ELECTRICAL MUSIC APPARATUS CAPABLE OF CONNECTING WITH EXTERNAL DEVICE

CROSS REFERENCE TO RELATED APPLICATION

This application is based on Japanese Patent Application 2004-230930, filed on August 6, 2004, and Japanese Patent Application 2004-230930, filed on August 6, 2004, the entire contents of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

A) FIELD OF THE INVENTION

The present invention relates to an electrical music apparatus and specifically relates to an electrical music apparatus that can be used while connecting with an external device.

B) DESCRIPTION OF THE RELATED ART

Conventionally, functions of an electric music apparatus such as an electronic musical instrument, a musical tone generator, a mixer, etc. have been extended by inserting an external device such as a LAN-card, etc. into its extended slot. In order to use the external device under control of the electrical music apparatus, a device driver (hereinafter called the driver) corresponding to the external device must be installed to the electrical music apparatus.

Generally, the driver is installed by selecting from built-in drivers pre-stored in the electric music apparatus. When the driver corresponding to the external device can not be found in the built-in drivers, the installation of the driver will be executed by loading it from a storage medium distributed with the external device in a form of a CD-ROM or the likes to the electric music apparatus, or by downloading it from a server via a public network such as the Internet to the electric music apparatus.

A communication setting apparatus that detects a driver corresponding

to the connected (inserted) external device automatically from a plurality of drivers stored in the apparatus is well known. The communication setting apparatus transmits data to the external device by using each one of the plurality of the drivers and judges that each driver supports the external device by detecting responses from the eternal device (e.g., refer to Japanese Laid-open Patent No. 2003-256348).

In the above-described type of the communication setting apparatus, even if an external device of which compatibility with the communication setting apparatus has been already assured is connected, the detections and judgments are carried out for every built-in drivers; therefore, the automatic installation of the driver takes a long time.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide an electric music apparatus that can quickly determine a device driver for connected external device.

According to one aspect of the present invention, there is provided an electric music apparatus, comprising: an interface to which an external device can connect; a memory that stores device drivers for operating external devices connected to the interface and a list recording relationship between external devices and corresponding device drivers; an obtaining device that obtains external device information from an external device connected to the interface; a searching device that searches a device driver corresponding to the connected external device with reference to the list in accordance with the obtained external device information; a first determining device that determines to control the connected external device by using the device driver corresponding to the

connected external device when the searching device detects the corresponding device driver from the list; and a second determining device that executes a specific command to the connected external device by using a device driver stored in the memory when the searching device cannot detect the corresponding device driver and determines to control the connected external device by using a device driver that makes the external device respond to the specific command without an error.

According to the present invention, an electric music apparatus that can quickly determine a device driver for connected external device can be provided.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram showing a hardware structure of an electric music apparatus 1 according to an embodiment of the present invention.

FIG. 2A and FIG. 2B are schematic diagrams showing operation confirmed drivers lists according to the embodiment of the present invention.

FIG. 3 is a schematic diagram showing relationships between external devices and corresponding drivers listed in the operation confirmed drivers lists shown in FIG. 2.

FIG. 4 is a flowchart showing a driver determination process according to the embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 is a block diagram showing a hardware structure of an electric music apparatus 1 according to an embodiment of the present invention.

The electric music apparatus 1 is an electronic device for executing a

process relating music such as an electronic musical instrument like a musical keyboard, a musical tone generator, a PA device like a mixer or a recorder, and an audio-visual device like a stereo component system, etc.

The electric music apparatus 1 is consisted of a bus 6, a RAM 7, a ROM 8, a CPU 9, a timer 10, a detecting device 11, a panel operating switch 12, a musical performance switch 22, a display control device 13, a display 14, an external storage device 15, a MIDI interface 16, a musical tone generator 18, an effecter 19, a sound system 20 and an extended slot 23.

To the bus 6 are connected the RAM 7, the ROM 8, the CPU 9, the detecting device 11, the display control device 13, the external storage device 15, the MIDI interface 16, the musical tone generator 18, the effecter 19 and the extended slot 23.

The RAM 7 has a buffer region and a working area of the CPU 9 where a flag, a register, various parameters, etc. are stored.

