

USB Data Acquisition Using Peripheral Interface Control Microcontroller

Dhiya Mahdi Fairs, Assistant Professor
Computer Engineering Tech. Department
Technical College -Mosul

Kasim Abdullah Ahmed
Computer Engineering Department
University of Mosul

Abstract

Supervisory Control and Data Acquisition (SCADA) systems have been widely used in industry applications. This paper describes design and implementation low-cost universal serial bus (USB) Data Acquisition (DAQ) which connected with computer through USB port and tested in SCADA system, the designed Data Acquisition build using new generation of microcontroller peripheral interface control PIC18F4550 which supported with USB module and it programmed in C-Language, the developed DAQ have 13 input analog channels and 16 digital I/O lines. At personal computer (PC) side, graphical user interface (GUI) application is designed using Laboratory Virtual Instrument Engineering Workbench (LabVIEW) program as SCADA software to acquire and monitor data and control all system I/O.

The system consists of hardware part and software part. The main part of hardware is a data acquisition card, and the software part includes data acquisition card control procedures and PC software. The system hardware developed is a low-cost alternative to commercially available systems that use USB controllers. The work has been implemented for temperature sensing and controlling and the results shown currently support the hardware and software developed. It can be used for biomedical analysis, industrial temperature loggers, industrial process, instrumentation controls and in SCADA system.

Key words: SCADA systems, USB data acquisition, microcontroller peripheral interface control PIC, GUI, LabVIEW.

بطاقة اكتساب البيانات ذو الناقل التسلسلي العام باستخدام المسيطر الدقيق

الخلاصة

أصبح نظام التحكم الإشرافي واكتساب البيانات (سكادا) ذو نطاق واسع الاستخدام في لتطبيقات الصناعية. يتناول هذا البحث تصميم وتنفيذ بطاقة اكتساب البيانات منخفضة الكلفة ذو الناقل التسلسلي العام يمكن استخدامها في أنظمة السكادا الصغيرة. تم بناء الطاقة المصممة باستخدام الجيل الجديد من المسيطر الدقيق وتم برمجتها باستخدام لغة سي، البطاقة المطورة لها 13 خط إدخال للإشارات التناظرية و16 خط إدخال أو إخراج للإشارات الرقمية. في جهة الحاسوب الشخصي تم تصميم واجهة المستخدم الرسومية باستخدام برنامج مختبر المشغل الهندسي للأجهزة الافتراضية كبرنامج بديل لبرنامج سكادا التقليدية لغرض الحصول على البيانات وعرضها والسيطرة على جميع خطوط إدخال وإخراج المنظومة. المنظومة تتكون من جزأين، جزء المكون المادي وجزء المكون البرمجي. الجزء الرئيسي للمكون المادي هو بطاقة اكتساب البيانات والجزء البرمجي مكون من وحدة السيطرة وبرمجيات الحاسوب. تتميز المنظومة المطورة برخص تكلفتها مقارنة بمثيلاتها التجارية التي تستخدم الناقل التسلسلي العام. تم تنفيذ

العمل لتحسس والسيطرة على درجات الحرارة والنتائج أظهرت مباشرة ما يدعم الجزأين المادي والبرمجي المطور. كما يمكن استخدامها في تحليلات الطب الحيوي، العمليات الصناعية، والسيطرة على الأجهزة في أنظمة السكادا. الكلمات الدالة: نظام التحكم الإشرافي واكتساب البيانات، الناقل التسلسلي العام، المسيطر الدقيق. واجهة المستخدم الرسومية، المشغل الهندسي للأجهزة الافتراضية.

