XVII IMEKO World Congress
Metrology in the 3rd Millennium
June 22–27, 2003, Dubrovnik, Croatia

WILL USB BE A STANDARD OF PC - INSTRUMENTATION COMMUNICATION ?

Tomasz Twardowski

Measurement Division, AGH Technical University, Cracow, Poland

Abstract – The goal of this paper is to provide analysis of Universal Serial Bus, multipurpose communication interface, in application to laboratory measurement systems. This interface is compared with others, established measurement interface standards, especially IEEE-488, in a view of requirements in this specific application. The current market offer and trends in software and hardware design are reviewed. Obstacles to common use of USB in measurements are pointed out and projections to the future use are made.

Keywords: measurement systems, communication interfaces.

1. INTRODUCTION

Even small laboratory measurement systems consist of several instruments, which shall cooperate. The supervisor's role in laboratory conditions is most often played by a multipurpose PC microcomputer. Equipped with a communication interface to the bus shared with instruments it can serve as a task dispatcher, controller and coordinator. Because of speed issues the most often used standard of communication interface has been measurement oriented parallel 8-bit IEEE-488, also known as HP-IB or GP-IB. Its latest version High Speed GPIB (HS488) has data rate enhanced by loosening data reception acknowledgement rules. Between two devices and 2 m of cable, HS488 can transfer data up to 8 MB/s. For a fully loaded system with 15 devices and 15 m of cable, HS488 transfer rates can reach 1.5 MB/s.

IEEE-488 bus controller is a plug-in card, an additional instrumentation required to integrate measurement system, rising the overall cost. The popular alternative is to use standard multipurpose I/O interface available on every PC, such as serial RS-232 or parallel printer port. Especially the latest solution has been used as a cheap and fast (up to 500kB/s) one, e.g. in Iotech's DaqBook [1]. However most advanced modifications of the original Standard Parallel Port (SPP) specification towards open multi-device bus with addressing and transfer rate negotiation (ECP) do not find response in measurement instrumentation manufacturers society.

This fact has a partial explanation in development of fast general purpose serial interfaces, Universal Serial Bus and IEEE 1394 (also known as FireWire) to be used as a basic input/output bus on a PC platform for speed demanding multimedia applications [2].

2. KEY USB FEATURES

Universal Serial Bus [3] is designed to serve communication with low and medium speed devices, such as printers, external CD drives and sound devices. The most important feature of USB contributing to its widespread use is the presence of a USB controller in every new PC. The controller circuit is accompanied by a low level operating system driver and higher level drivers handling standard devices. These are supplied by all popular operating systems including MS Windows, Linux and MacOS. Unfortunately measurement instrumentation is not standardized as yet, i.e. there is no widely recognized profile of such a device. The first attempts towards standarization are described in the following. The task of providing specialized software driver set on top of low level USB driver is, as yet, a duty of a device manufacturer.

Development of USB is correlated with development of new ideas in PC structure and its forecasted application. PC evolves towards a sealed-box machine processing massive information (as live sound and video) for non-expert. Two techniques inherent to USB support this development: Hot Swap and Plug & Play. First one allows connecting and disconnecting a device without switching off PC and the second one relieves the user of installing software handling connected device. These features are complemented by power supply to peripherals available directly from the bus.

The basic concept of USB bus is a master-slave architecture, with every transmission initiated by the host. While still preserving this rule, USB is changing. It started in 1995 with 1.5 (Low Speed) and 12 Mbits/s (Full Speed) data rate, and was refined in 1998 with revision 1.1. In year 2000 USB revision 2.0 appeared with 480 Mbits/s data rate (High Speed). At every bus speed it offers three modes of data transfer, apart from control mode, i.e. interrupt, bulk and isochronous. Interrupt mode is provided for transferring small amounts of data on a host controlled regular basis. Bulk transfers are suitable for large amounts of data when consistency, guaranteed by automatic retransmission of erroneous data, is a concern. Isochronous transfer is designed for time critical applications, as it assures constant bandwidth but without handshaking. Control transfers are used to configure and send commands to a device.