The ROM 8 can store various parameters, various control programs, a program for realizing the embodiment of the present invention, etc. In addition, the ROM 8 can store device drivers (drivers), each supporting at least one of plurality types of external devices 24.

The CPU 9 executes a calculation and controls the electric music apparatus 1 in accordance with the controlling programs and the program for realizing the embodiment stored in the ROM 8 or the external storage device 15. The timer 10 is connected to the CPU 9 and supplies a standard clock signal, an interruption timing, etc. to the CPU 9.

A user can input various parameters and settings by using the panel operating switch 12 connected to the detecting device 11. The panel operating switch may be any type of a switch, for example, a switch, a pad, a fader, a slider,

an alpha-numerical keyboard, a mouse, a rotary encoder, a joy stick, a jog shuttle, etc. that can input a signal in accordance with an input operation of the user. In addition, the electric music apparatus may be equipped with a plurality of the panel operating switches of the same type or the different types. Moreover, the panel operating switch 12 may be a software switch displayed on the display 14, which is operated by using other switches like a mouse, etc.

The musical performance switch 22 is connected to the detecting device 11 and supplies a musical performance signal in accordance with a musical performance by a user. A musical keyboard, a pad, etc. can be used as the musical performance switch 22. In addition, the musical performance switch 22 may be any type of switch that can input a musical performance signal in accordance with a musical performance by a user. Moreover, the electric music apparatus may be equipped with a plurality of the musical performance switches 22 of the same type or the different types.

The display controlling device 13 is connected to the display 14 and controls the display 14 to display various information on its screen. A user executes various inputs and settings with reference to the information displayed on the screen of the display 14. Moreover, the display 14 may be an external device connected to the electric music apparatus 1 via a cable, etc.

The MIDI interface (MIDI I/F) 16 can be connected to a MIDI device 17, other electronic musical instrument, an audiovisual device, a computer, etc. and is an interface that can transmit and receive at least a MIDI signal. The MIDI interface (MIDI I/F) 16 is not limited to a dedicated MIDI interface but also may be a general interface based on RS-232C, universal serial bus (USB), IEEE1394 Standards, etc. In this case, other data than a MIDI signal can be simultaneously transmitted or received.

The MIDI device 17 is an electronic musical instrument or an audiovisual device connected to the MIDI interface 16. A type of the MIDI device 17 is not only a keyboard type but also a stringed instrument type, a wind instrument type, a percussion type, etc. In addition, the MIDI device 17 is not only a device equipped with a musical tone generator, an automatic musical performance device etc. in one body but also a group of devices connected by a communication network via a communication interface such as a MIDI interface, USB, IEEE1394, etc.

The musical tone generator 18 generates a sound signal in accordance with music contents store in the ROM 8 or the RAM 7, etc., a musical performance signal supplied from the musical performance switch 22 or from the MIDI device 17 connected to the MIDI interface 16 or a MIDI signal and supplies the sound signal to the sound system 20 via the effecter 19.

The effecter 19 adds various musical effects to the sound signal supplied from the musical tone generator 18.

The sound system 20 converts the supplied digital sound signal to an analogue sound signal by a built-in D/A converter and sounds it by loud speakers.

The extended slot 23 is a general interface for attaching an external device such as a PCI slot, a PCMCIA slot, etc. An external device 24 is connected to the electrical music apparatus 1 via the extended slot 23. The extended slot 23 is not only an extended slot to which the external device 24 is inserted to use but also any type of an interface to which the external device 24 can be connected to use, e.g., USB interface, etc.

The external device 24 is inserted (or connected) to the extended slot 23 to extend a function of the electric music apparatus and is a device commonly available for a use with a personal computer. The external device 24 is

preferably a type of device that one device driver can support a plurality of external devices from different vendors because a maker of a built-in chip (processor, etc.) of the external device is limited to small numbers although a multiplicity of vendors (makers) exist, namely, a LAN adaptor (card), an audio interface, USB loudspeakers, a storage device (a semiconductor memory, HDD, FDD, DVD drive, CD-ROM drive, etc.), a modem, a monitor display, an external musical tone generator, etc.

Further, the external device 24 stores external device information including a vendor ID specifying a company (manufacturer or distributor) of the external device 24 and a product ID specifying a product. The external device 24 supplies the external device information to the electric music apparatus 1 upon request. In addition, the external device information is not only consisted of a combination of the vendor ID and the product ID but also any information that can specify the external device 24.

