions and telephony handshake functions for handling the voice, data modem, and fax transfers over the telephone line 22.

5 The host telephony application program 200 communicates with a host phone control driver 160 and a host fax-modem driver 166 through a telephony application program interface 170. The host phone control driver 160 and the host fax-modem driver 166 communicate with corresponding DSP-OS tasks on the multifunction I/O subsystem 14 through the DSP
10 interface 120 of the host DSP driver 122.

 The telephony application program interface 170 provides telephony control services to the telephony application program 200. The telephony application program interface 170 provide services for handling inbound
15 calls, for initiating outbound calls, for accessing features provided by the analog phone line interface circuit 70, and for communicating with remote equipment.

 A fax application program interface 172 provides a set of fax control
20 operations to the telephony application program 200. The fax application program interface 172 provides operations for managing incoming and outgoing faxes.

 The host audio application program 202 enables recording and playback
25 of sound files. The audio application program 202 enables transfer of information between sound files on the mass storage subsystem 33 and audio playback devices (not shown) coupled to the audio signal lines 56. The audio

application program 202 also enables transfer of information between sound files on the mass storage subsystem 33 and the serial telephone 15 coupled to the local line 20.

5 The host audio application program 202 communicates with a host audio driver 164 through an audio application program interface 171. The host audio driver 164 communicates with corresponding DSP-OS tasks on the multifunction I/O subsystem 14 through the DSP interface 120 of the host DSP driver 122.

10

 The functions provided by the audio application program interface 171 include querying audio devices, opening and closing audio devices, recording and playing audio data, and controlling playback and recording.

15 The host audio application program 202 also communicates with the host phone control driver 160 through the telephony application program interface 170 if the serial telephone 15 is employed as a sound file input/output device.

20 The host audio application program 202 implements a user interface function that enables a user of the personal computer 12 to select the serial telephone 15 as the sound file input/output device.

 For one embodiment, the audio application program interface 171
25 comprises a Waveform application program interface provided by Microsoft® Corporation for processing Waveform sound files. The Waveform application program interface is part of Microsoft® Windows. The

Waveform application program interface provides sound record and playback functions to Microsoft® Windows host applications.

Figure 7 illustrates the software architecture of the multifunction I/O subsystem 14 for one embodiment. A DSP-OS dispatcher task 146 processes requests from the host phone control driver 160, the host audio driver 164 and the host fax-modem driver 166 to create and delete DSP-OS tasks.

The DSP-OS dispatcher task 146 matches the requested task name to the appropriate DSP-OS tasks and then loads and starts the new DSP-OS tasks. The DSP-OS dispatcher task 146 also creates mailboxes for communications between the host phone control driver 160 and the host fax-modem driver 166 and corresponding DSP-OS tasks. The DSP-OS dispatcher task 146 also creates mailboxes for communications between the host audio driver 164 and a corresponding DSP-OS task. The DSP-OS dispatcher task 146 also creates mailboxes for communications between the new DSP-OS tasks and other DSP-OS tasks.

A DSP file input/output (FIO) driver 270 enables access the host processor 32 file system by the DSP-OS tasks. The DSP-OS tasks use the DSP FIO driver 270 to perform open, read, write, and close operations, and to transfer data to and from files on the host processor 32 including files on the mass storage device 33.

The host phone control driver 160 and the host fax-modem driver 166 cause the DSP-OS dispatcher task 146 to create a set of telephony DSP-OS tasks 260 - 263 for normal processing of incoming calls over the telephone line 22.

The digital set emulation DSP-OS task 262 is created if the serial telephone 15 is a digital telephone. The digital PBX emulation DSP-OS task 263 is created if the telephone line 22 is coupled to a digital PBX network.

5 The telephony DSP-OS tasks 260 - 263 perform analog and digital telephony functions through a DSP telephony driver 282. The DSP telephony driver 282 communicates through the phone interface circuit 66 to the analog phone line interface circuit 60 and the codec circuits 70 and 72.

10 The host audio driver 164 causes the DSP-OS dispatcher task 146 to create an audio DSP-OS task 252. The audio DSP-OS task 252 performs sound file record and playback functions through a DSP stereo audio codec driver 280 in a normal mode. The DSP stereo audio codec driver 280 communicates through the audio interface circuit 68 to the stereo audio codec circuit 54.

