XVIII IMEKO WORLD CONGRESS
Metrology for a Sustainable Development
September, 17 – 22, 2006, Rio de Janeiro, Brazil

SOFTWARE FOR ADVANCED USB CONTROLLER TO BE APPLIED IN MEASUREMENT INSTRUMENTATION

Andrzej Podgórski [1], *Sławomir Górski* [2], *Andrzej Miękina* [3]

[1] Warsaw University of Technology, Faculty of Electronics an Information Technology, Warsaw, Poland, A.Podgorski@ire.pw.edu.pl
[2] SG Elektronik-Usługi Inżynierskie, Tychów Nowy, Poland, sgorski@svantek.com.pl
[3] Warsaw University of Technology, Faculty of Electronics an Information Technology, Warsaw, Poland, A.Miekina@ire.pw.edu.pl

Abstract: The application of an advanced USB controller in measurement instrumentation is proposed. That solution enables the direct connection of any mass memory to a measuring device in order to enhance its functionality. The corresponding software is briefly described and the results of its testing are presented.

Keywords: USB controller, specialized software, flash disk memory.

1. BASIC INFORMATION

The Universal Serial Bus (USB) interface – making possible relatively fast transmission of data [1], [2], [3], [4] – is more and more commonly used in various devices. The USB communication between two devices is established if each of them is equipped with a controller. There are various types of controllers. In an ordinary peripheral device (*i.e.* printer, external memory, measurement instrument), there is usually installed a controller which furnishes only the DEVICE mode. On the other hand, the USB controller of a PC is working in the HOST mode. The peripheral instrument with an ordinary USB controller cannot communicate with another peripheral device but only with the PC. Nowadays, there are on the market more advanced USB controllers which allow for more sophisticated usage of this interface. One of them is the circuit called ISP 1362 [5], [6], [7]. It has been chosen for application in a sound-and-vibration measurement instrumentation to be considered in this paper. Such a controller enables the connection between a measuring device and a PC in order to establish bi-directional communication. The settings can be sent from the PC to a device and the measurement results can be transferred in real time or off line from a device to the PC. Additionally, what is the most interesting future, the device with the ISP 1362 USB controller can send measurement results to an external flash memory [8], [9] enhancing the capacity of the available memory. In particular, all mass storage devices are served where the data transfer is made in Bulk-Only protocol with FAT16 and FAT32 disk file system [10], [13]. It enables the broadening of the application scope of the device. A portable measurement instrument can be used additionally as a data logger – the data coming from an input converter of the unit can be directly recorded on a Flash Disk whose capacity nowadays reaches even 4 GB [11]. According to our knowledge, such application of an advanced USB controller does not exist in the instrumentation for sound-and-vibration measurements and there is no software for such implementation.

In the next sections, a short description of the ISP 1362 is given and the developed software is presented which enables the user to fully utilize in a measurement instrument the potential of the presented circuit. Finally, the results of the performed tests are provided which prove that the considered controller can be successfully used in the sound-and-vibration measuring instruments.

2. DESCRIPTION OF THE ISP 1362 CIRCUIT

The ISP 1362 circuit is the USB controller which integrates various available functions of its interface: HOST, DEVICE and OTG. Simultaneous work in HOST and DEVICE mode enables one to connect the instrument, where the circuit is installed, to a PC, as well as to another peripheral device, *e.g.* a flash disk memory or any other mass storage unit. The controller is compliant with Universal Serial Bus Specification Rev. 2.0, supporting data transfer at full speed (12 Mbit/s).

The ISP 1362 circuit has two USB ports. One of them (number 2) is constantly assigned to the HOST functionality and another one (number 1) can be used in HOST, DEVICE or OTG mode, depending on the requirements. The configuration of this port is made by means of the ID line of the USB cable. In the described application, this port is set in the DEVICE mode; OTG mode is not used.

The communication with the ISP 1362 circuit can be done in two different modes, called PIO and DMA. In our application, the PIO mode is used, which utilizes 2 address lines, 16 data lines, as well as read (RD), write (WR) and chip select (CS) line. Two addresses are assigned to the HOST mode and two others to the DEVICE mode. In each module, one address is dedicated to data transmission and another to the commands. The parallel interface can reach 10 MB/s data transfer rate between the microprocessor and the ISP 1362 circuit.