Introduction

The use of electronic equipment has been growing rapidly. Electronic systems controlled by software have marked great strides in today's rapidly changing environment. New electronic equipment with lots of enhanced features can be easily embedded using software and can be made to work efficiently. However, this requires careful monitoring besides the presence of the user. Some examples are: systems for monitoring parameters such as temperature, pressure in closed spaces, systems for commanding different execution elements, data logging systems and the like. A widespread operation found in the above applications is data acquisition^[2]. Data from the external world is read, processed, used for decisions and eventually, memorized. The digital systems may be PCs with dedicated interfaces through which a good user-friendly environment can be provided. There are different ways of connecting a microcontroller based system to a PC: the industrial system adapter ISA or peripheral control interface PCI connector, the serial port, the parallel port, or the USB port. USB based data acquisition systems can be developed using USB controllers like ISP1181 connected to P89c51RD2 microcontroller^[2], using PIC16F877 microcontroller^[3] or using T232/245BM USB interface chip to microcontroller^[8,9]. Proposed system uses PIC18f4550 with inbuilt USB hardware stack. This paper describes low-cost data acquisition hardware with

USB interface basing on the PIC18f4550 microcontroller and tested in both simulation and real hardware case as SCADA system. First, a short review on USB has been discussed. Second, the developments of the system and a brief on hardware and software resources used have been discussed. This is followed by system integration and validation results. Finally conclusions and the development directions are given.

USB in Brief

USB is a serial bus conceived for an easy and expandable way for connecting peripherals to PC. Its final purpose is to establish the unique interface for the PC peripherals. Similarly different peripherals are connected with the same cables and connectors. The specific features of each peripheral are assumed by the software. USB divides the serial compatible equipment into 2 classes: hubs and functions. A hub has connections, called ports, where the functions may be connected. A function is peripheral equipment which achieves a certain type of operations. USB Speeds are High Speed - 480Mbits/s, Full Speed - 12Mbits/s and Low Speed - 1.5Mbits/s. USB specification defines four transfer/endpoint types, Control Transfers, Interrupt Transfers, Isochronous Transfers and Bulk transfers. The Universal Serial Bus is host controlled^[2]. A USB device indicates its speed by pulling either the D+ or D- line high to 3.3 volts^[3, 4]. A full speed device, pictured in figure 2(a)

uses a pull up resistor attached to D+ to specify itself as a full speed device. And in figure 2(b) a pull up resistor is attached to D- to specify itself as low speed device. These pull up resistors at the device end will also be used by the host or hub to detect the presence of a device connected to its port. Without a pull up resistor, USB assumes there is nothing connected to the bus.. The more common USB descriptors are Device Descriptors, Configuration Descriptors, Interface Descriptors, Endpoint Descriptors and String Descriptors. Enumeration is the process of determining what device has just been connected to the bus and what parameters it requires such as power consumption, number and type of endpoint(s), class of the product and the like. The host then assigns the device an address and enables a configuration allowing the device to transfer data on the bus.

Systems Design

Figure 3 shows the block diagram of PIC-USB Data Acquisition System .It uses familiar microcontroller from Microchip Technology Incorporated PIC® Peripheral interface control series called pic18f4550^[6] as the heart DAQ system. It has 40-pin, high-performance, 32 Kbytes Flash program memory, USB microcontrollers with nano watt technology , in build USB module and internal 10 bit A/D converter .It has 5 I/O ports,32 general I/O lines, which 13 input analog input lines ,19 digital I/O lines .

Universal Serial Bus Features

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)

- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers

Internal A/D converter Features

10-Bit A/D Converter: This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated, without waiting for a sampling period and thus, reducing code overhead. The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The system design passed through two parts software and hardware.

Software Design

Circuit connection and PIC programming

The system designed and tested as SCADA system in simulation using LabVIEW program instead using traditional SCADA software which represent GUI our man machine interface(MMI). Proteus 7 professional software programs from Lab center Electronics used to design and test the DAQ circuit and the source code program was written in C language and compiled using mikroC for PIC software program. The microcontroller and other components of the circuit are connected as shown in the figure5, and it need to install the virtual USB from the start>> all programs>>Proteus 7 professional software program and figure6 show the flow chart for the source code program.