15

 The audio DSP-OS task 252 also performs sound file record and playback functions through the DSP telephony driver 282 in a telephone audio I/O mode. A user interface function of the host audio application program 202 determines whether the audio DSP-OS task operates in the
20 normal mode or the telephone audio I/O mode.

 The audio DSP-OS task 252 functions as a server for the host audio driver 164. The audio DSP-OS task 252 performs the commands requested by the host audio driver 164 according to messages routed through the
25 corresponding mailboxes.

During normal mode audio record operations, the DSP stereo audio codec driver 280 receives sound samples from the stereo audio codec circuit 54. The sound samples from the stereo audio codec circuit 54 comprise 16 bit integer data values. The DSP stereo audio codec driver 280 converts the
5 sound samples into a sound data stream having the standard data format, and transfers the sound data stream to the audio DSP-OS task 252.

The audio DSP-OS task 252 preprocesses the sound data stream through either an 8:1 compression driver 210, a 2:1 compression driver 212, or a pulse
10 code modulation (PCM) audio driver 214 according to the requirements of the host application program 202. The preprocessed sound data stream is then transferred to the host audio application program 202 through the device independent I/O interface of the DSP host driver 130, and up through the host audio driver 164.

15

The 8:1 compression driver 210 converts the sound data stream from the standard data format into an 8:1 compressed data format. The 2:1
compression driver 212 converts the sound data stream from the standard data format into an 2:1 compressed data format. The PCM audio driver 214
20 converts the sound data stream from the standard data format into a PCM format for Waveform files.

During normal mode audio playback operations, the host audio application program 202 transfers a sound data stream to the host audio
25 driver 164 through the audio application program interface 171. The host audio driver 164 transfers the sound data stream to the host DSP driver 122 through the DSP interface 120. The audio DSP-OS task 252 receives the sound

data stream from the host DSP driver 122 through either the 8:1 compression driver 210, the 2:1 compression driver 212, or the PCM audio driver 214 according to the requirements of the host application program 202.

5 The 8:1 compression driver 210 converts the sound data stream from the 8:1 compressed data format into the standard data format during audio playback. The 2:1 compression driver 212 converts the sound data stream from the 2:1 compressed data format into the standard data format during audio playback. The PCM audio driver 214 converts the sound data stream
10 from the PCM format for Waveform files into the standard data format during playback.

 The audio DSP-OS task 252 then transfers the standard format sound data stream to the DSP stereo audio codec driver 280 through the device
15 independent I/O interface. The DSP stereo audio codec driver 280 converts the sound data stream into the 16 bit integer data values required by the stereo audio codec circuit 54 and then transfers the decompressed 16 bit sound data values to the stereo audio codec circuit 54 to generate sound playback.

20 During telephone audio I/O mode record operations, the DSP telephony driver 282 receives sound samples of the serial telephone 15 over the local line 20 from the analog phone line interface circuit 60 and the codecs 70 and 72. The DSP telephony driver 282 converts the sound samples into a sound data stream having the standard data format. The DSP telephony
25 driver 282 then transfers the sound data stream to the audio DSP-OS task 252.

The audio DSP-OS task 252 preprocesses the sound data stream through either the 8:1 compression driver 210, the 2:1 compression driver 212, or the PCM audio driver 214 to convert the sound data stream from the standard data format to an application dependent format according to the requirements
5 of the host application program 202. The preprocessed sound data stream is then transferred to the host audio application program 202 through the device independent I/O interface of the DSP host driver 130, and up through the host audio driver 164.

10 During telephone audio I/O mode playback operations, the host audio application program 202 transfers a sound data stream to the host audio driver 164 through the audio application program interface 171. The host audio driver 164 transfers the sound data stream to the host DSP driver 122 through the DSP interface 120. The audio DSP-OS task 252 receives the sound
15 data stream from the host DSP driver 122 through either the 8:1 compression driver 210, the 2:1 compression driver 212, or the PCM audio driver 214 to convert the sound data stream into the standard data format.

The audio DSP-OS task 252 then transfers the sound data stream to the
20 DSP telephony driver 282 through the device independent I/O interface. The DSP telephony driver 282 converts the standard data format sound data stream into sound data values required by the analog phone line interface circuit 60 and the codecs 70 and 72. The DSP telephony driver 282 then transfers the decompressed sound data values to the analog phone line
25 interface circuit 60 and the codecs 70 and 72 to generate sound playback on the serial telephone 15 over the local line 20.

