ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2008. No. 4(84)
ELEKTRONIKA IR ELEKTROTECHNIKA

T170 ELECTRONICS
ELEKTRONIKA

Delay Evaluation in Interactive Exchange with Controllable Devices for Some Types of Computer Interfaces

E. Boole, D. Stepin, V. Stepin
Institute of Electronics and Computer Science
Dzerbenes st. 14, LV-1006 Riga, Latvia; phone: +371 7554500; e-mail: buls@edi.lv

Introduction

Twenty years ago the National Instruments presented the concept of virtual instrumentation[1]. During these years it was improved and approved. Now there is a great amount of measurement and control instrumentation which fits in this concept. The virtual instrument consists of standard PC, equipped with an application software, and hardware module(s). All this together performs a monitoring of some production line or process, controls all necessary parameters and manages the process flow. As a rule the included PC executes not only this process control, but also gathered data processing, simulation and analyze, interaction with a personal, exchanging via network with other PCs, involved in general task solving. Considering such various PC employments there is a problem of real-time interaction with controlled devices through the standard external PC interfaces. Manufacturers define the physical interface characteristics, but in the interaction process it is essential to know a real interface performance including all possible delays, appearing by reason of shared programs execution and a continuous process.

Interface performance

In spite of existing differences between the standard PC interfaces there are two main parameters, which characterize the interface performance for PC and hardware interactions:

- PC/HW response time for randomly occurring events;
- input/output throughput for a great data volume.

The second parameter can be defined by a simple *Timer()* function, included in the application program. This function in usual operation mode will give an adequate accuracy for time of great data array transferring or receiving. But the response time determination is more complex problem, because this time is comparable or in most cases is less than the timer function accuracy, available in application programs.

Beside this a PC operational system shares computer resources between all running programs and it is not granted that the measurement will not be affected by other programs: different drivers, screen function, network activities, etc. The most prevalent operational system for standard PC is MS Windows and a further disquisition will refer to it. This system benefit is in manufacturer supporting of all mutations for built-in drivers for standard interfaces and in a lot of application software, created on MS Windows base. Naturally the response time is predicted for real-time operational systems (for example RTOS[2] or MS DOS[3]) by definition. In such systems a programmer himself defines the response time, considering the real hardware performance.

Let us assume that we have a standard PC with MS Windows and the connected hardware device, which monitors some sensors and controls some nodes in a process. In Fig.1 the hardware informs the PC about the sensor value changing with predicted delay and waits an action to retransmit it in the controllable node. The application software (SW) must response on hardware's messages to prevent an unwanted process deviation. The *PC response delay* between input event and output command is the parameter, which defines the possibility of the virtual instrument to work in real-time. Besides the main control function the software should screen the necessary information, present it in accordance with the current user requirements, and communicate with some other computers connected to the network.

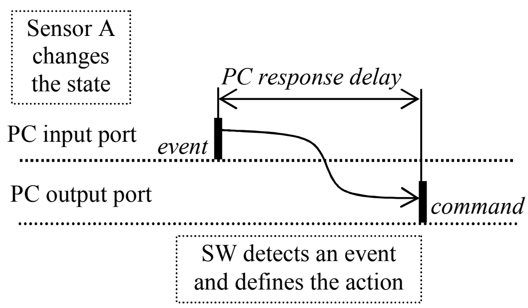

Fig. 1. Response delay definition

To measure the PC response time to randomly occurring events under the circumstances of many simultaneously working programs the special instrumentation is designed.

Response time measurement method

The time diagram of a proposed method, which is implemented in hardware, is presented in Fig.2. The *Clock counter* and a pseudo-random sequence generator are working all the time. The pseudo-random sequence generator is implemented on a base of Linear Shift Register[4]. It is shifted any time when the clock counter overflows. The Register has 7 bits and the state of the last bit *LSR(6)* is used for counter overflows randomized choice. Without activities from the PC nothing else is happening. But after the PC sends a command to the device, this command is interpreted as the *event from PC or response* and the hardware begins to generate *events from HW* at moments, when the clock counter overflows and the state of the *LSR(6)* is HIGH. The PC program should response to this event as soon as possible. At the moment, when the *response* arrives in the hardware, a clock counter value is fixed in the *output data* register. These data are sent to PC in the case of duplex interface or can be read by PC in the case of half-duplex interface. The measurement process continues until the hardware detects that issued event is not responded during the full counter cycle. In this case the hardware stops to generate events and a measurement is accomplished.