The terminals of microcontroller are used as below:

PORTB (RB0-RB7) for Input (Digital Sensor)

PORTA (RA0-RA5), PORTE (E0-E2) for Input (Analog Sensor)

PORTD (RD0-RD7) for Output (Digital Sensor)

PORTC (RC0-RC2 & RC6-RC7), RA4 for LCD

RC4, RC5 for USB

Device Enumeration

The main step for identification is USB device enumeration - this complex sounding feat is in fact the initial communication with the USB host (your PC) when the device tells the host what it is and how it wishes to communicate. USB communication is performed using 'endpoints' which send information either to the host or to the device. As well as setting up the communication channels the device must also pass its device name and two other important values: the VID and product ID. The VID is the Vendor ID and identifies the manufacturer of the device. To get your own VID you need to pay a thousand bucks or so to the USB standards body. In this example Microchip VID is used to save the expense. If you are serious about producing and selling devices you will need to register one of your own. The Product ID together with the VID they form a unique identifier for your device. When your device first enumerates Windows will store the VID and Product ID combination for the device; this is true even if you use a generic driver like the HID since it cuts down on the amount of time Windows needs to get your device ready. This is important because, if you decide to change your device's enumeration information (add more endpoints, etc.), you will also need to at least change the

Product ID before reconnecting otherwise you will get 'Device not started' errors even if you code is flawless (from experience I've noticed that Linux is not quite as fussy and tends not to complain if you keep the same VID/ Product ID combination).

In mikroC program which the source code is written in C language it possible for user programmer to create its own descriptor file for the microcontroller from Tolls>>HID Terminal >>Descriptor, to write vendor name and product name and it saved in the same location on the program folder, I named my DAQ SCADA System as shown in figure7.

VID : Vendor ID	Special code for the vendor company
PID : Product ID	Product number
Report length	number of bytes can send and receive (maximum is 64 byte)
End points Pooling int.	average time for sending and receiving
Vendor name	Vendor company name
Product name	The name of the product which show in the task bar when it connected with the PC

And it possible to reconfiguration this information from Project>> Edit Project in mikroC program where the source program of microcontroller is written as shown in figure8. When the simulation program is run the following message is shown figure9 in task bar contain the device name which had named in descriptor table. And is the device manager is appear as USB Human Interface Device as shown below. Then we can send control signal and command to digital actuator and

message to simulation program and writing messages on LCD from the tools HID Terminal as shown in figure 10:

LabVIEW programming for GUI

LabVIEW V9 software program used to create GUI to enable the user to monitor and control the peripheral system from the screen of PC. LabVIEW is a development platform for engineers which provides graphical programming language for writing engineering-related programs. It is used here as the front panel of SCADA in this project. Programs written in LabVIEW programming language are multithreaded and exploit parallelism. Programmers use wires and function nodes to define the structure of the program. Before we describe our implementation, some important terminologies regarding LabVIEW programming need to be discussed. The structure of LabVIEW program is such that all the logical implementation is done on a block diagram using function nodes and wires to define the flow of the program [6]. The designed system has been completely developed using LabVIEW graphical programming language. Figure 11 shows the front panel of our LabVIEW program and figure 12 shows the block diagram of LabVIEW.

The main GUI using LabVIEW program divides into three sections

A. Open communication link with USB port.

B. Send Data to microcontroller.

C. Receive Data from microcontroller.

A. Communication with USB port

In LabVIEW it must open a session with the USB port and use it to send and receive data as shown in figure 13.

B. Send data from LabVIEW front panel to microcontroller. We can control 8 devices connected to the

micron roller as shown in figure 14. Also, we can write on the LCD of the microcontroller. The LCD can display 16 characters on each row.