The external storage device 15 includes an interface for an external storage device and is connected to the bus 6 via the interface.

The external storage device 15 has a storage area for storing the later-described user list CL2. As the external device 15, for example, a semiconductor memory such as a flash memory, etc. may be equipped with.

In addition to that, as the external storage device 15, a flexible or floppy (trademark) disk drive (FDD), a hard disk drive (HDD), a magnet optical (MO) drive, a compact disk read only memory (CD-ROM) drive, a digital versatile disc (DVD) drive, etc. may be connected. Furthermore, those drives may be connected via the extended slot 23.

The electric music apparatus 1 may be equipped with a communication interface 21. The communication interface 21 can connect with a

communication network 3 such as a local area network (LAN), the Internet, public phone line, etc. and can connect bilaterally with a server 5 via the communication network 3.

The communication interface 21 and the communication network 3 are not only wired but also wireless or a combination of wired and wireless networks. Moreover, the communication interface 21 may be equipped with the electric music apparatus 1 as a built-in device or connected to the electric music apparatus 1 via the extended slot 23 as the external device 24.

By the way, this embodiment is considered to be more effective for the electric music apparatus 1 that is not equipped with an external storage device that is used by inserting an external storage medium and the communication interface 21 at a time of distribution and to which a device driver supplied with the external device 24 is difficult to be installed. Therefore, the external storage device that is used by inserting an external storage medium and the communication interface 21 are not essential to realize the embodiment of the present invention.

FIG. 2A and FIG. 2B are schematic diagrams showing operation confirmed drivers lists according to the embodiment of the present invention. FIG 2A shows an operation confirmed drivers list (preset list) CL1 stored in the ROM 8 in advance. FIG. 2B shows an operation confirmed drivers list (user list) CL2 created or overwritten by the later-described driver determination process shown in FIG. 4.

Both of the preset list CL1 and the user list CL2 record relationships between the external device information including the vendor ID specifying company and the product ID specifying the product and a driver ID specifying a device driver supporting an external device corresponding to the external device

information.

In the user list CL2, an external device that is confirmed not to be used by the electric music apparatus 1 is further recorded as an unsupported device. In this embodiment, the driver ID "#0" means that there is no supporting driver for the external device. The way to show absence of the supporting driver may be any way that the absence of the driver can be recognized.

FIG. 3 is a schematic diagram showing relationships between external devices and corresponding drivers listed in the operation confirmed drivers lists shown in FIG. 2.

The electric music apparatus 1 according to this embodiment has a driver storage region DS, for example, in the ROM 8 shown in FIG. 1 and stores plurality types of device drivers (a driver A, a driver B, a driver C and so on) in the driver storage region DS. In addition, those drivers may be ranked in accordance with compatibility or functions.

For example, when a product A from a company A is inserted into the extended slot 23 shown in FIG. 1, a device driver supporting (corresponding to) the product A is searched the vendor ID "#2" and the product ID "#5" as searching keys at first in the user list CL2 and thereafter in the preset list CL1. Then, it will be found that the device driver corresponding to the vendor ID "#2" and the product ID "#5" is the driver A having the driver ID "#1".

For another example, when a product C from a company B is inserted into the extended slot 23 shown in FIG. 1, a device driver supporting (corresponding to) the product C is searched the vendor ID "#10" and the product ID "#21" as searching keys at first in the user list CL2 and thereafter in the preset list CL1. Then, it will be found that the device driver corresponding to the vendor ID "#10" and the product ID "#21" is the driver A having the driver ID "#1".

In this embodiment, the driver A is recorded as being used for the product A from the company A and the product C from the company B. There are many cases that the same driver can support the products from the different vendors as the above-described examples. Therefore, by storing combination of products and a driver, the electric music apparatus 1 can support much more external devices. Generally, an electric music apparatus is not equipped with a CD-ROM drive, etc. and cannot be connect with the Internet, etc., and so a device driver completely supporting the external device is difficult to be installed. Therefore, a relationship (combination) between the external device and the device driver is stored even if the vendor of the external device does not distribute the driver as far as the driver can use the external device.

