*Jurnal Teknologi*, 41(D) Dis. 2004: 43–52


# AN OPTICAL TOMOGRAPHY SYSTEM BASED ON THE UNIVERSAL SERIAL BUS

RUZAIRI ABDUL RAHIM[1], TEE ZHEN CONG[2] & CHAN KOK SAN[3]

**Abstract.** In the past, most of the researches in tomography system concentrated more on the design of the sensor, and made use of the data acquisition system as the interface to the personal computer. This will increase the cost of the system. In this project, Universal Serial Bus was successfully implemented in an optical tomography system, providing a low cost, high-speed data acquisition system. The system, however, is an offline system. The system had successfully processed all the data in the lower level, using the micro controller before sending the data into the PC through the USB.

*Keywords:* Universal Serial Bus, optical tomography system, data acquisition system, tomogram

## 1.0 INTRODUCTION

Process Tomography is a technique of obtaining images of a cross section of an industrial process, for example, a multi-phase flow in an oil pipeline, so that the internal behavior can be investigated [1]. Process Tomography system consists of three main units: a sensing unit, sensing electronics, and an image reconstruction unit. The sensor is mounted directly onto the process equipment and the sensing electronics is located at the site of measurement. The image reconstruction computer may be located far away from the process, for example, in the central control room, as the site may be hazardous. A process tomography system is capable of producing tens to hundreds of images per second, which requires a high-speed serial link, say 20Mbits per second.

Electrical Tomography provides inexpensive non-intrusive imaging systems with low but sufficient resolution of the internal distributions of processes [1]. Optical tomography system, such as fan beam optical tomography system, and parallel beam optical tomography system uses optical transmitters and receivers.

## 2.0 UNIVERSAL SERIAL BUS TECHNOLOGY

Universal Serial Bus (USB) has been the most advanced interfacing standard for computer communications and I/O devices so far [2]. USB interfacing has enabled the computer peripherals to be connected by a single type of connector. USB was

[1,2&3]Process Tomography Research Group (PROTOM), Control & Instrumentation Engineering Department, Faculty of Electrical Engineering, Universiti Teknologi Malaysia. 81310 UTM Skudai, Johor, Malaysia. Email: ruzairi@suria.fke.utm.my

developed by a group of leading companies in computer and communication, such as Compaq, DEC, IBM, Intel, Microsoft, and Northern Telecom [3]. USB connection is a plug and play connection without having to open the box, install a card into dedicated computer slots, and reconfigure the system [3].

USB makes it convenient to simultaneously use and connect up to 127 peripherals to a computer without using several connectors, different interrupts, and I/O address [3]. The computer automatically recognizes the device connected and installs the appropriate drivers. It enables computer users to "hot-plug" computer peripherals to their personal computer (PC) and start using them without having to reboot the PC [4].

## 3.0 HARDWARE DEVELOPMENT

This research is divided into two tasks, which is to establish the communication between hardware and PC using USB technology, and also to display the results of the data transfer into a PC. In other words, the objective is to display tomograms of the flow regime inside a pipeline.

### 3.1 Communication Between PC and Hardware

A micro controller from Microchip Company, PIC16C765, is shown in Figure 1.

It was used as the hardware that will send data to a PC. This micro controller is chosen because it has a built-in USB controller. This micro controller is fully compliant with Universal Serial Bus Specification 1.1. The micro controller will be used to generate

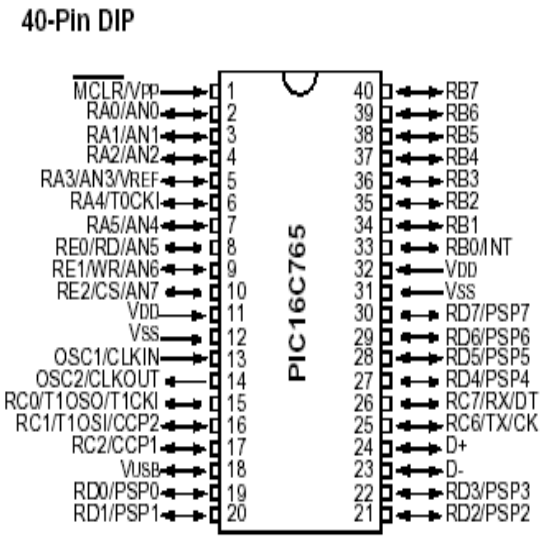


**Figure 1** Pin distribution for PIC16C765

a simulated data, and the data will be processed, and then sent to a PC using USB. As for the PC, a program is written in Visual Basic to recognize the hardware, and get the data from the hardware.