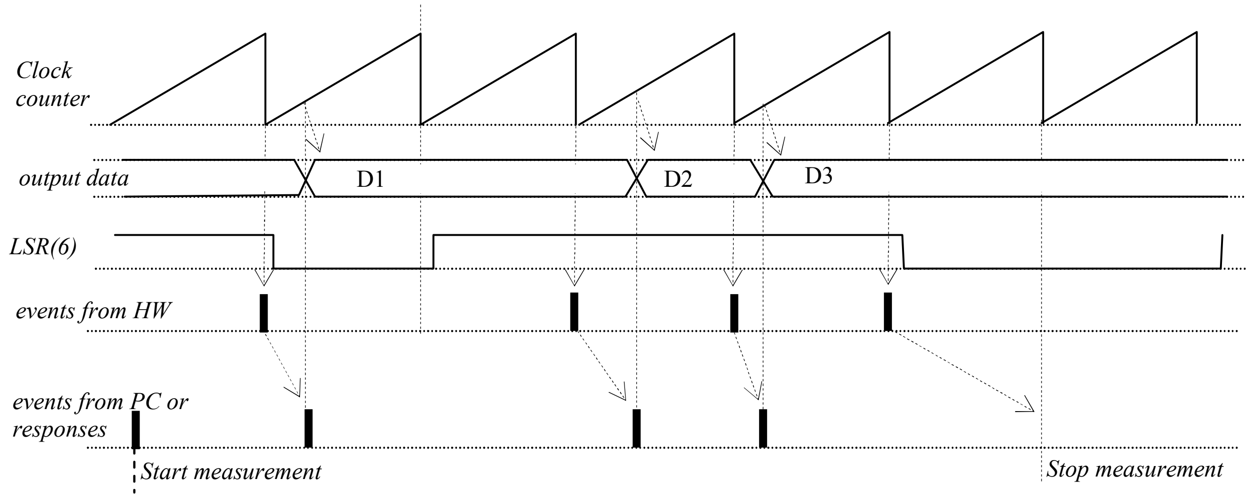

Fig. 2. Response time measurement

The counter cycle should be greater than the time which is necessary for the PC response and data reading in the worst case of delays. The PC software should provide the responses and data receiving in fast time. The data can be processed at once or/and stored for the next processing. For the shown diagram in result of measurement three values of response time will be found: D1, D2, D3.

If the PC is working under real-time operational system all measured delays should be equal. But in the MS Windows all depends on the current system configuration and common resource usage. For the most of users the task priority configuration and resource sharing is too complex problem. Having the instrumentation for response time measuring ones can study this problem and solve it on the base of the experiment.

Virtual instrumentation for response time evaluation

The above described method is implemented in the virtual instrumentation consisting of the special hardware device and some application programs. The hardware is designed as a stand-alone peripheral device, interfacing with the PC through one from four standard PC ports:

RS-232;
Parallel Port;
USB 1 (up to 12 Mbs) and
USB 2 (up to 480 Mbs).

The block diagram of hardware schematic is presented in Fig.3. Interface protocols on the signal level (for USB the complete set of protocols) are realized in corresponding standard chips. USB chips execute the complete set of protocols. Accordingly the device has four connectors:

- DB9-F for PC RS-232 port connection;
- DB25-F for Parallel Port connection;
- two USB (B type) connectors for USB 1 and USB 2 port.

All other schematic, related with digital data processing, communication and measurement functions is implemented in CPLD chip from Altera Inc.

Only one interface should be connected for a work with PC. The operating interface is selected by user with a help of an external button and *Control and communication block.* This block communicates data and generates necessary control signals for the selected interface. *Data converter* modules in connection with the corresponding interface chips execute specialized for each interface data and control exchanges, but in the exchange with the *Measurement block* all operations are uniformed and narrowed down to the input signals *event*, output signal *response*, and measured data block – *data*. Such versatile structure allows to add in the design new interface chip and VHDL module for corresponding data conversion and to evaluate this new interface performance.