B.1 Controlling 8 devices connected with the microcontroller from the front panel of LabVIEW on the screen of PC as shown below in figure 15. This process is accomplished by the following way:

1. Grouping the output of 8 switches in a cluster.

2. Convert this cluster to an array.

3. Convert this array to numbers from 0 to 255 depending on the selected switches.

4. Convert the number from decimal to ASCII code.

B.2 Writing on LCD. In normal case the output is what is written in 'Write Me To LCD' and when pressed on the button 'Refresh me' it sends a blank text to clear the previous text from the LCD because it contains memory as shown in figure 16.

B.3 Grouping the two previous parts. Here the two previous parts are grouped and the verification of the length of the data is made to ensure they do not exceed 33 characters, which is the sum of the special character of 8 keys and 32 characters to write on the LCD.

C. Receive Data from microcontroller.

The values of analog sensors are viewed on gauges and/or graphs and the values of input digital sensors are viewed on LEDs as shown in figure 17.

The data are received in the form of an array and elements inside the array are arranged by that which has been identified within the program microcontroller.

DAQ Card Identification in LabVIEW

The communication between LabVIEW program and any hardware device is done through NI-VISA, which is an acronym for Virtual Instrument Software Architecture and it can be

responsible for the programming and the definition of connected devices and made it available with LabVIEW. To define microcontroller (DAQ) as device in LabVIEW program the following steps are followed:

4.1 Driver Wizard opened from

- Start > All Programs > National Instruments > Visa > Driver Wizard
- select USB type of connection
- typing Vendor ID and Product IP as same as in descriptor
- Select location for saving the identification
- Finally we find the device installed and identified correctly in LabVIEW and the device appears in device managers as shown

Running and Testing the Designed System

After all mentioned we successfully run the system (SCADA) in simulation program and the interface and communication are satisfied between the microcontroller (DAQ) side and the LabVIEW (GUI) side case1 all analog sensor represented by variable resistor are easily displayed as gauges indicator also the digital sensor represented by digital switches and displayed as 8 LEDs in LabVIEW front panel in other hand we made full control on the 8LEDs connected to the output of the DAQ which can control the 8actuators through drive circuit the figure 18 show samples of 2 cases for testing the system. Case2 the 4 analog sensor connected to 4 signal generator each signal generator set to generate square, saw tooth and sinusoidal signal respectively .

Hardware Design

Hardware design consists from these stages:

Building the Card

Figure 19 Shows the USB interface that has been adapted by making use of RC4, RC5, and VDD AND VSS of PIC18F4550 microcontroller. The RC4 line (PIN23) is used as data line D- and connect to pin2 of USB B type connector and The RC5 line (PIN24) is used as D+ data line and connect to pin3 of USB B type connector, VDD pin11 is connect to pin1of USB connector and VSS pin12 is connect to pin4 (GND) of USB connector .The USB interface lines are connected to pins 2 and 3 (D- and D+ channels) of USB B Type connector. osc1 (pin13) and osc2 pin(14) connected to crystal20MHZ frequency and connected to ground across two capacitor each of 22pf . VUSB (pin 18) connect to ground across 1nf capacitor. All other ports of microcontroller are used, 13 lines for input analog channels, 8 lines for digital sensor and 8 for digital output as shown in figure 19(b).

Download hex file which contain source code program in into the microcontroller

Testing the DAQ in real hardware

The system tested in real hardware it used to read two signals from variable resistor and 8 digital signals from digital switch and receive control signal from the user from front panel and it accomplished successfully figure 20 show the proto type card connected with PC via USB port.

Conclusions

PIC microcontroller based USB Data acquisition has been designed, programmed, implemented and test in SCADA system. The system hardware developed is a low-cost alternative to commercially available systems that use USB controllers because it cost only 20\$. Using a unique combination of hardware and software with optimum

utilization of the resources, a low cost Data Acquisition System has been developed to data logging and supervisory control in mini industry process and monitor voltage variations from analog channels, for temperature controlling and the like. The experimental result shows that the system can work stably ,effectively and the results show currently support the hardware and software developed. Further, this system can also be used to increase the USB transfer speed. There is a possibility of continuous development of the software which can be adapted to various transducers, to calculate and display various parameters.