3. COMPARISON OF USB AND IEEE-488

USB is not a dedicated measurement interface, as IEEE-488 is. Thus it lacks strictly measurement oriented features, such as group trigger of multiple instruments which is present in IEEE-488. However these features can be realised either in hardware, or on top of USB protocol.

The features of USB supported by operating system, i.e. Hot Swap and Plug&Play abilities allow construction of portable measurement systems, easy to connect and reconfigure. Possibility of supplying power to measurement instrument directly from PC power source enables handling in-field data acquisition tasks [4].

Comparison of data transfer rate between USB 1.1 and HS488 is in favour of the latest, however comparison with USB revision 2.0 shows the strength of serial bus. Not only speed is critical in this comparison. USB allows 3 different data transfer modes suitable to different classes of devices. Among them so called isochronous data transfer with guaranteed bandwidth is useful for continuous acquisition.

USB controller can serve up to 127 peripheral devices while IEEE-488 serves up to 15 devices. Simple static addressing scheme of peripherals used by IEEE-488 is practical for measurement systems put together for longer time. USB uses more sophisticated approach of dynamic bus enumeration, i.e. assigning addresses after (dis-)connecting any device. Operating system supporting Plug&Play takes the responsibility of loading and initialising proper driver for every hot-swapped device.

To conclude above comparison, despite the fact that IEEE-488 works and is in widespread use, there are more comfortable and faster solutions.

4. PREDICTIONS TO THE FUTURE

According to Keithley's survey [5], current state of interface use in measurements shows dominating position of RS-232. That is easily explained by its accessibility on the rear of every PC. IEEE-488 and 4-20mA (used mainly in industry) are slowly loosing users. USB and PCI are growing.

TABLE I. Communication buses and its data rate

Bus	Maximum data rate
RS-232	15 kB/s (115 kbits/s)
EPP	500 kB/s
IEEE-488	8 MB/s
Ethernet	12 MB/s (100 Mbits/s)
USB 2.0 (High Speed)	60 MB/s (480 Mbits/s)
FireWire-2 (scalability limit)	400 MB/s (3.2 Gbits/s)
PCI-X 2.0 spec. (internal bus)	4.3 GB/s

Comparing buses speeds contained in Table I, it is evident that RS-232 will be replaced by USB, equally accessible and with additional helpful features. PCI internal computer bus (parallel) is much faster than serial interfaces, so plug-in data acquisition cards will not be replaced by external measurement devices in speed demanding applications. Even despite the noisy environment inside PC. Nowadays popular DAQ plug-in cards commonly handle over 1 Msample/s continuous 12-bit data stream. Need for higher resolution requires use of external measurement devices, so PCI plug-in cards will coexist with USB/Ethernet/FireWire equipped instruments.

4. MARKET OFFER

For a long time the only choice on the market of USB devices related to measurement were USB – IEEE-488 converters. First sampling devices with USB 1.0 compatible interface controllers offered only very slow sampling not exceeding 100 samples/s. After a period of growing recognition and mastering of USB bus there appear more measurement instruments suited to bus capabilities. As a sample of market offer two instruments have been chosen as representatives of changing approach to USB data acquisition. These are IOtech's Personal Daq and ADLINK's USBDAQ-9100-MS.

TABLE II. Representative USB-equipped instruments

Property	Personal Daq	USBDAQ
Resolution	22 bits	12 bits
maximum sampling rate	80 samples/s	500 Ksamples/s
continuous sampling rate	80 samples/s	100 Ksamples/s
analog inputs	5Diff, 10SE	8Diff
analog outputs	No	2
digital I/O, timer/counter	8DIO, T/C	8DIO, T/C
USB bus powered	yes	yes
simultaneous sampling	no	yes (4 channels)

As shown in Table II, IOtech's product is a high resolution standard A/D + frequency/pulse + digital I/O instrument equipped with new interface. 22-bit resolution is high in comparison with PC plug-in cards, so the effect of lower electromagnetic noise outside PC case is visible. The sampling rate is slow compared to bus capabilities, but partially explained by high resolution.

