Computer Interfacing Through the USB Port
(Building a USB Device)

Abioye, Ayodeji Opeyemi
Department of Electrical and Electronic Engineering,
School of Engineering and Engineering Technology,
Modibbo Adama University of Technology (MAUTECH), Yola.

Nathan David
Department of Electronic Engineering,
Faculty of Engineering, University of Nigeria, Nsukka.

ABSTRACT

Traditional computer interfacings (for robot control, measurements, reading and logging data, etc.) have been mostly achieved through the serial port or parallel port. However, the modern trend utilizesthe USB in interfacing with the computer system. For instance, mobile PCs such as the laptops, mini-laptops, etc. no longer come with the serial or parallel port; hence project development using these computers becomes a challenge to the project developer whose interface knowledge is limited to the serial or parallel port. This is even more challenging for developing countries where projects of this kind would require importing some advance components (development boards) at extra expenses. This paper focuses on practical USB interfacing using PIC18F4550 as the microcontroller for the device; mikroC as the C language development environment for the PIC18F4550; and visual basic (VB9) in developing the windows based application. A USB test device was built as well as its windows based test application. The computer is used to generate control signals to switch ON or OFF some LEDs. This work presents students, lectures, researchers, industrialists, etc. in institutions and industries with an alternative to computer interfacing.

Keywords: USB, Firmware, Software, Computer Application, microcontroller, Computer Interfacing.

1 INTRODUCTION

Most of the computer interfacing projects being developed by undergraduates in Nigerian and most other developing countries alike are serial port based, even in this era of USB interfacing. This problem arises due to the difficult experienced in implementing the USB protocol. The work presented in this paper is a summary of the procedure used to achieve the USB interfacing by the use of available resources in Nigeria and most developing countries, were advance components mostly have to be imported (with extra expenses).

Specification for the USB is specified by the USB Implementers Forum (USB-IF).[1] USB has had several improvements over the years from USB1.0 through USB1.1, USB2.0 to USB3.0. This work is centred on USB2.0 which is more popular. USB3.0 is relatively recent and hardware is still being developed to meet it requirements.

This work presents students, lecturers, researchers, industrialists, etc. in institutions and industries with an alternative to computer interfacing. Advancement beyond the traditional serial and parallel port gradually becomes important as most mobile PCs (Laptops, Notebooks, Mini-laptops, Net-books, etc.) these days only come with the USB port for interfacing. This paper aims at encouraging further

advancement in electronics (in developing countries, like Nigeria) which cannot be achieved without embracing the current trend and then using it as a take-off point to greater heights.

In order for this paper to achieve it aim, it is focused on practical USB device interfacing. The knowledge of the Microsoft's Visual Basic programming language and Microchip's PIC microcontrollers (18 series) would be required. [2]

The project requirements are: Visual Studio 2008 software (the development environment for Microsoft's Visual Basic programming language), MikroC software (a development environment used for the Microchip PIC18F4550 firmware development in C language), and a computer system running on the Windows operating System (preferably windows XP and above). This project uses windows library (.dll), utilities, and interfaces. This project was developed and tested on windows 7 though.

2   MATERIALS REVIEW

Few materials have been written on USB. One prominent material is the book written by Jan Axelson.[3] It explained in details how the USB really works. Some of the library functions he wrote were utilized in this project.

Microcontrollers could be programmed in high level languages such as C. The C language is developed by different third parties such as MikroC, melab C, HiTech C, etc. in addition to MPLAB C18. MikroC was used in this project. Ibrahim Dogan did a fine work on programming the Microchip PIC 18 series with MikroC in his book. This book was practical although it VB codes needed update to more recent versions for better compatibility with more recent operating system. This made it necessary to develop new lines of codes in this project. It should be noted that Ibrahim Dogan's[2] Visual basic required an EasyHid application to ease development but compatibility with recent windows operating system was an issue here.

This project had used the Microsoft visual basic programming language although several other languages like C#, Java, etc. could have been used. James Foxall[4] book was very helpful in learning the VB language.