The ISP 1362 has two interrupt lines, one for the HOST and another for the DEVICE mode. It is also possible to use only one interrupt line for both modes as it is implemented in the consider case.

The circuit has additionally two lines which are used for signalizing the sleep mode of its parts, the HOST mode and the DEVICE mode respectively. The same lines are used to wake up the circuits by the control processor. This feature is very useful if the circuit is used only as the HOST or DEVICE – the power consumption is reduced than significantly.

The HOST and DEVICE functionalities are almost completely independent: they use separate addresses, separate configuration registers, and different memory buffers.

To the HOST functionality, the 4 kB of configurable memory is assigned. The memory buffer is directly addressed in order to enable the programmer to change any part of the memory. In that functionality, all four possible USB transmission types are implemented: CONTROL, BULK, INTERRUPT and ISOCHRONOUS.

To the DEVICE functionality, 14 configurable endpoints and 2462 B of data memory are assigned. The memory can be assigned in any way to the endpoints. It is possible to work in a ping-pong mode (double buffering). This option is configurable for each endpoint, without zero endpoint.

The configuration registers have 16 or 32 bits. The 16-bits data are transmitted between the control processor and the ISP 1362 circuit.

3. DATA TRANSMISSION BETWEEN THE ISP 1362 CIRCUIT AND A PROCESSOR

The data transmission is differently organized in the HOST mode and the DEVICE mode.

In the HOST mode, the data are written in and read from the registers and buffers. At the beginning of each transmission cycle, the address of the register is written in a dedicated register. For data “write down”, the seventh bit of the address register should be set. After the address writing, the data transmission phase follows: the “write to” or “read from” another dedicated register.

In the DEVICE mode, the command option and data option are used. In the command state, the 16-bit number is written to a dedicated register. After writing a command, one can write data to another dedicated register or read them from the same one. The number of the data depends on the command type.

4. SOFTWARE STRUCTURE

The software for ISP 1362 circuit has been developed using an assembler language and the C language. It consists of the following parts:

- initialization of the ISP 1362,
- organization of the DEVICE mode,
- organization of the HOST mode,
- organization of the HOST and DEVICE mode for real time data transmission.

The transmissions in the HOST and the DEVICE modes can be performed simultaneously. In a particular, if the instrument is connected to a PC and at the same time a flash disk is connected to the instrument, the transmission between a PC and a flash disk is possible. The real-time HOST and HOST mode can not operate simultaneously.

4.1. Initialization of the ISP 1362

The initialization is performed by a procedure written as an assembler function. It configures the circuit and starts the HOST and DEVICE modes of the controller.

4.2. Organization of the DEVICE mode

The data transmission handling is granted to a PC which is responsible for each transmission initiation. In such case, the most sensible solution is the usage of the interrupts from the DEVICE part of the ISP 1362 circuit. The organization of the DEVICE mode is based on the analysis of the interrupts coming from the ISP 1362: the interrupt calls the special function which finds out the cause and calls the proper service function. Some parts of the DEVICE software allied with the real-time data transmission are performed in the interrupts from the input converter of the measuring instrument.

The presented software uses four endpoints:

0 – control transmission with the following standard requests: GET DESCRIPTOR (device or configuration), SET ADDRESS, SET CONFIGURATION, SET INTERFACE, CLEAR FEATURE;

1 – data transmission in BULK OUT mode from a PC to a device;

2 – data transmission in BULK IN mode from a device to a PC;

3 – real-time data transmission with double buffering in BULK IN mode from a device to a PC.

The endpoint 3, used for the measurement data transmission to a PC in real-time mode is described in Section 4.4. The endpoints 1 and 2 are used for the measurement and settings data file transmission between a PC and the instrument, as well as for sending commands to the instrument. This type of transmission is performed off line so the speed is not a critical parameter.

An algorithm for data transmission to a PC in the DEVICE mode consists of: writing proper data to the endpoint’s buffer (address setting, sending the number of data to be transferred, sending the data); setting the bit confirming the data correctness; waiting for the interrupt from the endpoint confirming the transmission end and the possibility of the beginning of the next one.

An algorithm for data transmission from a PC in the DEVICE mode consists of: waiting for the interrupt from the receiving endpoint; reading out the data from the endpoint’s buffer (address setting, read out the number of data placed in the buffer, read out the data from the buffer); in case of the control endpoint – the confirmation of data receiving; clearing the buffer and preparing it for the next data reception.