For one embodiment, the host audio application program 202 transfers sound data values from Waveform sound files to the audio DSP-OS task 252 during audio playback operations. The Waveform sound samples conform to the Waveform application program interface provided by Microsoft®

- 5 Corporation for processing . The Waveform sound samples support sample rates of 8 Khz mono or stereo, 8 Khz mono or stereo, 11.025 Khz mono or stereo, 16 Khz mono or stereo, 22.05 Khz mono or stereo, 32 Khz mono or stereo, 44.1 Khz mono or stereo, and 48 Khz mono or stereo. Each sample comprises either 8 bits or 16 bits.

10

- If the serial telephone 15 comprises a digital telephone, then sample rate conversion and stereo to mono conversion is performed for digital telephone compatibility. The audio DSP-OS task 252 performs sample rate conversion on the Waveform sound samples, and performs a stereo to mono
- 15 conversion on the sound samples if necessary to conform the sound samples to the sample rate required by the serial telephone 15.

- The standard format sound data streams are processed by a universal conversion driver 220 to perform the required sample-rate conversion and
- 20 stereo-to-mono conversion. The universal conversion driver 220 converts a stream of standard format data from one sample rate to another using standard DSP up sampling, down sampling, and filtering techniques.

- The universal conversion driver 220 converts stereo-to-mono by
- 25 extracting interleaved samples from a standard data format sound data stream and summing together the interleaved samples to produce a non-interleaved data stream. The universal conversion driver 220 converts mono-to-stereo by

duplicating samples from a standard data format sound data stream and interleaving the samples.

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



CLAIMS

1. A method for audio recording in a computer system, comprising the
5 steps of:
- selecting a normal audio mode or a telephone audio mode for the
computer system;
- if the telephone audio mode is selected, generating a set of sound
samples by sampling audio signals over a local line coupled to a telephone,
10 and converting the sound samples into a sound data stream having a
standard data format;
- if the normal audio mode is selected, generating the sound samples by
sampling audio signals over an audio line, and converting the sound samples
into the sound data stream having the standard data format;
- 15 transferring the sound data stream to an audio task executing on a
digital signal processing subsystem in the computer system, the audio task
converting the sound data stream into a format required by a host audio
application program executing on a host processor in the computer system,
the audio task then transferring the sound data stream to the host audio
20 application program.
2. The method of claim 1, wherein the step of converting the sound data
stream into a format required by a host audio application program executing
on a host processor in the computer system comprises the step of performing
25 a data compression function on the sound data stream.

REPLACED SHEETS SG

- ☐ Description : paged to paged
☒ Claim(s) : paged 31 to paged 37
☐ Drawing(s) : paged to paged
☐ Abstract : paged to paged



This Sheet Has Been Replaced



- 32 -

3. The method of claim 1, wherein the step of converting the sound data stream into a format required by a host audio application program executing on a host processor in the computer system comprises the step of converting the sound data stream from the standard data format into a pulse code modulation data format.
4. The method of claim 1, wherein the telephone comprises a digital telephone.
5. The method of claim 4, wherein the steps of generating a set of sound samples by sampling audio signals over a local line coupled to a telephone, and converting the sound samples into a sound data stream having a standard data format further comprises the step of performing a sample rate conversion on the sound data stream from a data rate corresponding to the digital telephone to a predefined sample rate supported by the host audio application program.
6. A method for audio playback in a computer system, comprising the steps of:
- selecting a normal audio mode or a telephone audio mode for the computer system;
 - receiving a sound data stream from a host audio application program executing on a host processor in the computer system;
 - converting the sound data stream from a data format required by the host audio application program into a standard data format;
 - transferring the sound data stream having the standard data format to an audio driver executing on a digital signal processing subsystem in the



computer system if the normal audio mode is selected, the audio driver converting the sound data stream into a set of sound data values and transferring the sound data values to a stereo audio codec circuit for sound generation over an audio signal line;

- 5 transferring the sound data stream to a telephony driver executing on the digital signal processing subsystem if the telephone audio mode is selected, the telephony driver converting the sound data stream into a set of telephony data values and transferring the telephony data values to a telephony circuit, such that the telephony circuit generates telephony audio
10 signals corresponding to the telephony data values and transfers the telephony audio signals to a telephone over a local line.

7. The method of claim 6, wherein the step of converting the sound data stream from a data format required by the host audio application program
15 into a standard data format comprises the steps of performing a data decompression function on the sound data stream and converting the decompressed sound data stream into the standard data format.