The high performance of the Microchip micro controller can be attributed to a number of architectural features commonly found in RISC microprocessors [5]. The examples of these features are Harvard architecture, long word instruction, single word instruction, single cycle instruction, instruction pipelining, and reduced instruction set [5].

The PIC16C765 device is a low cost, high-performance, CMOS, fully static, 8-bit micro controller in the PIC16Cxx mid-range family. This device has 33 input output pins and 256 bytes of RAM. The PIC16C765 device fit nicely in many applications ranging from security and remote sensors to appliance controls and automotives. The special feature of this device is that it has a USB controller built in the chip.

## 3.2 USB in PIC16C765

The PIC16C765 peripheral module supports low speed (1. 5 Mbps) control and interrupt (IN and OUT) transfers only. It supports three endpoint numbers, 0, 1, and 2 for a total of 6 endpoints.

When the USB transmits or receives data, the SIE will first check that the corresponding endpoint and direction Buffer Description (BD) UOWN bit equal to 1. The USB will move the data to or from the corresponding buffer. When the TOKEN is complete, the USB will update the BD status and change the UOWN bit to 0. The USTAT register is updated and the TOK_DNE interrupt is set. When the MCU processes the TOK_DNE interrupt, it reads the USTAT register, which gives the MCU the information it needs to process the endpoint. At this point, the MCU will process the data and set the corresponding UOWN bit.

The USB module is controlled by 7 registers, plus those that control each endpoint and endpoint/ direction buffer. These registers include USB interrupt register, USB interrupt enable register, USB error interrupt status register, USB error interrupt enable register, status register, USB control register, USB address register, USB endpoint control register, buffer descriptor status register, buffer descriptor byte count register, and buffer descriptor address register.

The micro controller is connected in a way that it will function as a micro controller, and also a USB controller. The most important element in a micro controller is its oscillator. In this PIC16C765 micro controller, a 24 MHz crystal is used as the source for the oscillator. Figure 2 shows the connections for the crystal for this micro controller.

Besides that, four LED are connected to PORT A of the micro controller. These LED will show whether the micro controller functions as demanded or whether it malfunctioned. Lights at the LED show that the micro controller is functioning.

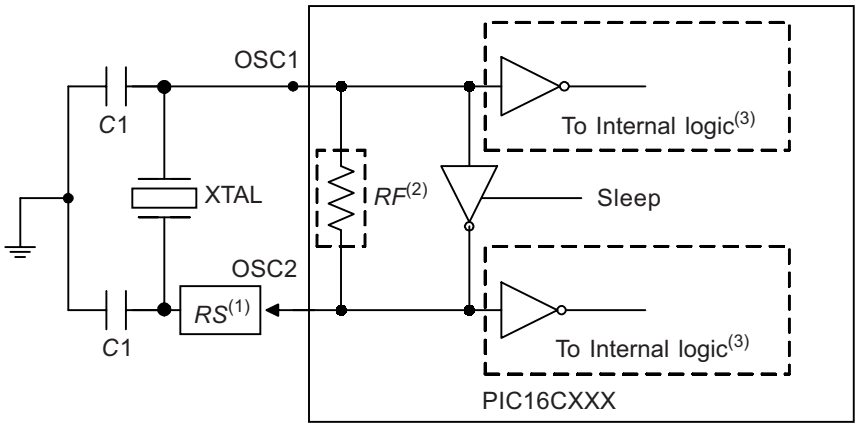


**Figure 2** Connection of crystal in PIC16C765

3 pins for USB port are connected to a USB connector, and it will be connected to the PC using a USB cable. A 3.3V regulator provides the D+ / D– drives with power, as well as an external pin. This pin is intended to be used to power a 1.5 kohm pull-up resistor on the D– line, to signal a low speed device. A 200 nF capacitor is required on pin Vusb of the PIC16C765 for regulator stability. Figure 3 shows the circuitry for the USB application.