3   USB PROJECT

This project uses PIC18F4550[5] microcontroller in the design of the control device. Microchip developed PIC18F4550 (40 pin) and PIC18F2550 (20 pin) with the ability to be interfaced through the USB. The firmware for the PIC18F4550 hardware was developed in C using mikroC.[6] The software application in this project was developed in Visual Basic using Microsoft's Visual Studio 2008.The USB test device built in this project is as shown in figure 1 below:
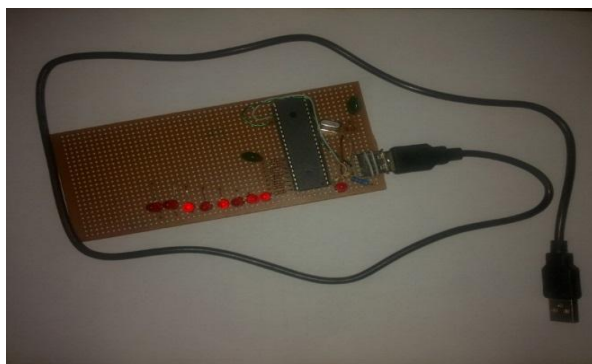

Figure 1: The Built USB Test Device

3.1 Firmware Code Flowchart

The mikroC firmware algorithm is as shown in the figure 2 below. Power is supplied to the USB device when connected to the USB port. The device is started and several initialization routines such as buffer specification, tmr0 interrupt setting, I/O port initializations, crystal pre-scaling, HID initialization, etc. are called. The buffer input is being read and displayed on PORTB of the PIC18F4550. The HID descriptor terminal in mikroC is used to generate a HID header file (USBdes.c) which is imported into the source code during the initialization stage. The rest of the program is a simple loop which is taking what is available at the read buffer and sending it to PORTB for display.
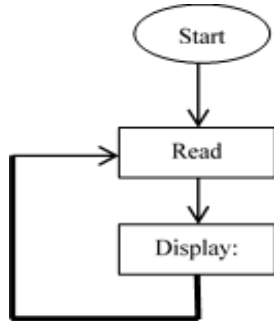

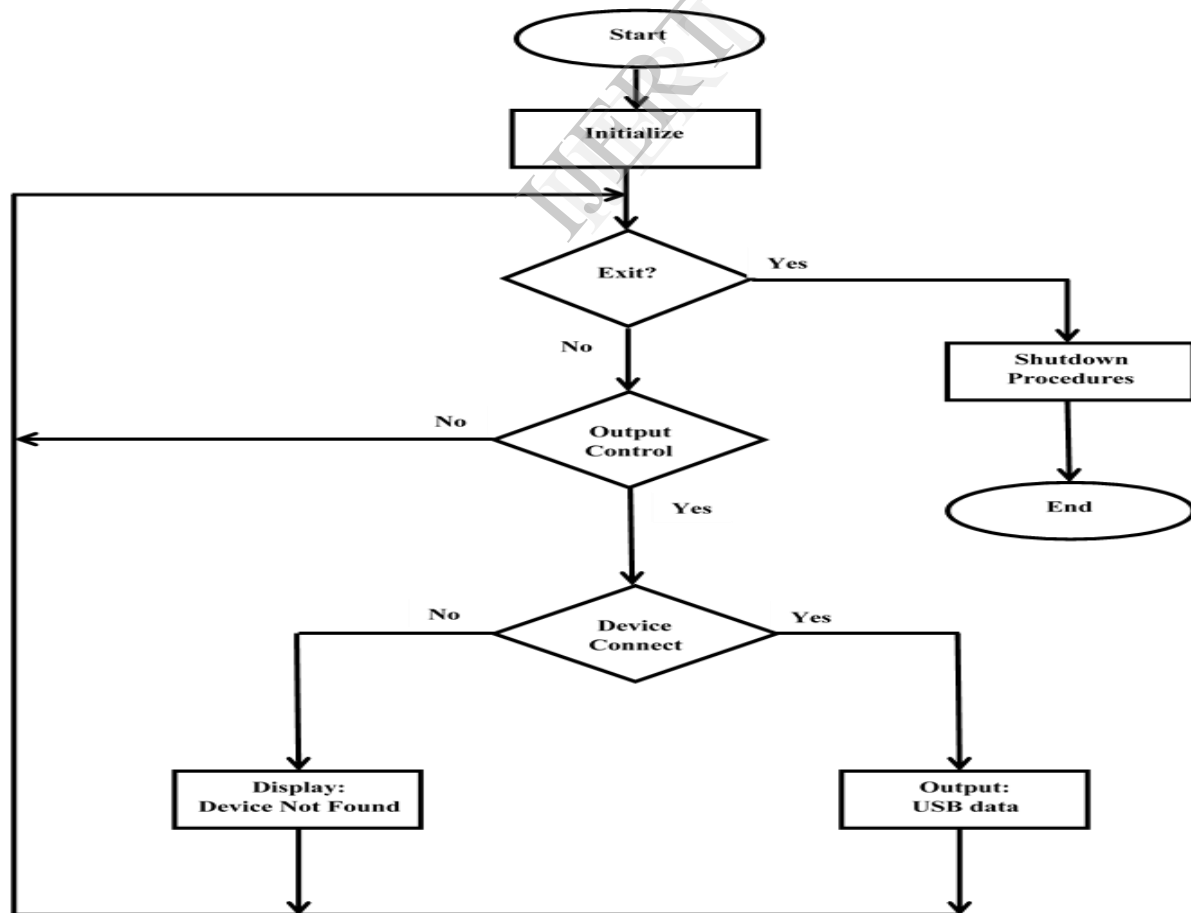*Figure 2: A chart describing the mikroC Firmware Algorithm*