4.3. Organization of the HOST mode

The organization of the HOST mode is used for the service of the flash disk memory. The instrument is responsible for the transmission handling in this mode. The data saved in the mass storage devices (*i.e.* hard or flash disks) are usually organised in a file system which should be recognisable in a PC without additional software. Our software can handle widely used and relatively simple FAT16 and FAT32 file systems, however with some limitations. The data writing in a file system at a given moment requires additional memory and computational power of the processor. Two types of files are written to a flash disk: the measurement data having up to few kilobytes results and the measurement results from the instrument's buffer. The data writing is described in Section 4.4. The developed procedure consists of three shells: USB, SCSI and the part servicing the FAT16 or FAT32 disk file system.

The USB shell was done using the Bulk-Only protocol and consists of data transmission in two directions with data writing and reading based on two buffers. The whole 4 kB of HOST memory was dedicated to Aperiodic Transfer (ATL) transmission type as only CONTROL and BULK modes are used. The ATL memory is configured into blocks, we made 8 blocks, each containing 512 bytes. To each block, special bits in the selected registers are attributed. The Philips Transfer Descriptor (PTD) 8-byte header is added to each transferred block. The data transmission algorithm consists of the following steps: the transfer to the ATL buffer (data and PTD in the case of writing and only PTD in reading) – setting the address and the number of data to be transferred in the proper registers and then – data writing to the buffer, switching on the interrupt from the ATL, waiting for the interrupt with reading special register and checking the setting of its certain bit (it is necessary because of the known error of the ISP 1362 circuit), switching of the ATL processing, identification of the block being processed, reading data from the ATL buffer (data and PTD in the case of reading and only PTD in writing) – setting the address and the number of data to be transferred in the proper registers and then – data reading from the buffer. Read header with PTD contains the information about the currently performed operation, its status, number of transmitted data with or without success.

The SCSI shell was written in the assembler and partially is used in the interrupt procedures from the converter in order to speed up the real-time operations. The following SCSI commands were implemented: testing the presence of the external flash disk, asking for its parameters and its state, reading out its capacity, writing and reading data sector. A single transmission starts with the CBW command (40-B data sending including PTD and 8 B of PTD receiving). Then, if there any error occurres, the PTD is written and, depending on the direction of the transmission, the data are written and PTD is read or PTD is read and the data. Finally, the CSW status is read (8 B of PTD is send and then read together with 8 B of status). So, if the transmission of one 512 B data block is performed, 600 B are transferred: CBW (40 B + 8 B) + Data (512 B) + CSW (8 B + 16 B).

The servicing of the disk file system (FAT16, FAT32) utilizes two SCSI commands: read block and write block. Practically, there is only one size of the sector available in the disk file systems equal to 512 B. Other, theoretically possible (1024 B, 2048 B or 4096 B), are practically unused. The data are transferred using 512 B sector as a unit. The system servicing starts with the disk initialization and the read out of the data describing the disk partition parameters. The proper data structures and global variables were declared and initialized. The notion of current directory was defined in order to simplify the disk operations. All commands (file creation, write, read, delete, and change of the directory) are made in the current directory. After the initialization the main directory is the current one. The structure of the current directory contains the data about the number of the files and sub-directories and the placing of the directory on the disk. Due to the fact that in FAT16 system there is a limited number of files in the main catalogue the developed software forces a user a selection or creation of new working catalogue.

The FAT is used in two modes which differ slightly. In the real-time operation the writing to the disk is performed in the interrupt servicing procedures coming from the converter and simplified as much as it is possible. In the normal mode (not the real-time one) the continuous free memory space is not required. The free sectors are used. The different functions of file handling were implemented: for data reading, data appending, file creation, file deleting, scanning the contents of the directories, changing the directories, etc.

The FAT file system requires during the creation and writing of a new file the access to three places on the disk: the catalogue, the FAT table and the data sectors.

The data writing requires to find out on the disk, after searching the FAT table, a free space. Because this operation is time consuming, in a spare time one of the programme plots looks for the free clusters in the FAT table and writes down their numbers in the special buffer. This solution enables one to minimise time of FAT table searching. This time depends on the disk capacity and file system (FAT16 or FAT32) and can be even up to 20 seconds.

