Biomedical Engineering, Vol. 34, No. 4, 2000, pp. 167-172. Translated from Meditsinskaya Tekhnika, Vol. 34, No. 4, 2000, pp. 3-8. Original article submitted November 25, 1999.

Research, Design, and Technology

Use of the USB Universal Serial Bus in Computer Medical Systems

E. V. Vetvitskii, A. V. Plotnikov, D. A. Prilutskii, and S. V. Selishchev

Personal computers (PC) are now the most readily available multimedia devices. They are widely used in various medical systems (Fig. 1). Personal computers can be used for processing, storage, and display of medical information.

The device–PC interface is an important component of any computer-assisted medical system. Either external PC ports or special integral adapters can be used for this purpose. Special integral adapters are usually developed if standard communication channels are ineffective because of high rate of information exchange, sophisticated configuration of system, etc.

Specifications of external interfaces of modern PCs are given in Table 1. Although IEEE 1394 (FireWire) is the most effective external interface, its practical use is presently rather difficult because of lack of sufficient information and experience.

The Universal Serial Bus (USB) is an interface designed to connect peripheral devices. This interface is available from Compaq, DEC, IBM, Intel, Microsoft, NEC, and Northern Telecom. The first USB Standard was adopted in 1996. A modified version of the USB Standard was developed recently [7].

The goal of this work was to describe the USB interface and some medical computer systems implemented on its basis.

The USB Universal Serial Bus

The USB provides two rates of data transmission: 12 and 1.5 Mbyte/sec. This interface is based on tiered-star topology. According to this topology, only one device can be connected to the PC, whereas other devices are connected to the USB concentrator (hub). The USB concentrator can be made as an individual device or as an integral part of the PC monitor, keyboard, or printer. This allows as many as 127 USB devices to be connected simultaneously. Because the USB controller requires only one interrupt regardless of the number of connected devices, the problem of insufficient resources of inner PC buses is thought to be solved in this case.

The USB supports the universal autoconfiguration standard Plug and Play [4]. This standard completely determines the process of installation of new peripheral devices. Newly connected devices are identified automatically, including selection of IRQ parameters, DMA channels, and input/output addresses. No extension cards or special drivers are required in most cases.

The USB connector provides a +5 V line sufficient to power the majority of medical devices. Therefore, individual power sources are not required.

Because extension cards are not required, the process of development of new peripheral devices becomes significantly simpler. Because of the virtually universal compatibility of the USB, the volume and cost of test-

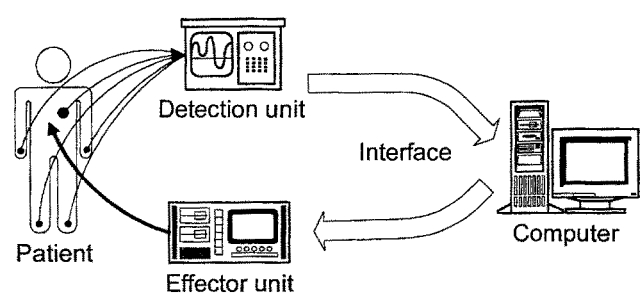

Fig. 1. Medical system based on a personal computer.

Moscow State Institute of Electronic Engineering, Zelenograd, Russia; E-mail: pdima@mcs.miee.ru

TABLE 1. Specifications of External PC Interfaces

Interface	Data transmission rate	Number of wires in cable	Power source	Maximum connection length, m	Number of devices	Level of implementation
Serial (RS232-C)	Up to 11.5 kbyte/sec	2-9	No	Up to 15	1	Physical
Parallel (IEEE 1284)	Up to 300 kbyte/sec	8-25	No	Up to 3	1	Physical
USB	Up to 1.2 Mbyte/sec	4	5 V, up to 0.5 A	Up to 5	Up to 127	Physical, logical, network
FireWire (IEEE 1394)	Up to 40 Mbyte/sec	6	8-40 V, up to 1.5 A	Up to 4.5	Up to 63	The same

ing of various hardware and software combinations can be significantly reduced [1].

Taking into account that the demand in high-efficiency peripheral devices is continuously growing, a number of manufacturers advertise a new version of the USB Standard (USB v. 2.0), which will be commercially available in late 2001 [5, 6]. An increase in the clock frequency from 12 to 360-480 MHz allows this system to serve 30-40 times more peripheral devices than previous version. This standard will be compatible with all devices supported by USB v. 1.1.

USB Topology

The USB Standard describes two types of USB devices:

- main unit (host);
- peripheral devices (PD).