References

1. Vinh Ich Nguyen , Watit Benjapolakul , Krisada Visavateeranon, “A High-speed, Low-cost and Secure Implementation based on Embedded Ethernet and Internet for SCADA Systems” , pp.1692 -1699 , SICE Annual Conference 2007.
2. Popal,M.; Marcu,M.; Popa,A. S. “A microcontroller based data acquisition system with USB interface”, Electrical, Electronic and Computer Engineering,2004. ICEEC
3. Axelson, J., “USB Complete: Everything You Need to Develop Custom USB Peripherals”, Lakeview Research, 2001, ISBN 09655081958
4. Craig Peacock, USB in a nut shell - Technical document from <http://www.beyondlogic.org>
5. PIC18F2455/2550/4455/4550 Data Sheet from <http://ww1.microchip.com/downloads/en/DeviceDoc/80478a.pdf>
6. Jeffrey Travis, Jim Kring, LabVIEW for Everyone: Graphical Programming Made Easy and Fun, Third Edition
7. Rick Bitter, Taqi Mohiuddin, Matt Nawrocki , LabVIEW:Advanced Programming Techniques, Second Edition
8. Ziad Salem, Ismail Al Kamal, Alaa Al, Bashar “A Novel Design of an Industrial Data Acquisition System” --0-7803-9521-2/06 IEEE 2006.
9. S. Thanee S. Somkuarnpanit and K. Saetang -FPGA-Based Multi Protocol Data Acquisition System with High Speed USB Interface. ISSN: 2078-0958 IMECS2010.

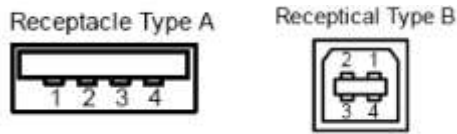
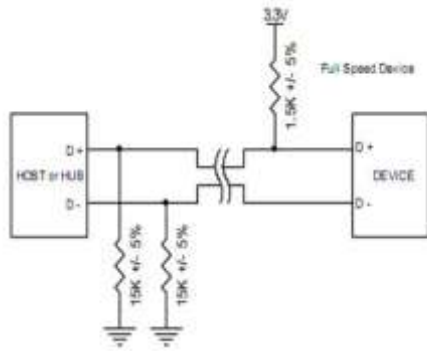
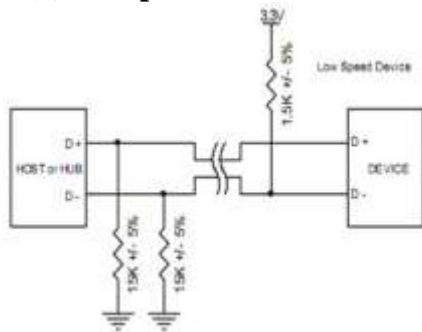