The file writing in the catalogue requires checking whether in this catalogue the file with the given name already exists or not. This operation can be also time consuming in the case of the big number of the files in the catalogue. In the instruments there is a possibility of writing the measurement data to the consecutive files which name differ only on the last few characters in which the current increasing number is written. The files can be written even every 10 seconds. In such case only simplified search in FAT table is performed – the biggest number is looked for.

The small file writing requires a lot of time for servicing the FAT system comparing to the writing file itself. For example the writing of one-sector-long file requires at least the transfer of 5 sectors (the reading of the sector with the place for writing a catalogue, new file insertion and renewed writing of the sector, the reading of FAT table sector, complement of the inscription and its renewed writing and the writing of the data sector).

In order to minimise the number of data transfers in the processor's memory one sector of FAT table is kept. It is written to the disk only in the case when the access to another FAT's sector is required one sector in FAT32 system consists of 128 inscriptions and in FAT16 – 256. If the disk contains a lot of free space and the consecutive clusters are used, the number of data transfers is significantly reduced.

4.4. Organization of the HOST and DEVICE mode for the real-time data transmission

The real-time mode software is included in the procedures of the input converter of the measuring device interrupts servicing. It consists of three special functions which permit the data transmission using the USB in the HOST and in the DEVICE mode. These functions are called sequentially depending on the flags set by them.

The creation and writing a file in the HOST mode consists of three stages: file creation and transmission initialization, data writing to the consecutive sectors of the disk in interrupt procedure and completion of the file system structure and file closing. In order to achieve the fastest possible speed of data writing in the real-time, the free continuous disk space is used assigned in the initialization process. In order to fulfil that requirement the flash disk should be empty or, at least, defragmented. The process of file closing is done after the end of data transmission and it takes certain amount of time (even up to few second!) depending on the number or written clusters.

The first function takes two bytes from a buffer and writes them to the sending buffers of the HOST or DEVICE mode of the ISP 1362. After 64 B transmission the end flag is set as well as the certain bits of the ISP 1362 registers permitting the USB transmission. In the DEVICE mode the USB transmission is started after each 64 B, while in the HOST mode – after 512 B (one sector).

The second function is called and executed after the first one. It waits for the completion of the USB transmission sequentially checking its state. After the end of the transfer the status is read and the proper bits of the ISP 1362 registers are set.

The third function is called after the second one. It calculates the data to be transmitted together with the samples in the next 64 B block and initiates the transmission for the HOST and DEVICE modes. At the end that function sets the proper flags permitting calling the first function.

The functions, described briefly above, are called sequentially depending on the flags set by them. During the execution of the first one the interrupts from the USB are blocked because any transmission with the ISP 1362 could be treated in that time as the writing to the sending buffer of the HOST or DEVICE modes.

In this mode there is a limitation of the maximum length of the file to be written to the flash disk which can not be greater than 4 GB (namely 0xFFFFFFFF). In our application it allows to register in one file the signal during 6 hours and 12 minutes of measurement. In the case when the 4 GB was already fulfilled, in the RT mode, the registration is not finished. The consecutive signal is registered in the following files.

In the HOST mode the FAT table is not complemented during the measurements in order to enable one the achievement of the biggest possible, unstopped transmission of the results. After the transmission the filling up of the FAT table can take even tens of seconds.

4.5. Signalization and handling of data transmission errors

In the real-time mode, any error is stopping the transmission and closing the currently written file. There is a variable containing the code of the transmission error which can occur in the reading or writing operation.

During the ordinary operation, in the case of an error all functions report it to the calling function, which interprets it properly.

5. RESULTS OF TESTING

The developed software is dedicated for wide class of external devices fulfilling the Mass Storage requirements: data transfer in Bulk-Only protocol using FAT16 or FAT32 disk file system. The prospectus implementation of the ISP 1362 controller and the presented software in the measurement instrumentation should permit to register large amount of data on commonly used and widely accessible mass memories enhancing the functionality and the field of the application of the devices in which the controller is going to be used.