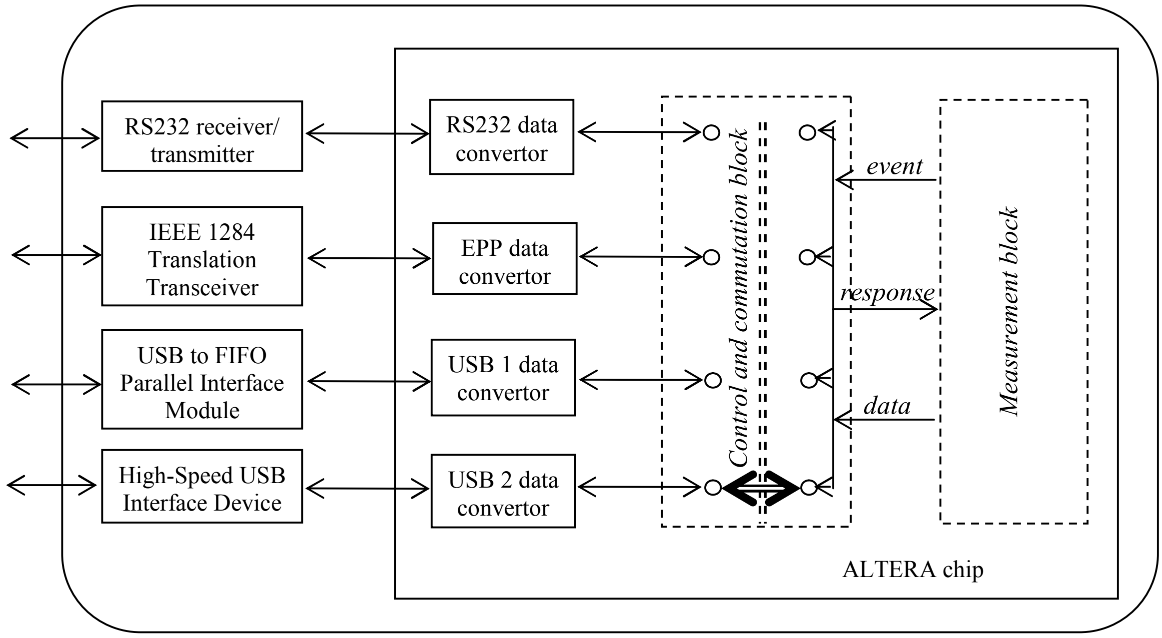

Fig. 3. Schematic block diagram of measurement hardware

A precision of the response time measurement is directly defined by a system clock, which in realized design is 25 MHz and gives the precision 40 ns. One measurement data block has three bytes and can present response delays up to $40 \times 10^{-6} \times 2^{24} = 670$ ms as a number of this clock periods. It is quite enough for all types of interfaces, also including the data reading time, but for fast interfaces the design allows to place the clock generator 125 MHz and get the precision 8 ns.

Beside the response delay measurement the device can measure the time intervals with the same precision for events coming in special BNS connector. The SMA connector is designated for external clock input. The currently active (chosen by user) interface is indicated by corresponding LED. The hardware device is designed as a single board, which is shown in Fig.4.

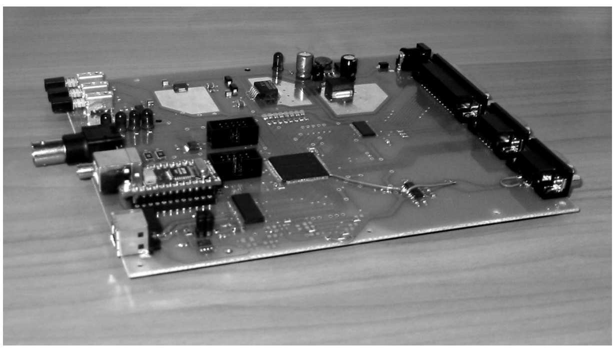

Fig. 4. Hardware printed circuit board implementation

The application programs interact with the hardware device and accumulate measured data. There are four separate programs for each realized interface although they have some common functions. Such separation excludes all additional resources usage for a functions not related with the selected interface. Programs are written as samples in C language and can be built in the user application software to simulate the real hardware instrumentation and evaluate a general virtual instrument performance.

Performance evaluation results

RS232. The Recommended Standard 232 is widely used in PC serial ports. RS232 is the slowest interface from realized in the design, but it is used in many connections with a hardware not demanding for fast interactions. This standard requires the interaction parameter adjustment, such as a bit rate, number of bits in transmitted batches and so on. These parameters directly affect the interface performance. The evaluation was done for 10-bit batches (1 start bit, 8 data bits, 1 stop bit) and two bit rates: 9600 and 115 200 bps.

Parallel Port. The Parallel Port is the most commonly used port for interfacing the hardware requiring the real-time interaction. Enhanced Parallel Port Protocol (EPP) for this interface is described in details in many manuals and is relatively simple in use. As defined in the standard IEEE-1284 the transfer rates up to 10 Mbytes/s can be achieved.

Universal serial bus. Universal Serial Bus (USB) is elaborated to get uniformity for different PC peripheral connections and to improve plug-and-play capabilities. USB specification allows three types of bit rates: low speed = 1.5 Mbs, full speed = 12 Mbs and high speed = 480 Mbs for USB2. Manufacturers promise arriving in 2009 the products confirming the USB 3 (4.8 Gbs!). But in despite of such bit rates the typical high speed USB devices operate at lower speeds, often about 3 MBs overall, sometimes up to 10-20 MBs. The highest USB data transfer rate claimed by USB vendors is 40 MBs.

In the design both USB1 and USB2 are realized. For the USB1 realization the chip set and drivers from FTDI Ltd are used. This product allows quickly replace the RS232 by the USB1 interface, increasing the bit rate in 100 times. The USB2 is realized on EZ-USB FX2LP chip from

Cyprees Semiconductor. This product provides the 480 Mbs, but requires more competence in microprocessor programming and in drivers for PC.