Figure (1): USB Connector Types

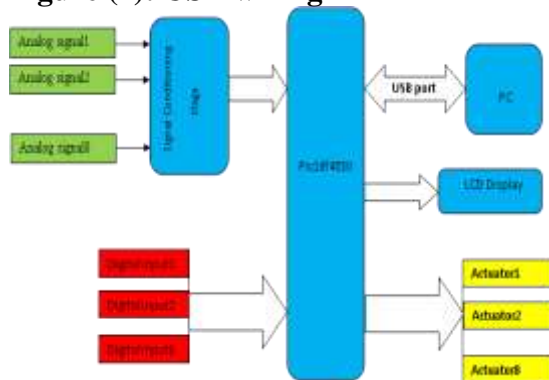


(a) Full speed

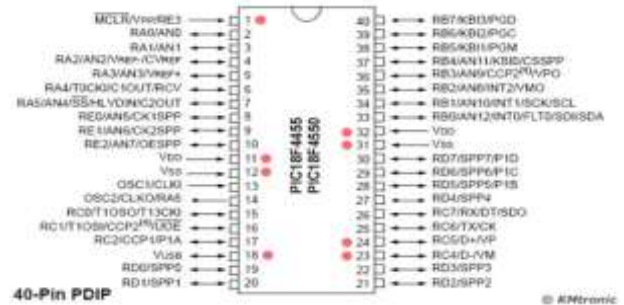


(b) Low Speed

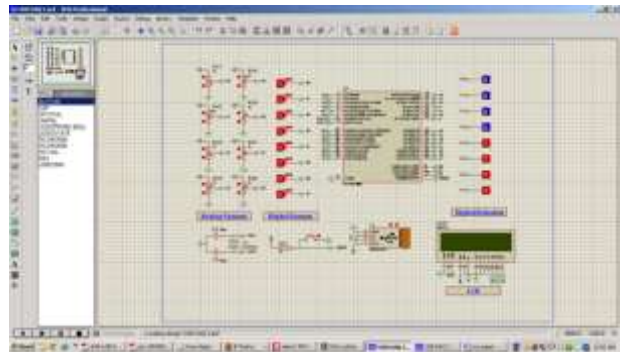
Figure (2): USB wiring



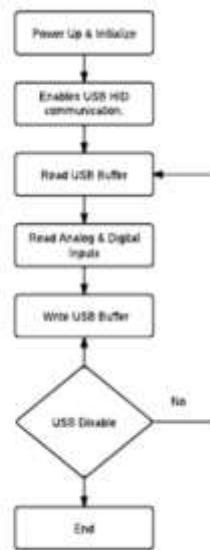
Figure(3): Block diagram of PIC-USB Data Acquisition System



Figure(4): Pin diagram of pic18f4550 microcontroller



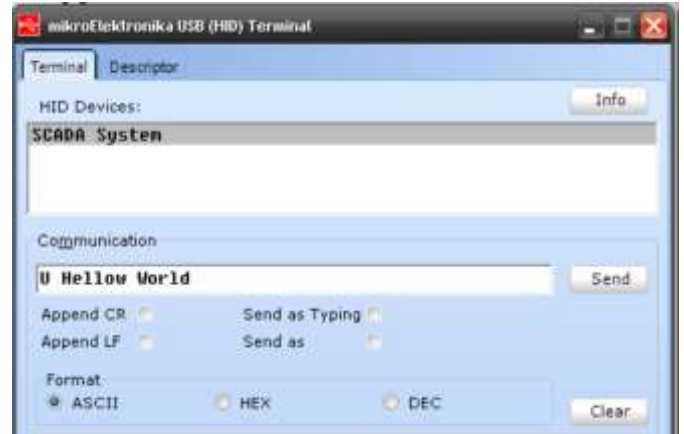
Figure(5): Circuit diagram of DAQ component



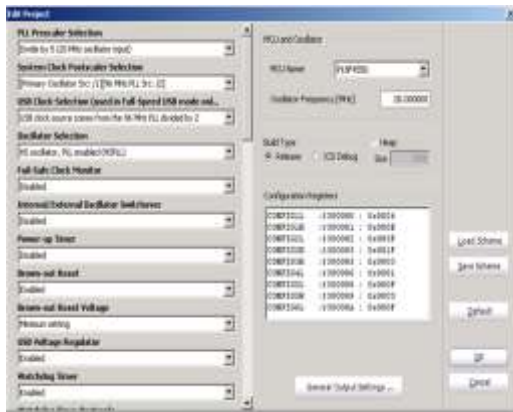
Figure(6): Flow chart of Source code program



Figure(7): Creation of descriptor on mikroC program



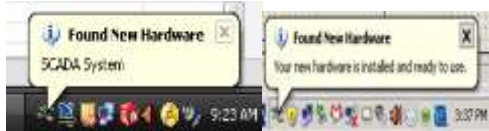
Figure(10):Interface window between DAQ and user