Further, for example, when a product G from a company C is inserted into the extended slot 23 shown in FIG. 1, a device driver supporting (corresponding to) the product G is searched the vendor ID "#83" and the product ID "#31" as searching keys at first in the user list CL2 and thereafter in the preset list CL1. Then, it will be found that the device driver corresponding to the vendor ID "#83" and the product ID "#31" is the driver C having the driver ID "#3".

Moreover, for example, when a product F from the company A is inserted into the extended slot 23 shown in FIG. 1, a device driver supporting (corresponding to) the product F is searched the vendor ID "#2" and the product ID "#7" as searching keys at first in the user list CL2 and thereafter in the preset list CL1. Then, it will be found that the device driver corresponding to the vendor ID "#2" and the product ID "#7" is a driver having the driver ID "#0", that is, the driver supporting the product F is not stored in the driver storage region DS.

For other external devices, the corresponding drives are determined with reference to the operation confirmed drivers lists as similar to the external

devices in the above-described examples, and their relationships will be as shown in the drawing.

FIG. 4 is a flowchart showing a driver determination process according to the embodiment of the present invention. This process is executed by the CPU 9 in accordance with the program stored in the ROM 8, etc. of the electric music apparatus 1. Further, the driver determination process is started when the electric music apparatus 1 is turned on, setting relating to the external device is executed, connection of a new external device 24 is detected or at a time of usage of the external device by a user. The time of usage of the external device is, for example, a time when the user starts browsing software to connect with the Internet, etc. if the external device is a LAN adaptor or a modem.

At Step SA1, the driver determination process is started. At Step SA2, information of an un-configured external device 24 connected (or newly connected) to the extended slot 23 (FIG. 1) is obtained. The information obtained at this step is, for example, the vendor ID and the product ID stored in the external device 24. The vendor ID and the product ID uniformly are called the external device information.

At Step SA3, a device driver supporting (corresponding to) the external device 24 is searched with reference to, for example, the operation confirmed drivers list (user list) CL2 shown in FIG. 2B with the external device information obtained at Step SA2 as keys. In this embodiment, the searching is executed from the top of the user list CL2 and stopped when the corresponding driver is found.

At Step SA4, it is judged that the external device of which driver is searched at Ste SA3 is registered as an unsupported device (e.g., the driver ID corresponding to the external device is "#0") in the user list CL2 or not. When

the external device is an unsupported device, the process advances to Step SA 17 as directed by an arrow marked "YES". When the external device is not an unsupported device, the process advances to Step SA 5 as directed by an arrow marked "NO".

At Step SA5, It is judged whether the device driver supporting (corresponding to) the external device 24 is found by the searching at Step SA3 or not. When the corresponding driver is detected (found), the process advances to Step SA16 as directed by an arrow marked "YES". When the corresponding driver is not detected (found), the process advances to Step SA6 as directed by an arrow marked "NO".

At Step SA6, the device driver supporting (corresponding to) the external device 24 is searched with reference to, for example, the operation confirmed drivers list (preset list) CL1 shown in FIG. 2A with the external device information obtained at Step SA2 as keys. In this embodiment, the searching is executed from the top of the preset list CL1 and stopped when the corresponding driver is found.

At Step SA7, It is judged whether the device driver supporting (corresponding to) the external device 24 is found by the searching at Step SA6 or not. When the corresponding driver is detected (found), the process advances to Step SA16 as directed by an arrow marked "YES". When the corresponding driver is not detected (found), the process advances to Step SA8 as directed by an arrow marked "NO".

At Step SA8, a device driver that has not been tested (untested driver) is loaded from the driver storage region DS. In this embodiment, a plurality of the device drivers are stored in the driver storage region DS, and the device drivers may be ranked with priory sequence to be loaded. For example, a driver

with higher performance may have priority over others to be selected more frequently or a driver with higher compatibility or with higher availability may have priority over others to increase processing speed for testing the drivers.

At Step SA9, a specific command sequence is executed by using the device driver loaded at Step SA8. The specific command sequence executed at this step may be initialization of the external device. Commands sent to the external device for the specific commands sequence (e.g., the initialization process) are different depending on the device drivers; therefore, the specific commands sequence (e.g., the initialization process) cannot be completed (the external device does not respond or responds to the specific command with an error message) when a device driver not supporting (corresponding to) the external device is used.