ADLINK's product is a typical reasonable resolution DAQ instrument with well used bus bandwidth. Interesting options are simultaneous sampling of 4 channels and 100 Ksamples/s real time operation. This device conforms to USB 1.1 specification with maximum data rate of 1,5MB/s.

As yet, there are no instruments on the market with interface conforming to 2.0 specification. It is expected that with the dissemination of USB 2.0 standard and growing availability of new design controllers much faster instruments will be available.

5. BUILDING USB MEASUREMENT DEVICE – HARDWARE

The simplest connection of a measurement device equipped with IEEE-488, RS232 or EPP to USB bus is by use of a switch changing protocol. This is a temporary solution for laboratories equipped with older bus instrumentation and for occasional "in field" laptop driven measurements. One cannot expect transfer rate boost in this case - the slower bus is a transmission bottleneck. However, this solution can be cheaper than using plug-in bus controller: compare for example National Instruments' PCMCIA IEEE-488 controller and their USB - IEEE-488 switch.

Measurement device capable of transmitting packets over a USB bus must follow a strict set of electrical and protocol rules. Handling the task of packet transmission/reception at high speeds requires hardware solutions. There exist a broad range of micro-controllers, equipped with USB core, covering low level protocol details via USB Serial Interface Engine. Grace to it data transmission on high level is managed with simple input/output queue reading/writing. An example of a general purpose microprocessor equipped with USB core (e.g. Cypress EZ-USB 8051 compatible, USB 1.1) is shown in figure 1.

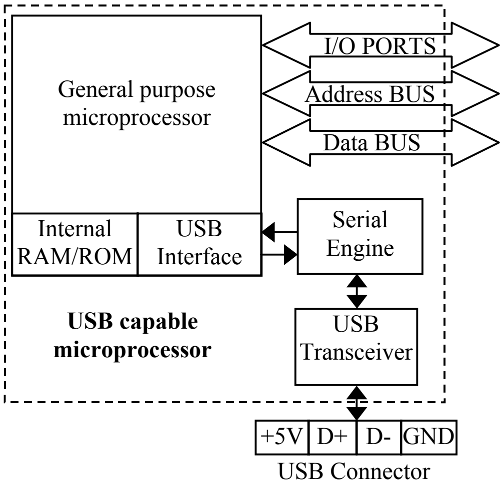

Fig. 1. Example of USB capable general-purpose microprocessor

Interfacing with measurement electronic circuits can be realised with data/address buses or through input/output ports. Data flow in low transfer speed applications are easily handled with software or interrupt control. Higher speeds require DMA service with writing to a USB FIFO directly by a measurement circuit. This case excludes data preprocessing on device's side. Specific application of DMA service is a USB streaming controller, such as Texas Instruments TUSB3200A (8052 compatible, USB 1.1), employing a Codec Port (in the mentioned example AC'97 and I^2S standards) as a source and destination of isochronous data pipe.

6. BUILDING USB MEASUREMENT DEVICE – SOFTWARE

Communication of user's application with an USB device is possible when this device is seen on a bus. Detection of configuration changes on a bus is performed on behalf of a pull-up resistors turned on in plugged devices, indicating device transfer speed as well. Every resistor pull-up initiates process of enumeration, i.e. assigning device address and gathering information (general descriptors) about device class or vendor/product IDs. Up to this point no dedicated, device specific software is involved. All actions are performed on low level of operating system, in Microsoft Windows by Host Controller and USBD drivers. After the device is identified by its class (or if it is a specific device by its VID/PID), establishing USB connection proceeds with loading appropriate software driver for the device. The latter performs device-specific initialization. Figure 2 shows relations of described USB system software in specific application to outer world observation (i.e. measurement).