At the logical level the system has direct connections between host and all PD (direct information exchange between host and periphery). At the physical level, the USB structure is based on tiered-star topology (Fig. 2).

USB Host

Any USB system contains only one host. Usually, this host is equipped with an integral root hub. Modern PC models have a two-port USB. The host controller should provide the following functions:

- control of host–PD service packages;
- control of host–PD information exchange;
- power supply of connected PD.

The system software of the USB host implements the following functions:

- configuration and numeration of PD;
- synchronous and asynchronous data transfer;

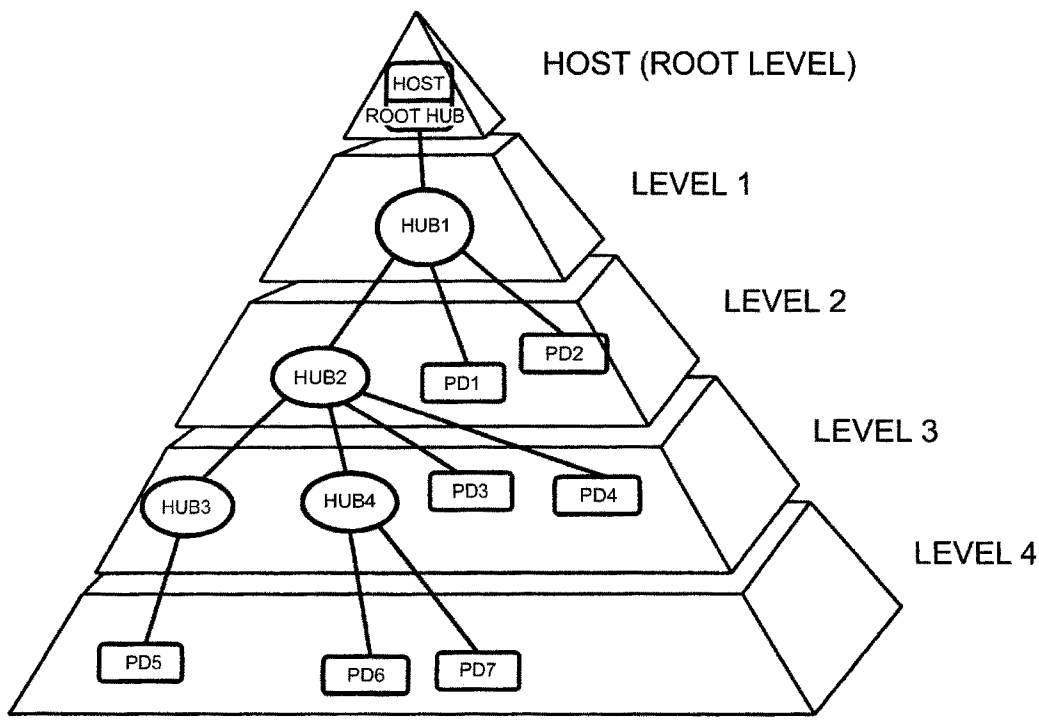

Fig. 2. USB topology.

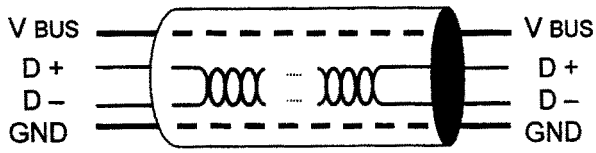

Fig. 3. Electrical cable of the USB.

– power supply of devices connected to bus.

Various functional devices and the USB concentrator (hub) are USB PDs.

Electrical Connection of the USB

A 4-wire cable (Fig. 3) is used for connecting the USB. Two types of connection are available: a shielded cable (length up to 5 m) designed to provide high-rate connection (12 Mbaud) and a simple cable (length up to 3 m) designed to provide low-rate connection (1.5 Mbaud). Two signal wires form a differential pair with sensitivity of 200 mV. The NRZI (non-return-to-zero, invert) coding with logical levels of 3.3 V is used for data transmission. A terminator at the end of the cable identifies the presence and type (high-rate, low-rate) of the devices connected to the bus.

The cable shown in Fig. 3 provides a power line with voltage of +5 V and maximum current of 500 mA per PD. The normal working mode of PDs with working current of more than 100 mA is achieved only after configuration. A suspend mode is supported for devices with maximum current of 500 μA.

Mechanical Connection of the USB

Two types of connectors are available (Fig. 4): type A (host or hub) and type B (hub or PD).