At Step SA10, it is judged whether the specific commands sequence (e.g., the initialization process) executed at Step SA9 is completed or not. When the command sequence is successfully executed, the process advances to Step SA13 as directed by an arrow marked "YES". When the command sequence is not successfully executed, that is, it turns to be error, the process advances to Step SA11 as directed by an arrow marked "NO".

At Step SA11, it is judged whether the testing for all the device drivers stored in the driver storage region DS has been completed or not. When the testing for all the device drivers has been completed, the process advances to Step SA12 as directed by an arrow marked "YES", and the external device (the external device information of the external device) is added to the top of the user list CL2 as the unsupported device. Thereafter, the process advances to Step SA17. When the testing for all the device drivers has not been completed yet, the process returns to Step SA8 as directed by an arrow marked "NO", and the

next device driver to be tested will be loaded.

At Step SA13, it is asked to the user that the user would like to use the external device without an operation guarantee. For example, the inquiry to the user may be executed by letting the user select from "use" or "not use" with displaying statement to the display 14, telling that an operation of the external device is not guaranteed. The reason why the operation of the external device is not guaranteed is that all the operations of the external device cannot be guaranteed in accordance with testing just one command sequence and an error may be occurred in some operations.

At Step SA14, it is judged whether the user selected to use the external device upon the inquiry at Step SA13 or not. When the user selected to use the external device, the process advances to Step SA15 as directed by an arrow marked "YES". When the user selected not to use the external device, the process advances to Step SA17 as directed by an arrow marked "NO".

At Step SA15, a relationship between the external device (the external device information) and the device driver by which the command sequence has been successfully executed at Step SA9 is added to the user list CL2. For example, the external device information and the driver ID are added to the user list CL2.

At Step SA16, the electric music apparatus 1 is set to use the external device by using the specific driver (the device driver found by searching at Step SA5 or SA7 or the device driver by which the command sequence has been successfully executed at Step SA9. The process at this step includes validation of functions using the external device and loading of the specific driver. Thereafter, the process advances to Step SA18 and the driver determination process will be terminated.

At Step SA17, the electric music apparatus 1 is set not to use the external device. Thereafter, the process advances to Step SA18 and the driver determination process will be terminated.

As described in the above, according to the embodiment of the present invention, a relationship between the external device 24 connected to the electric music device 1 and the supporting driver is registered in the user list CL2; therefore, it is unnecessary to re-test the operation of the external device 24 by using available drivers, and the driver to be used can be quickly determined automatically.

In addition, the external device 24 of which supporting driver is not stored in the electric music apparatus 1 is registered as the unsupported device; therefore, when the unsupported device is connected to the electric music apparatus 1, it will be instantly known that the connected external device cannot be used, and the testing time will not be wasted.

Although the external device 24 of which supporting driver is not stored in the electric music apparatus 1 is registered as the unsupported device, and it is judged whether the external device is the unsupported device or not at Step SA4 in FIG. 4 in the embodiment, the processes at Step SA4 and Step SA12 may be omitted in order to simplify the driver determination process.

Although the operation confirmed lists are the preset list CL1 and the user list CL2 in the embodiment, a list combining those two lists may be used. In this case, combined list will be stored in the rewritable storage device (medium), and the newly confirmed relationship between the external device and the device driver will be added to the combined list.

Although the searching in the user list CL2 is executed before the searching in the preset list CL1 in the embodiment, the searching in the preset list

CL1 may be executed before the searching in the user list CL2.

Although the communication interface 21 is not necessary in the embodiment, when the electric music apparatus 1 is equipped with the communication interface 21 or a communication device such as a LAN adaptor, etc. is connected to the extended slot 23 as the external device 24, the electric music apparatus 1 may connect with the server 5 via the communication network 3 to update the preset list CL1, etc. and refer to the operation confirmed drivers lists stored in the server 5. By that, the electric music apparatus can handle a case that a device driver for the external device used with the electric music device has not been developed or that a genuine device driver for the external device cannot be installed due to specifications of the electric music apparatus 1 or the server 5.

The present invention has been described in connection with the preferred embodiments. The invention is not limited only to the above embodiments. It is apparent that various modifications, improvements, combinations, and the like can be made by those skilled in the art.