8. The method of claim 6, wherein the step of converting the sound data
20 stream from a data format required by the host audio application program into a standard data format comprises the step of converting the sound data stream from a pulse code modulation data format into the standard data format.

- 25 9. The method of claim 6, wherein the telephone comprises a digital telephone.



This Sheet Has Been Replaced



- 34 -

10. The method of claim 9, wherein the step of transferring the sound data stream to a telephony driver executing on the digital signal processing subsystem further comprises the step of performing a sample rate conversion on the sound data stream from a predefined sample rate supported by the
5 host audio application program to a data rate corresponding to the digital telephone.

11. An apparatus for audio recording in a computer system, comprising:
means for selecting a normal audio mode or a telephone audio mode
10 for the computer system;
means for generating a set of sound samples by sampling audio signals over a local line coupled to a telephone, and converting the sound samples into a sound data stream having a standard data format if the telephone audio mode is selected;
15 means for generating the sound samples by sampling audio signals over an audio line, and converting the sound samples into the sound data stream having the standard data format if the normal audio mode is selected;
means for transferring the sound data stream to an audio task executing on a digital signal processing subsystem in the computer system,
20 the audio task converting the sound data stream into a format required by a host audio application program executing on a host processor in the computer system, the audio task then transferring the sound data stream to the host audio application program.

25 12. The apparatus of claim 11, wherein the means for converting the sound data stream into a format required by a host audio application program



This Sheet Has Been Replaced



- 35 -

executing on a host processor in the computer system comprises means for performing a data compression function on the sound data stream.

13. The apparatus of claim 11, wherein the means for converting the
5 sound data stream into a format required by a host audio application program
executing on a host processor in the computer system comprises means for
converting the sound data stream from the standard data format into a pulse
code modulation data format.

10 14. The apparatus of claim 11, wherein the telephone comprises a digital
telephone.

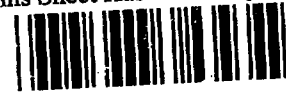
15 15. The apparatus of claim 14, wherein the means for generating a set of
sound samples by sampling audio signals over a local line coupled to a
15 telephone, and converting the sound samples into a sound data stream
having a standard data format further comprises means for performing a
sample rate conversion on the sound data stream from a data rate
corresponding to the digital telephone to a predefined sample rate supported
by the host audio application program.

20

16. An apparatus for audio playback in a computer system, comprising:
means for selecting a normal audio mode or a telephone audio mode
for the computer system;
means for receiving a sound data stream from a host audio application
25 program executing on a host processor in the computer system;
means for converting the sound data stream from a data format
required by the host audio application program into a standard data format;



This Sheet Has Been Replaced



- 36 -

means for transferring the sound data stream having the standard data format to an audio driver executing on a digital signal processing subsystem in the computer system if the normal audio mode is selected, the audio driver converting the sound data stream into a set of sound data values and
5 transferring the sound data values to a stereo audio codec circuit for sound generation over an audio signal line;

means for transferring the sound data stream to a telephony driver executing on the digital signal processing subsystem if the telephone audio mode is selected, the telephony driver converting the sound data stream into
10 a set of telephony data values and transferring the telephony data values to a telephony circuit, such that the telephony circuit generates telephony audio signals corresponding to the telephony data values and transfers the telephony audio signals to a telephone over a local line.

15 17. The apparatus of claim 16, wherein the means for converting the sound data stream from a data format required by the host audio application program into a standard data format comprises means for performing a data decompression function on the sound data stream and converting the decompressed sound data stream into the standard data format.

20 18. The apparatus of claim 16, wherein the means for converting the sound data stream from a data format required by the host audio application program into a standard data format comprises means for converting the sound data stream from a pulse code modulation data format into the
25 standard data format.



This Sheet Has Been Replaced



- 37 -

19. The apparatus of claim 16, wherein the telephone comprises a digital telephone.
20. The apparatus of claim 19, wherein the means for transferring the
5 - sound data stream to a telephony driver executing on the digital signal processing subsystem further comprises means for performing a sample rate conversion on the sound data stream from a predefined sample rate supported by the host audio application program to a data rate corresponding to the digital telephone.
21. A method for audio recording in a computer system, substantially as hereinbefore described with reference to the accompanying drawings.
22. An apparatus for audio recording in a computer system, substantially as hereinbefore described with reference to the accompanying drawings.