Accidental formation of a closed loop in the bus is avoided, because type A and type B connectors are different (not interchangeable).

USB Exchange Protocol

Any exchange procedure mediated by the USB includes three stages.

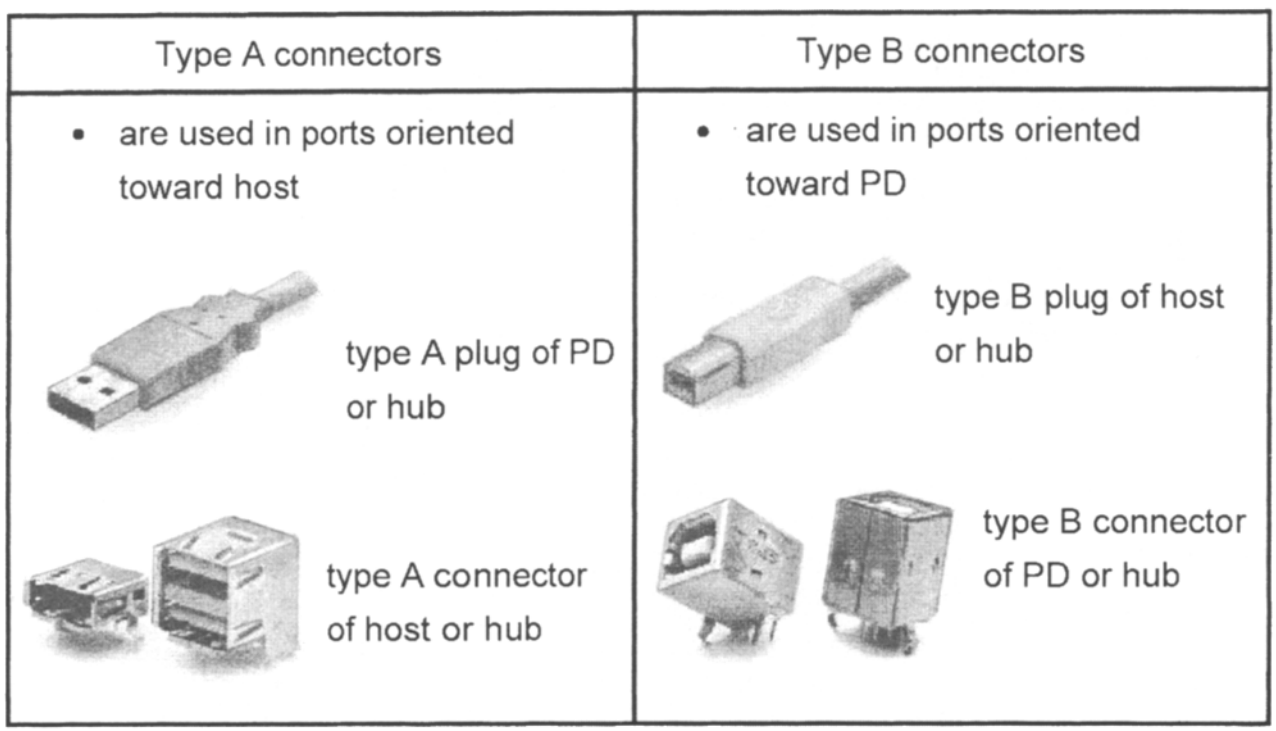

Fig. 4. Mechanical USB connectors.

1. The host-controller transmits a service package that determines the type and direction of exchange, address, and PD unit number.

2. The device (host or PD) transmits data or a package determining the absence of transmitted data.

3. Exchange is usually terminated by the package from the receiver (PD or host), which confirms the correctness of the exchange.

Any PD has several endpoints with individual exchange protocols. USB-mediated data exchange between PD and host proceeds through a so-called pipe channel. There are two types of such channels: stream and message. These types differ in data package structure, frequency bandwidth, protocol, and PD characteristics (exchange direction and data buffer size). All channels (except zero) become available after PD initialization. The zero channel is used for PD control and configuration; it is always available.

Noise Resistance of the USB

Noise resistance of USB-mediated data exchange is provided by reliable hardware (differential transceiver, protective shielding of cables, redundancy check of control command and data fields, autodetection of PD connection/disconnection to bus, protocol protection using time-out procedure, independent data exchange between individual channels, functions, etc.).

The cyclic redundancy check (CRC) control procedure is applied to control and data command fields of each package. CRC-codes detect error or corruption of 1 or 2 bits. Transmission of the corrupt block identified by the protocol is repeated. In case of persistent error, the host-controller repeats the PD request up to three times. If the error does not disappear, the host-controller sends an error message to the system program.