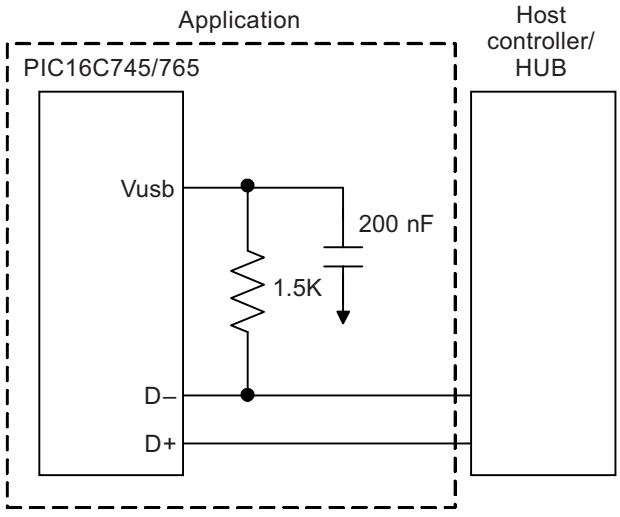


**Figure 3** Circuitry for USB application

## 4.0 IMAGE RECONSTRUCTION AND IMAGE DISPLAY

In order to display the data flow into the PC, an image reconstruction and image display system are needed. The Visual Basic language was used for the image reconstruction algorithm. The data which are successfully received by the personal computer will be converted into a tomogram.

The software is divided into three parts: the assembly language for PIC16C765; the programming language in the PC to communicate with the hardware and acquire the data, and the programming language in the PC to display the data as a tomogram.

### 4.1 Assembly Language in PIC16C765

The assembly language for the micro controller is a C-compiler provided by the CCS Company. This C-compiler is called the PCM C-compiler. This compiler can be linked together with the original compiler for the Microchip micro controller, MPLAB, and generate the hex file needed. Users need to type in the C code needed, and then, the compiler will generate the hex file, as well as a listing fail, which is written in the assembly language as in MPLAB. In this project, the assembly language had been divided into several parts; each part contained its own function. Every part contained many sub routines that will function as a micro controller and USB controller.

#### *4.1.1 PIC16C765 as a Micro Controller*

There is no way that an equation can be inserted into the assembly language. There is no such function in the C-compiler. Therefore, a waveform desired was drawn in a piece of paper, and then the waveform was divided into 32 parts, and each value was taken. The value will be the initial value for the simulated 32 sets of data, and therefore, the waveform desired had been successfully generated.

After obtaining the waveform, the 32 sets of data will be divided into two major sets, that is 16 x-axis data, and 16 y-axis data. This $16 \times 16$ data will be processed, and 256 data will be generated. These 256 data represents the $16 \times 16$ pixels that will form a tomogram. Among the 256 pixels, only 240 pixels are located inside the circle of the tomogram. In order to save some time, the data at the pixel, out of the tomogram circle are not send to the PC. The algorithm in processing the data involved adding the first data of the x-axis to all the data at the y-axis, and therefore, generates a set of 16 data. Then, the second data of x-axis will be added, and so on. At the end of the process, the data will be added with a change of value. When the data exceed the maximum or minimum value (determined by programmer), the changes of value will be replaced by another proper one.

### *4.1.2 PIC16C765 as USB Controller*

The USB device in this hardware will be treated as a Human Interface Device (HID) in PC. HID is useful because HID drivers are generally installed and supplied with a modern operation system. However, since HID is a general device with general drivers, there are limitations to what a HID device can do. Many sub routines are needed in order to use the USB feature in PIC16C765. This program is generally divided into two parts, the function generator part and the descriptor part. USB descriptor is a very important part in USB communication due to the fact that the PC will recognize what type of device it is from the descriptor. The PC will recognize the device from the vendor ID and product ID of the device. This ID is thought as the password that links the PC and the device, so that the PC will not get any wrong information from wrong sources, if there are many devices attached to the PC [4]. Besides, the descriptor also tells the PC about all the details in sending and receiving data, such as endpoints to transfer data and the total amount of data.