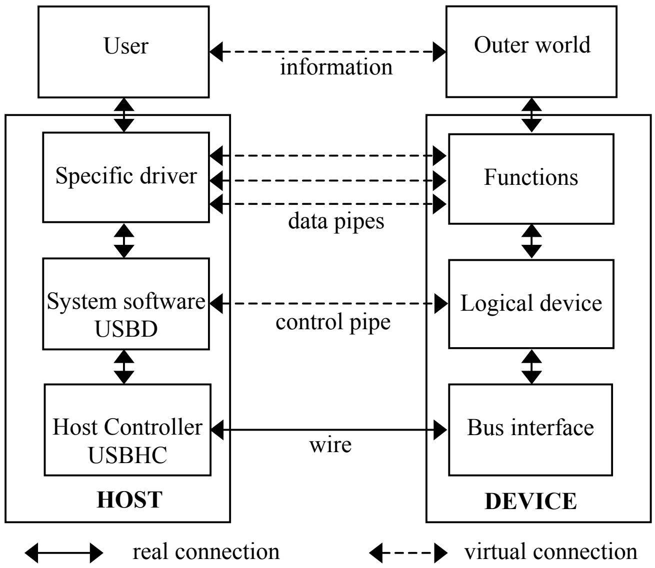

Fig. 2. Information flow through USB Stack

The main problem of programming USB device is to provide a device specific driver on host's side. If the device belongs to a group of popular equipment, there certainly exist an associated class driver. However it is not the case with measurement instrumentation, as yet. Writing one's own operating system driver is a complicated and time consuming task, even when adapting existing software.

As USB devices with data requirements outside the range of defined classes must provide their own class specifications and drivers as defined by the USB Specification, a reasonable choice is to adapt device's interface to existing class driver specification. Specialized class of devices is USBTMC Test and Measurement described in more detail in the next section.

Simple to create interface to the device is provided by a general Human Interface Device (HID) class driver [6]. It offers transfer rates of 800 B/s for low speed, 64 kB/s for full speed and 23.4 MB/s for high speed devices, exclusively in interrupt mode. Because there is inherent HID class driver in MS Windows and Linux, the required work is to supply proper descriptors and device's control code [7]. Software on device's side has simpler structure and is aided by controller's firmware. In a typical case USB capable controller offers a collection of input and output FIFO queues and possibility to specify VID/PID and device class (see previous section for examples).

7. USB TEST AND MEASUREMENT DEVICE CLASS

In January 2003 there appeared a novel specification of measurement device class contributed by major instrumentation manufacturers, including National Instruments, Agilent (former HP), Rohde & Schwartz and Tektronix. It seems to be the first step towards uniform standard of USB instrument access. The specification concerns only very general rules of interconnection, defining input and output data and control streams. The definition of Host API's for communication with USBTMC interfaces is outside the scope of this specification. USBTMC client software API's and any other specifications needed to achieve USBTMC interoperability will be documented in a future VISA standard specification [8].

As shown in figure 3, the new regulation imposes bulk transfer of measurement data and commands, i.e. high-performance guaranteed delivery transfer type.