Figure 3: Visual Basic USB Application Program Code Flow Chart

3.2 Visual Basic Application Flowchart

The windows and other controls are initialized on starting the application. When a control is triggered (in this case, a radio button is selected) to indicate an action (ON LED in this case), the action is converted to a specific 8 bit binary data and sent to the PIC18F4550 (which on receiving the data sends it for display at the output port). When the application window is closed, communication ceases with the USB device. Also, USB device is verified for connection before data is sent and a message is displayed in a text box "Device not found" if the device is not connected. The flowchart in figure 3 above describe the algorithm used in developing the program.

3.3 Windows Application

The windows application is shown in figure 4 below:
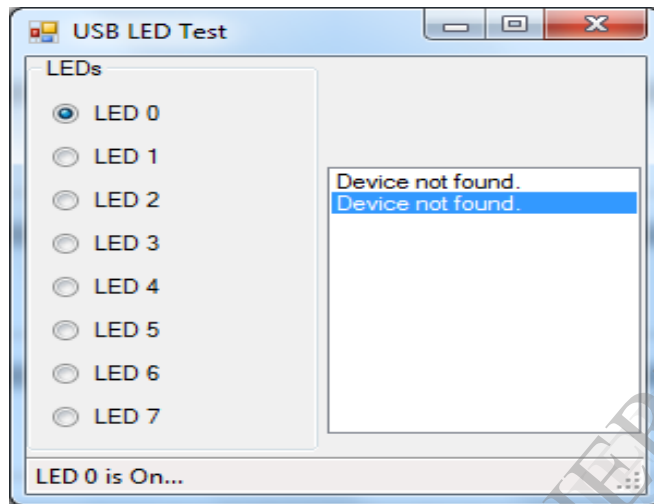