## 4.2 Programming Language in the PC to Communicate with Hardware

In order to take data through USB port, several procedures are needed. Taking data from the USB port is not the same as taking data from the parallel or serial port. USB port is just an address. Therefore, the process of getting data from the port is much more complicated then taking data from other ports. Several built-in functions in windows Device Development Kit (DDK) are used in the program, in order to get the data from USB port [2]. Window DDK need to be installed before all these functions can be used.

Visual Basic was used as the programming language to draw the tomogram in this project. Visual Basic is a user friendly, and easy to program language [6]. Besides that, it has good graphic user interface. This programming language is the language that was used to gather the data from hardware, and therefore, it is more convenient to use this language to do the display as well.

When the form of the tomogram is loaded, it is filled with colours that are predefined in the code. An ellipse is formed for the tomogram to be drawn up here. This tomogram is divided into 256 small pixel, which is equal to 16 x-axis data × 16 y-axis data that the hardware will send. The number of sensor is also written in the form from the code written. A colour bar is drawn too at the loading of the form. This vertical colour bar is divided into 256 small lines, which means the relevant colour with the voltage. The higher the voltage, the lighter the colour goes. However, this colour bar is hidden at the form load and will only be displayed when users click the button “show colorbar”.

Users will only see the form for the tomogram when they click the button “show tomogram” in the data form. However, the tomogram form will only loads its initialized value. To see the real tomogram data, users need to click the “start” button.

When the "start" button is clicked, the form will get the data from the data form, and then convert it into colours, one data by one data, into the tomogram.

Lines method is used to draw the tomogram interface and also the tomogram. With this method, a programmer can easily draw lines with different width [6], from any part of the form to another part, just by writing a few lines of codes. As for the display of the tomogram, every data that had been gathered from the data form will be arranged according to the tomogram. These data will be converted into colours and displayed dot by dot in the tomogram. Figure 4 is the interface for the tomogram form.

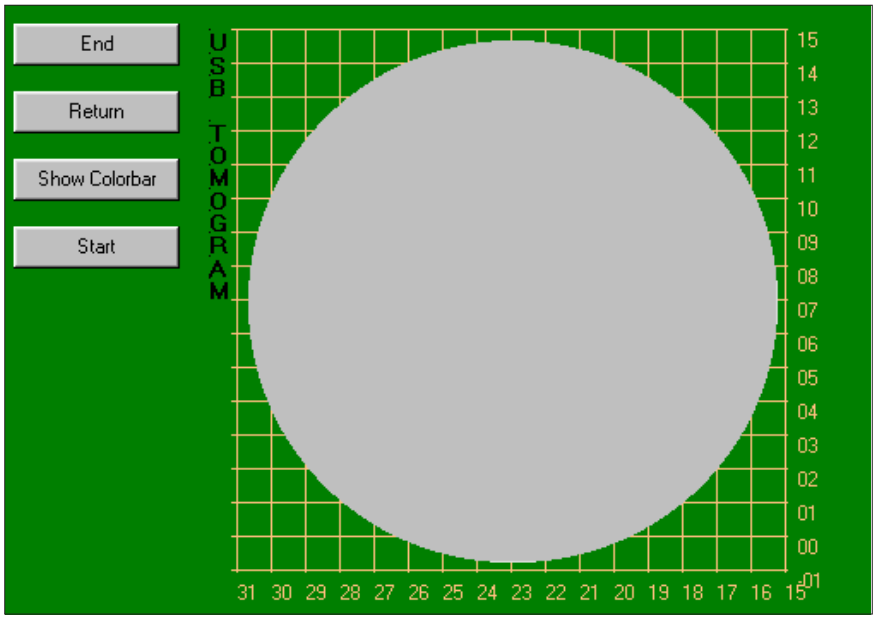


**Figure 4** Interface for tomogram form

## 5.0 RESULTS

The computer will detect the USB device on its own when the device is attached to the USB port of the computer. However, the detection may be a successful or an unsuccessful one. Unsuccessful detection will cause the USB device to be unknown to the PC. The detection of the USB device can be seen under the subtitle USB device of device manager in the system of the PC from the control panel. If the device is detected as an unknown device, the PC will not recognize the device, therefore, it is unable to install the appropriate driver for this device. Figure 5 shows a successful device installed.