In real-time mode the required capacity of the disk (FIFO buffer), which is necessary to avoid loosing the data during their writing to the external disk memory, depends on the type of the used memory. The performed tests with the Kingston Data Traveler Elite 1 GB [12] proved that 8 kB is sufficient. It generates the additional delay related to the sampling frequency of the converter, i.e. for 48 kHz the delay is equal to 166 ms. Such memory is sufficient to register the measurement data from the output of the 24-bit A/D converter, sampled at the frequency equal to 48 kHz, for a period of more than 93 minutes (each sample contains 4 bytes: 3 bytes for data from A/D converter and 1 byte containing the status). Additionally, the developed software was tested on a hard disk Seagate Barracuda 7200.7 120 GB, which was placed in the USB slot and in which 30 GB partition in FAT32 system was created.

We implemented and tested with the mass memory devices both: FAT16 and FAT32 disk file systems. There are some limitations coming from the FAT16 disk system, namely the number of files in the main directory is limited and the capacity of the disk can not be bigger than 2 GB. In the FAT32 disk system, the capacity of the disk can be much bigger but the single file can not be greater than 4 GB.

The size of the data buffer in the instrument (FIFO sequence) required for having the continuous writing to the disk in the real-time mode depends on the flash disk model and is not normed, *i.e.* for Kingston Data Traveler Elite 1 GB is equal to 8 kWords. In the case of the instrument's sampling frequency equal to 48 kHz it generates the delay equal to 166 ms. In the case of the 30-hours (3 times 10 hours) tests performed for the Seagate Barracuda 7200.7 120 GB hard disk in the USB slot 8 kWords buffer is also sufficient.

The initialisation of the disk takes an amount of time, mainly used for the searching of the continuous space for the RT transmission. The exemplary results of the initialisation, coming from performed tests, are given below:

- Hard disk in the USB slot, 30 GB, FAT32 – 52 s,
- Kingston 1 GB with FAT32 flash disk – 8 s,
- Kingston 1 GB with FAT16 flash disk – 3 s.

The initialisation time for FAT16 system is much more shorter than for FAT32 system due to the fact that the number of cluster is smaller, the cluster are bigger and the writings to FAT16 table consist of only 16 bits instead of 32 bits in the case of FAT32.

The closing of the file in the real-time mode during which some writings to the FAT table are made, also takes not negligible amount of time. The exemplary results of the performed tests are given below:

- Hard disk in the USB slot, 30 GB, FAT32 closing files with the total capacity of 6.2 GB – 19 s,
- Kingston 1 GB, FAT32 flash disk, 960 MB file – 13 s,
- Kingston 1 GB, FAT16 flash disk, 960 MB file – 3 s.

6. CONCLUSION

The developed software for the advanced USB controller is dedicated to a wide class of external devices fulfilling the mass storage requirements: data transfer in Bulk-Only protocol using FAT16 or FAT32 disk file system. The results of the performed tests have proven that the implementation of the ISP 1362 controller and the presented software in a measurement instrument should make possible the storing of large amounts of data on commonly used and widely accessible mass memories enhancing the functionality and the range of the applicability of the devices where the controller is going to be used.

ACKNOWLEDGMENTS

The presented work was sponsored by SVANTEK sp. z o.o., a producer of the instrumentation for sound and vibration measurements and supported by the State Committee of Scientific Research, Poland (grant # 4 T10C 017 24).

REFERENCES

[1] *http://www.usb.org/home*

[2] *http://www.intel.com/technology/usb/*

[3] *http://computer.howstuffworks.com/usb.htm*

[4] *http://en.wikipedia.org/wiki/Universal_Serial_Bus*

[5] *http://www.semiconductors.philips.com/cgi-bin/pldb/pip/isp1362.html*

[6] *http://www.chipcatalog.com/Philips/ISP1362.htm*

[7] *http://www.chipcatalog.com/Datasheet/BF8F9132BA68EFF6DCAAE51D3EBE33DB.htm*

[8] *http://www.kingston.com/digitalmedia/datatraveler_home.asp*

[9] *http://www.techadvice.com/help/Products/u/usb-flash-memory.asp*

[10] *http://www.usb.org/developers/devclass_docs#approved*

[11] *http://www.kingston.com/digitalmedia/dt_chart.asp*

[12] *http://www.kingston.com/digitalmedia/dt_elite.asp*

[13] *http://download.microsoft.com/download/1/6/1/161ba512-40e2-4cc9-843a-923143f3456c/fatgen103.doc*