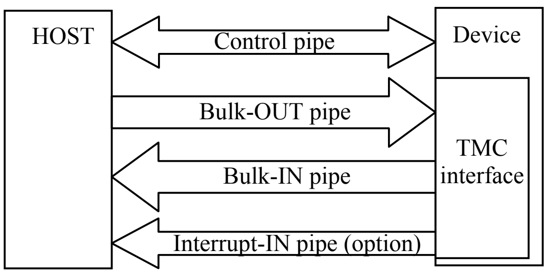

Fig. 3. USB TMC host-device communication model

The control pipe is required by the USB 2.0 specification and is used to send standard, class, and vendor-specific requests to the device. The Bulk-OUT pipe is required and is used to provide a delivery data path from the Host to the device. The Host must use the Bulk-OUT pipe to send USBTMC command messages to the device, and the device must process the USBTMC command messages in the order they are received. The Host must also use the Bulk-OUT pipe to set up all transfers on the Bulk-IN endpoint. The Bulk-IN pipe is required and is used to provide a data path from the device to the Host. The Host must use the Bulk-IN pipe to receive USBTMC response messages from the device. The Interrupt-IN pipe is used by the device to send notifications to the Host, and may be defined as required by TMC subclass. The Host USBTMC driver may optionally support additional pipes if these are required by a USBTMC subclass specification.

The primary goal of the specification is to fix standard headers and define flow control of TMC messages. There are few measurement specific elements in this document. Much more specific will be future subclass definitions. First of them just appeared: specification of SCPI communication over USB, called USBTMC-USB488 subclass specification [9]. There, the role of Interrupt-IN is strictly defined to be the pipe of SRQ requests and status responses. Device capabilities description sent over Control endpoint declares specific IEEE-488 abilities of a device, like REN_CONTROL, GO_TO_LOCAL, LOCAL_LOCKOUT. Devices with a 488.2 conformant USB488 interfaces must support the Message Exchange Protocol (MEP) [10].

Some of the IEEE-488 commands cannot be directly realised in USB, because the latter does not provide broadcast mechanism. IFC, REN, Device Clear, Local Lockout, SDC and GET (Group Execute Trigger) must be realised individually per device, what is not identical in effect. It should be also stressed that on USB only one device is a Master (usually PC host). However there is work on Master's role negotiating protocol (USB On-The-Go).

8. CONCLUSIONS

The main restraining force in USB spread seems to be a huge amount of useful measurement instrumentation with IEEE-488 interface and, related with that, inertia of its users' habits. These instruments will be in use for a long time, but USB has increasing popularity in manufacturers as well as in users society. It is expected that growing speed of future USB 2.0 conforming instruments will be accompanied with more sophisticated capabilities realised by use of onboard re-programmable microprocessors. The development of specialized measurement device drivers for USB bus will make the programming task of accessing USB device much simpler. The latest USBTMC-USB488 class specification joins traditional SCPI instrument control with USB capabilities. There still is the lack of unified Application Programming Interface to TCM class drivers, but one should emerge soon in a new VISA specification.

REFERENCES

[1] *IOtech PC-Based Data Acquisition & Instrumentation Catalog 2003*.
[2] A. Tracht, "Connecting With the New Universal Serial Bus", *Evaluation Engineering*, vol. 7, 1998.
[3] "Universal Serial Bus rev. 2.0 specification", December 2000, available at the web site www.usb.org.
[4] J.P. Keithley, "Customization is key to future instrumentation", *IEEE Spectrum*, vol. 36, no. 1, 1999.
[5] "Needs for Speed and Resolution Jump", Keithley Instruments survey, 1999, available at the web site www.keithley.com.
[6] "Device Class Definition for Human Interface Devices (HID)", 2001, available at the web site www.usb.org.
[7] S. Allman, "Using the HID class eases the job of writing USB device drivers", Electronic Design News, September 2002, available at the web site www.edn.com.
[8] "Universal Serial Bus Test and Measurement Class Specification (USBTMC)", January 2003, available at the web site www.usb.org.
[9] "Universal Serial Bus Test and Measurement Class, Subclass USB488 Specification (USBTMC-USB488)", January 2003, available at the web site www.usb.org.
[10] "IEEE Standard Codes, Formats, Protocols, and Common Commands", IEEE Std 488.2, 1992, available at the web site www.ieee.org.

Author: dr inż. Tomasz Twardowski, Measurement Division of AGH Technical University, al. Mickiewicza 30, 30-059 Cracow, Poland, ttward@agh.edu.pl