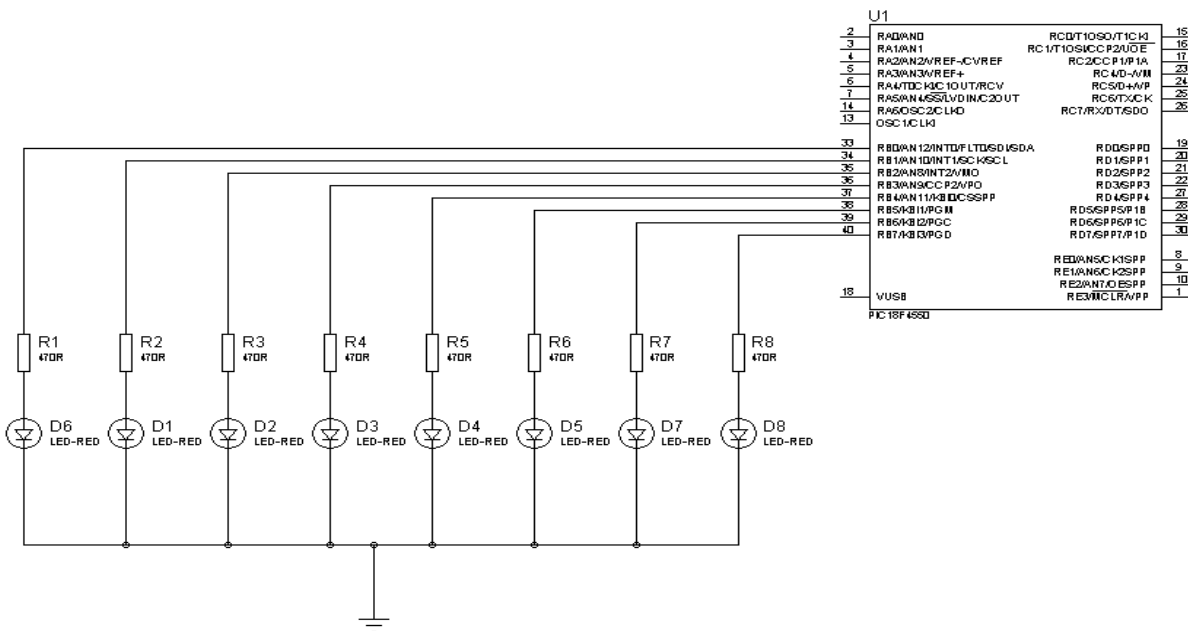*Figure 4: USB LED Test Windows Application*

Figure 5: Proteus design test circuit diagram of USB device

The USB device designed for this project was not plugged in during this test hence the warning "Device not found". This application was published into setup.exe file which is installable on any windows operating system. This application accompanies the hardware device.

3.4 Circuit Simulation Diagram

The USB device circuit diagram was designed and simulated using Proteus simulation software and is shown in figure 5 above. However, the power supply, crystal oscillator connection, and USB interfacing, have not been shown on this diagram.

4 CONCLUSION

4.1 Limitations

The limitations of this project are:

- The USB device is designed to receive data only and not send data to host. Hence computer control without feedback is being carried out (one-way communication).
- USB device requires a computer system host and has not been design to host (and communicate) with other USB devices. This is referred to as USB On-The-Go (USB-OTG).
- USB communication in this project is developed for the windows Operating system only.

4.2 Future prospects

This project is a starting step into interfacing USB based devices with other Operating system such as Linux, Macintosh, among others; and using other programming languages as well so as to have an easy universal template of USB interfacing for hobbyist, students, academics, and perhaps professionals too.

Further work is to be done on achieving two-way communication between USB devices and even more work to actualizing USB-OTG.

REFERENCES

[1] USB Implementers Forum Inc., [Online]. Available: www.usb.org. [Accessed 2 May 2012].

[2] I. Dogan, Advanced PIC Microcontroller Projects in C, Oxford: Elsevier Ltd, 2008.

[3] J. Axelson, USB Complete - The Developers Guide, 4th ed., Madison: Lakeview Research LLC, 2009.

[4] J. Foxall, Sams Teach Yourself Visual Basic 2008 in 24 Hours: Complete Starter Kit, Pearson Education, Inc., 2008.

[5] Microchip, [Online]. Available: www.microchip.com. [Accessed 2 May 2012].

[6] MicroElektronika Embedded Solutions, [Online]. Available: www.mikroe.com. [Accessed 2 May 2012].

[7] USB Implementers Forum, Inc., [Online]. Available: www.usb.org. [Accessed June 2012].