System Configuration in the USB

The USB provides system connection/disconnection to the bus at any required time. The corresponding dynamic procedure should continuously monitor the physical configuration of the USB.

Connection of PD to USB. Any PD is connected to the USB through a USB concentrator (hub). The hubs connected to the bus are continuously polled by the host. If a new device is connected to the bus, the corresponding hub port is initiated and a standard USB default address is attributed to the hub. This procedure of addressing of any USB device is used after connection or resetting. The zero channel is used for PD control and configuration. The following information is transmitted from PD to host:

- USB Standard version supported by the device;
- class (subclass) of the device;
- standard exchange protocols supported by the device;
- manufacturer, catalog number, serial number;
- number of interfaces and their characteristics;
- power requirements.

If the newly connected device is a hub connected to another PD, the procedure is repeated individually for each device.

Disconnection of the PD from the USB. If a PD is disconnected from a hub port, the hub automatically disengages the port and sends a corresponding message to the host. The host deletes the information about the device from the system. If the disconnected device is a hub connected to other PDs, all corresponding device drivers are deleted.

USB Data Transfer Protocols

Data exchange between the host system software and a PD is mediated by the USB. Each PD may have several communication channels. Some of these channels can be used for data transfer from host to device, whereas the other channels are for data transmission in the opposite direction (Fig. 5). Each channel is characterized by its own data transfer protocol. USB supports data transfer protocols of four types:

- control transfer protocol;
- bulk transfer protocol;
- interrupt transfer protocol;
- asynchronous transfer protocol.

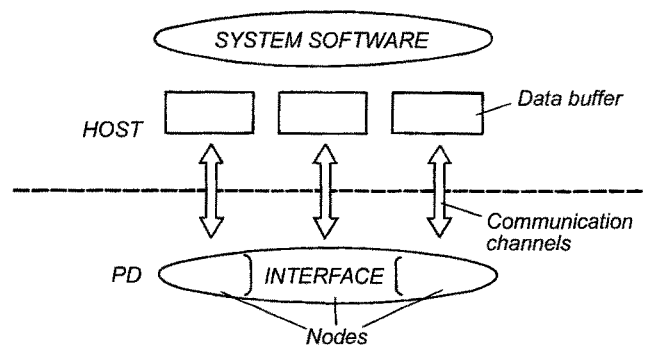

Fig. 5. An example of USB data transfer flows.

Control Transfer Protocol. Control transfer protocol is used for configuring a new PD and for transmitting special control commands. It has the maximum degree of noise protection.

Bulk Transfer Protocol. Characteristics:

- neither frequency bandwidth nor access time are guaranteed;
- enhanced noise protection;
- can be used only by high-efficiency devices.

This protocol should be used for transmitting relatively small volumes of data. Also, the time requirements for data transmission should not be too demanding. According to the bus operation, the host initiates data transmission at a convenient time. The correctness of the data transmission is tested and guaranteed. The data field size is 64 bytes. The data transfer rate is up to 1.2 Mbyte/sec (if the number of devices connected to the bus is not very high).

Interrupt Transfer Protocol. Characteristics:

- access time is guaranteed;
- enhanced noise protection;
- can be used by devices of any type.

This protocol should be used for strictly periodic transmission of relatively small volumes of data. Data are transmitted only from host to PD (the PD may generate interrupt commands). The maximum data field size in high-efficiency (data transfer rate up to 64 kbyte/sec) or low-efficiency (data transfer rate up to 8 kbyte/sec) devices is 64 or 8 bytes, respectively. PD interrogation time ranges from 1 to 255 msec. If a received package contains an error, its receipt is not confirmed, and data transmission is repeated.

Asynchronous Transfer Protocol. Characteristics:

- maximum rate of data transfer;
- frequency bandpath is guaranteed;
- can be used only by high-efficiency devices.

This protocol should be used for transmitting large volumes of data. The data field size is 1023 byte. The data transfer rate is up to 1.2 Mbyte/sec. Upon initial configuration of a PD, the host determines the data package size (with 1 msec interval between packages), and bus overload is continuously monitored throughout the whole procedure of data transfer. In contrast to other protocols, the data receipt is not confirmed. Therefore, some packages may be lost.