WHAT ARE CLAIMED ARE:

1. An electric music apparatus, comprising:

an interface to which an external device can connect;

a memory that stores device drivers for operating external devices connected to the interface and a list recording relationship between external devices and corresponding device drivers;

an obtaining device that obtains external device information from an external device connected to the interface;

a searching device that searches a device driver corresponding to the connected external device with reference to the list in accordance with the obtained external device information;

a first determining device that determines to control the connected external device by using the device driver corresponding to the connected external device when the searching device detects the corresponding device driver from the list;

a second determining device that executes a specific command to the connected external device by using a device driver stored in the memory when the searching device cannot detect the corresponding device driver and determines to control the connected external device by using a device driver that makes the external device respond to the specific command without an error;

a judging device that judges the connected external device as an unsupported device when the connected external device does not respond to the specific command executed by the second determining device with all the device

drivers stored in the memory without an error;

wherein the judging device judges with reference to the list whether the connected external device is an unsupported device without executing any command after once the judging device judges that the external device is an unsupported device, and

wherein the memory further stores a result of the judgment by the judging device when the connected external device is judged as an unsupported device.

2. The electric music apparatus according to claim 1, wherein the memory further stores a relationship between the connected external device and the device driver determined to be used by the second determining device.

3. The electric music apparatus according to claim 1, wherein the result of the judgment that the external device is an unsupported device is stored in the list.

4. The electric music apparatus according to claim 1, wherein the second determining device uses every device driver stored in the memory to execute the specific command.

5. A program executed by an electric music apparatus comprising an interface to which an external device can connect, and a memory that stores device drivers for operating external devices connected to the interface and a list

recording relationship between external devices and corresponding device drivers, the program comprising the instructions for:

(a) obtaining external device information from an external device connected to the interface;

(b) searching a device driver corresponding to the connected external device with reference to the list in accordance with the obtained external device information;

(c) determining to control the connected external device by using the device driver corresponding to the connected external device when the searching instruction (b) detects the corresponding device driver from the list;

(d) executing a specific command to the connected external device by using a device driver stored in the memory when the searching instruction (b) cannot detect the corresponding device driver and determining to control the connected external device by using a device driver that makes the external device respond to the specific command without an error; and

(e) judging the connected external device as an unsupported device when the connected external device does not respond to the specific command executed by the instruction (d) with all the device drivers stored in the memory without an error;

wherein the instruction (e) judges with reference to the list whether the connected external device is an unsupported device without executing any command after once the instruction (e) judges that the external device is an unsupported device, and wherein the memory further stores a result of the judgement by the

instruction (e) when the connected external device is judged as an unsupported device.

6. The program according to claim 5, wherein the instruction (d) uses every device driver stored in the memory to execute the specific command.

7. An installing method of an external device for an electric music apparatus comprising an interface to which an external device can connect, and a memory that stores device drivers for operating external devices connected to the interface and a list recording relationship between external devices and corresponding device drivers, the method comprising the steps of:

(a) obtaining external device information from an external device connected to the interface;

(b) searching a device driver corresponding to the connected external device with reference to the list in accordance with the obtained external device information;

(c) determining to control the connected external device by using the device driver corresponding to the connected external device when the searching step (b) detects the corresponding device driver from the list;

(d) executing a specific command to the connected external device by using a device driver stored in the memory when the searching step (b) cannot detect the corresponding device driver and determining to control the connected external device by using a device driver that makes the external device respond

to the specific command without an error; and

(e) judging the connected external device as an unsupported device when the connected external device does not respond to the specific command executed by the step (d) with all the device drivers stored in the memory without an error;

wherein the step (e) judges with reference to the list whether the connected external device is an unsupported device without executing any command after once the step (e) judges that the external device is an unsupported device, and wherein the memory further stores a result of the judgement by the step (e) when the connected external device is judged as an unsupported device.

8. The method according to claim 7, wherein the step (d) uses every device driver stored in the memory to execute the specific command.

9. The program according to claim 5, wherein the result of the judgement that the external device is an unsupported device is stored in the list.

10. The method according to claim 7, wherein the result of the judgement that the external device is an unsupported device is stored in the list.