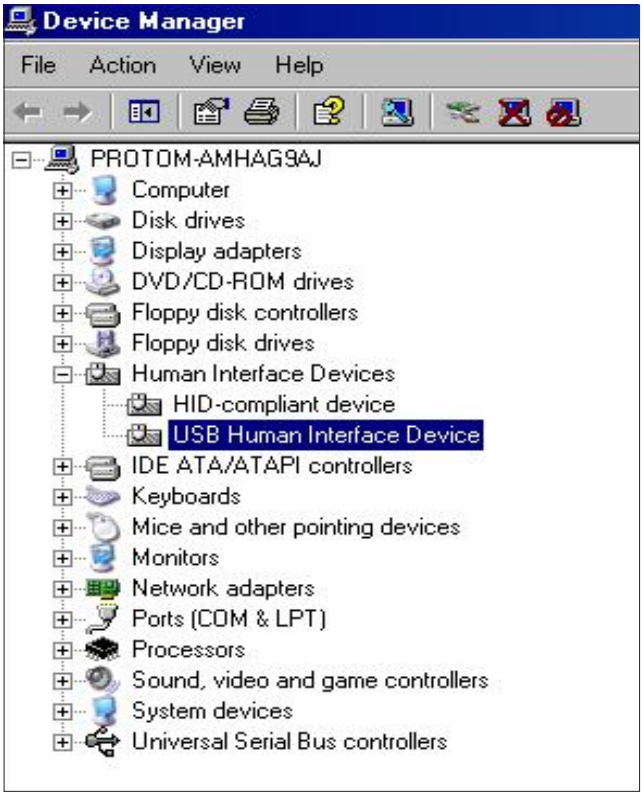


**Figure 5** Device manager with USB HID installed

Generally, if the PC had successfully detected the USB device, the Visual Basic program will also detect the device and collect or send data successfully. However, if the vendor or the product ID of the program and the device is not compatible, the program will not detect the device. Instead, it will show a message of "device not found" and the data will not be displayed at all. However, if the program successfully connect the bridge between the hardware and the program, data can be sent continuously. The values of the column in the form will change continuously, according to the data generated by the hardware. The data will stop changing if the user plug out the USB cable, which proves that the data is sent to the PC through the USB device. Figure 6 is an example of a successful data sending.

The tomogram drawn is based on the value in the column of the data form. The initialize value is 0 if there is no data in the column. Therefore, if the "start" button in the tomogram form is clicked while there are no data in the data form, the whole tomogram will be in black colour, which means that the data fetched from the data form is zero. The colour of the tomogram will change according to the data in the data form. Changes in data in the data form will change the colour in the tomogram too. Figure 7 shows the tomogram with data from the hardware.