Development of USB-Based Devices

USB Support by Peripheral Devices. The protocols listed above have rather elaborate structure, and the data coding system is rather sophisticated. This gave rise to development of special USB controllers. These controllers are sophisticated high technology devices designed to facilitate construction of new devices and low-level computer bus support (alignment of signal levels, data encoding/decoding, verification of CDC-code, addressing, and data buffering). The controller can be integrated with an external processor into a high-level system. Single-chip microprocessors with a USB port can also be used. Standard debugging plates are available from Cypress, Intel, National Semiconductor, Philips, and Siemens. These plates significantly facilitate construction of new devices.

USB Software. USB software consists of two main parts:

- device software (USB support of single-chip microprocessor);
- driver operating system of PC.

The device software is virtually independent of the PC operating system type, whereas drivers should be adapted to the operating system used in a given PC. USB is presently available for Windows 98/NTS, Mac, and Linux.

Usually, device software is based on the programs provided by manufacturers of USB controllers. The code volume is 1000-2000 strings written in a high-level language (C). About 30% of the microprocessor capacity is used to run this program. Special software is used to test the device compatibility.

Special software (the Driver Development Kit) is used for developing drivers for Windows. The Driver Development Kit is a set of libraries and examples of drivers, including USB exchange protocols. Because Windows does not contain a universal USB driver, a special driver should be developed for each type of PD. Auxiliary software available from other companies (Blue Water Systems, Video Software) can be used to solve this problem with less effort.

Development of USB-Based Medical Systems. Digital algorithms of data processing and microprocessors are widely used in modern medical devices. A typical diagram of a medical system based on a personal computer is shown in Fig. 1. This system consists of the following units.

1. Effector unit, detection unit.
2. Control unit and data processing unit (microprocessor).
3. Digital data transfer channel.

A finished medical device composed of units (1) and (2) can be supplemented with a channel for communication with the PC. This can be implemented integration with a special microprocessor-assisted USB controller. Although the integration modifies the system

Fig. 6. General view of the NV-40U 40-channel computer-assisted electroencephalograph.

only slightly, the system becomes compatible with the USB Standard. Therefore, USB can be implemented as an inexpensive and high-efficiency interface.

It is recommended to develop new medical systems base on a single-chip microprocessor equipped with an integral USB port.

The USB interface is recommended in the following cases:

– for complicated functional structures (several independent data flows with different requirements for exchange protocols);

– in medical systems containing several devices (up to 127) connected to one PC;

– for flexible configuration (rapid connection and disconnection of peripherals);

– data transfer rate, up to 1.2 Mbyte/sec;

– consumed power, 0.05-2.5 W.

An Example of Implementation of a USB-Based Device. A number of USB-based devices have been developed at the Department of Biomedical Systems, Moscow State Institute of Electronic Engineering. A 40-channel computer-assisted electroencephalograph (Fig. 6) is an example of these devices.

The NV-40U 40-channel computer-assisted electroencephalograph has the following specifications:

– number of channels: 40 unipolar or 39 differential;

– sampling frequency, 512 kHz (synchronously in all channels); digital filtration (sigma–delta analog-to-digital converters [2, 3] are used);

– data output frequency, up to 2 kHz per channel (24-bit signals);

– frequency bandwidth, 0-500 Hz (at the level of 3 dB);

– signal dynamic range, 117 dB;

– input resistance, no less than 90 MΩ;

– computer connection, USB port.

A special USBN9602 controller (National Semiconductor) provides USB operations. This controller is controlled by an Atmel AT90S8515 single-chip microprocessor (8-bit controller with RISC-architecture).

This electroencephalograph is an example of medical device integration with a PC mediated by the USB interface. The USB interface offers the following advantages: high rate of data transfer, direct power supply, easy connection, and relatively low cost.

The USB interface can be used to combine modern medical devices with PCs and provides effective channels of data transfer. The advantages of the USB interface facilitate the development of high-quality medical devices.

REFERENCES

1. "Components of Intel Universal Serial Bus (USB)", Internet: http://www.intel.ru/design/usb/.
2. S. F. Kurikov, D. A. Prilutskii, and S. V. Selishchev, Med. Tekh., No. 4, 7-10 (1997).
3. M. Curtin, Analog. Dialog. J., **28**, No. 2, 6-8 (1994).
4. PC 98 Hardware Design Guide. Microsoft Corporation. Washington (1998).
5. The PCMCIA Developer's Guide. Sycard Technology. Sunnyvale (1994).
6. "The USB Implementers Forum", Internet: http://www.usb.org/.
7. "Universal Serial Bus Specification. Revision 1.1", Internet: http://www.usb.org/developers/index.html.