The instrumentation was tested in PC Intel Celeron 1,66 GHz under Windows/XP. The evaluation results obtained for the response time and maximal throughput for all considered interfaces are presented in Table 1.

Table 1. Performance evaluation for different interfaces

PC interface	Response time (ms)			Maximal throughput
	min	max	avr.(99.5%)	
RS-232-9600 bps	4.0	7.6	4.1	0.96 KB/s
RS-232-115200 bps	0.4	3.8	0.5	11.5 KB/s
Parallel Port (EPP)	0.005	0.08	0.01	165.6KB/s
USB1 (12 Mbs)	2.9	8.4	4.0	1.2 MB/s
USB2 (480 Mbs)	0.8	4.3	2.2	13 MB/s

These data are preliminary, only for illustration. Actually all these estimates depend on the used PC, operational system and concurrent software running. A statistical distribution of the estimates as a function of environment configuration will require the special study.

Conclusions

The presented instrumentation provides the possibility to evaluate the performance in the interface with the controlled hardware in real operational environment. Using this instrumentation the customer can optimize his application software and MS Windows system parameter tuning. Having the information about delays in the interface, depending on various additional processes, it is possible to define interfering tasks and avoid their activity. The precise measurement of the response time and its variation gives very valuable data for "soft" real-time systems designs based on the standard PC with MS Windows, providing the minimal interchange service quality degradation.

References

1. **Truchard J.** The future of virtual instrumentation// Scientific Computing World**:** May / June 2004.
2. **Жданов А.** Операционные системы реального времени.// PCWeek, 8/1999.
3. **Финогенов К.Г.** Самоучитель по системным функциям MS DOS. / Телеком, 2001, 382 с. ISBN: 5-93517-051-5.
4. **Мелик-Шахназаров А.М., Маркатун М.Г.** Цифровые измерительные системы корреляционного типа. /Москва:Энергоатомиздат, 1985, 128 с.

Submitted for publication 2008 02 15

E. Boole, D. Stepin, V. Stepin. Delay Evaluation in Interactive Exchange with Controllable Devices for Some Type of Computer Interfaces // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 4(84). – P. 81–84.

In real-time control of devices, connected to the standard computer interfaces, it is essential to evaluate delays in the interactive program exchange with devices. These delays are defined both by the interface type and computer operational environment. The device implementation, registering the delay of program response to events from the device, is presented. These delays can be read and statistically processed in the computer. The presented device and related sample programs in C language support four standard interfaces: RS-232, EPP, USB1 and USB2. The results of the delay evaluation for these interfaces are presented, too. Ill. 4, tabl. 1, bibl. 4 (in English; summaries in English, Russian and Lithuanian).

Е. Буль, Д. Степин, В . Степин. Оценки задержек при интерактивном взаимодействии с управляемыми устройствами для разных типов компьютерных интерфейсов // Электроника и электротехника. – Каунас: Технология, 2008. – № 4(84). – С. 81–84.

При управлении устройствами реального времени, подключенными к стандартному компьютерному интерфейсу, важно установить задержки в интерактивном взаимодействии между программой управления и устройством. Эти задержки определяются как самим интерфейсом так и операционной средой компьютера. Предлагается реализация устройства, фиксирующего время ответной реакции программы на сообщения со стороны устройства. Эти времена считываются и статистически обрабатываются в компьютере. Предлагаемые устройство и примеры программ управления на языке С поддерживают четыре стандартных подключения к компьютеру: RS-232, EPP, USB1 и USB2. Приводятся также результаты измерений задержек для этих портов. Ил. 4, табл. 1, библ. 4 (на английском языке; рефераты на английском, русском и литовском яз.).

E. Boole, D. Stepin, V. Stepin. Interaktyvios keleto tipų kompiuterių sąsajų saveikos su valdomais prietaisais vėlinimo nustatymas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2008. – Nr. 4(84). – P. 81–84.

Realiu laiku atliekant prie standartinių kompiuterinių sąsajų prijungtų prietaisų kontrolę, būtina atsižvelgti į valdymo programos ir prietaiso tarpusavio sąveikos vėlinimą. Šie vėlinimai yra sąlygojami tiek pačios sąsajos tipo, tiek kompiuterio operacinės terpės. Jie gali būti nuskaitomi ir statistikai apdorojami kompiuteryje. Siūlomas prietaisas ir valdymo programos C kalba pavyzdžiai palaiko keturias standartines sąsajas: RS-232, EPP, USB1 ir USB2. Pateikti šių sąsajų vėlinimo matavimo rezultatai. Il. 4, lent. 1, bibl. 4 (anglų k.; santraukos anglų, rusų ir lietuvių k.).