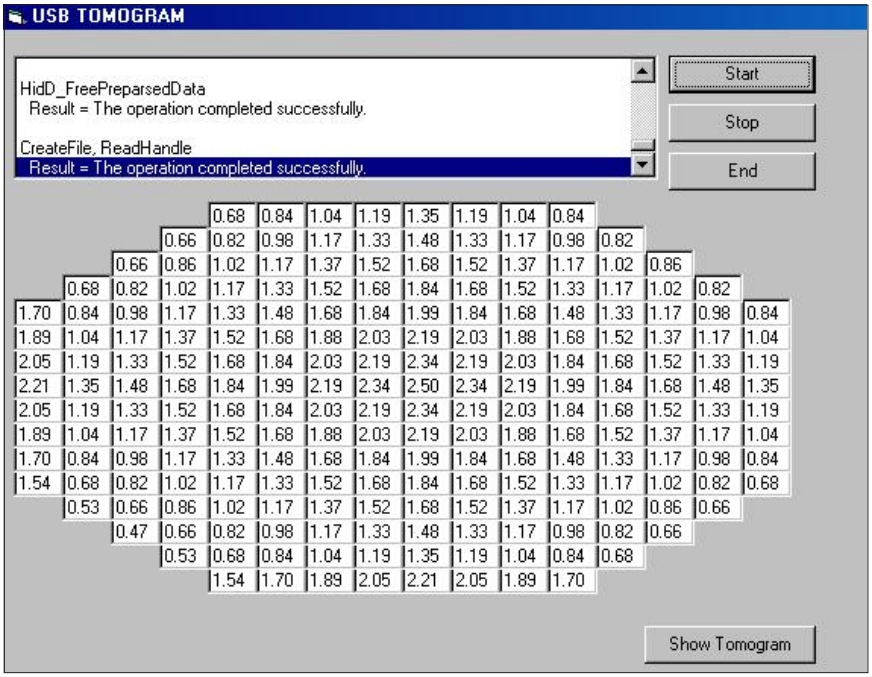


**Figure 6** Data form with successful sending data from USB

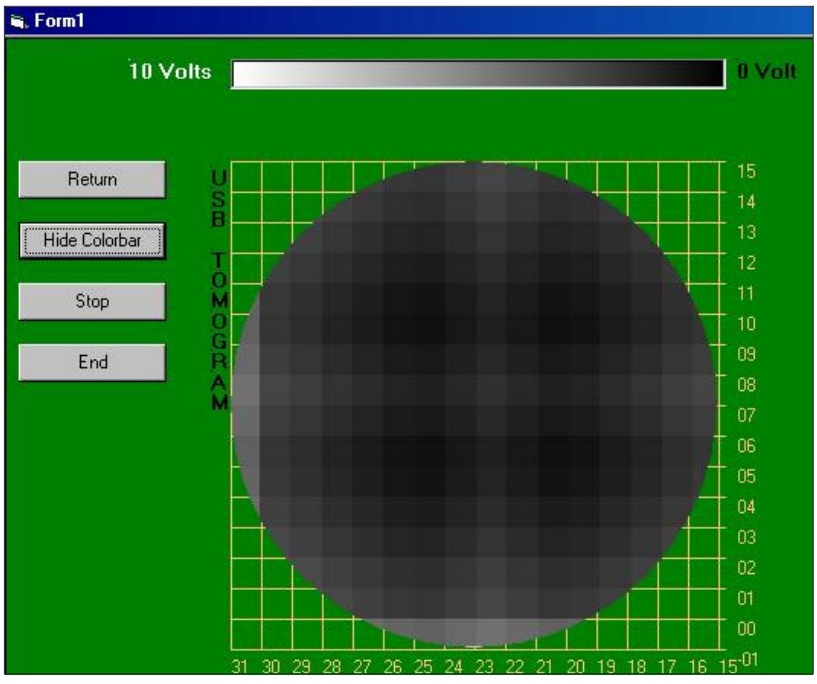


**Figure 7** Tomogram with data

## 6.0 CONCLUSION

There are still plenty of rooms for improvement and future development. It is a good suggestion to have a proper printed circuit board to house the entire circuit. With this, the system will be more stable, with smaller circuitry dimension, and more convenient to ship around.

A high speed USB device should be replaced with a low speed device, in order to better the performance of the system. A high speed USB will run 8 times faster than a low speed device, and hence should be implemented in the optical tomography system that needs high-speed data acquisition system.

PIC16C765 is not a suitable device as a micro controller as well as a USB controller. The performance of the system is lower down because this micro controller is already overloaded with too many work. Therefore, we suggest that two different chips, one as a micro controller and the other as a USB controller should be used. In fact, three chips, one for data processing, one for data transferring to USB controller, and the last one as a USB controller should be used to get the best performance of the system.

It will be very useful if the software of the system is developed in a higher speed programming tool. Visual Basic is not capable enough to handle a high speed USB device [6]. Therefore, a better programming tool, such as Visual C++ should be used as the platform of collecting data, as well as the platform to draw the tomogram.

## REFERENCES

[1] Sams. 1998. *Visual Basic 6 in 21 Days.* A Division of Macmillan Computer Publishing.
[2] Microchip. *PICmicro Mid-Range MCU Family Reference Manual.* 2000.
[3] Compaq, Intel, Microsoft and NEC Corporation. *Universal Serial Bus Specification* 1998. U.S. : Revision 1.1
[4] Chan, K. S. 2000. *Image Reconstruction For Optical Tomography*. Tesis Sarjana Muda. Universiti Teknologi Malaysia.
[5] Rehman, A. U. 2000. *CCD Video Camera Interfacing On USB Port Through Fiber Optic*. Tesis Sarjana Muda. Universiti Teknologi Malaysia.
[6] Chan, Y. Y. 2000. *Computer Based Motor Control Through USB*. Tesis Sarjana Muda. Universiti Teknologi Malaysia.