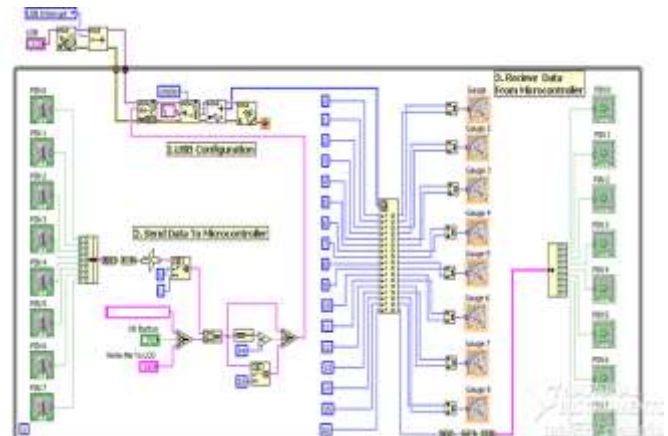
Figure(8):Configuration form on mikroC program



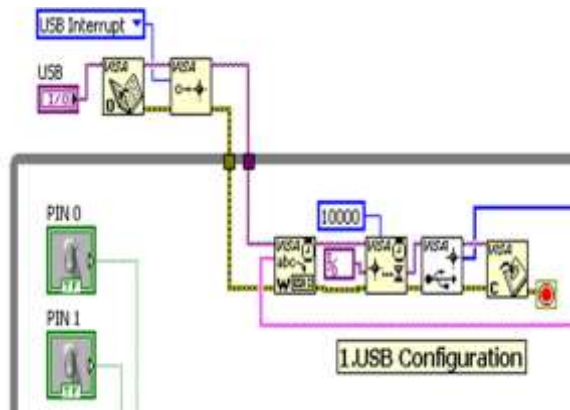
Figure(11): Front panel of main LabVIEW program as GUI



Figure(9): PCs task bar appears message when detect the DAQ



Figure(12): Block diagram of main LabVIEW program



Figure(13): Open session in LabVIEW

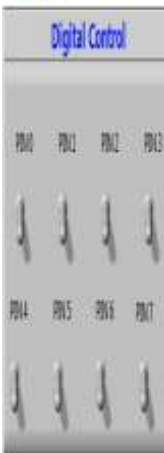
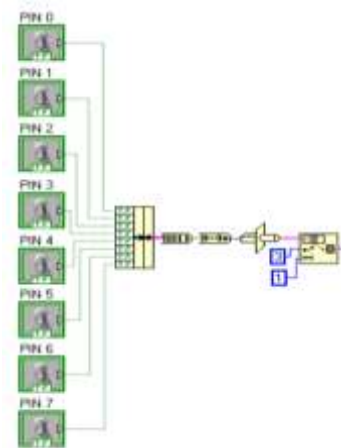
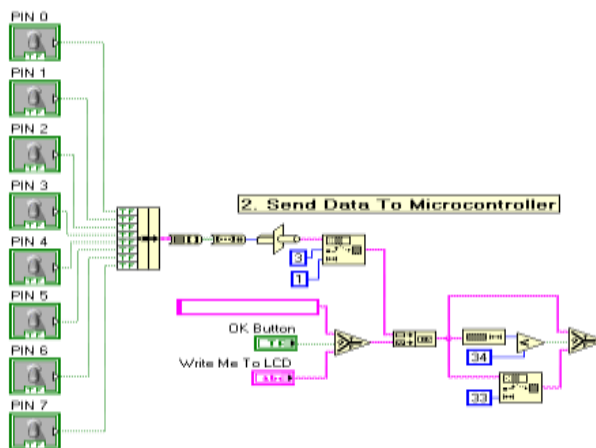
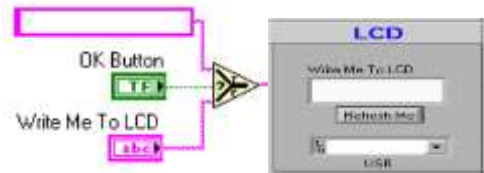


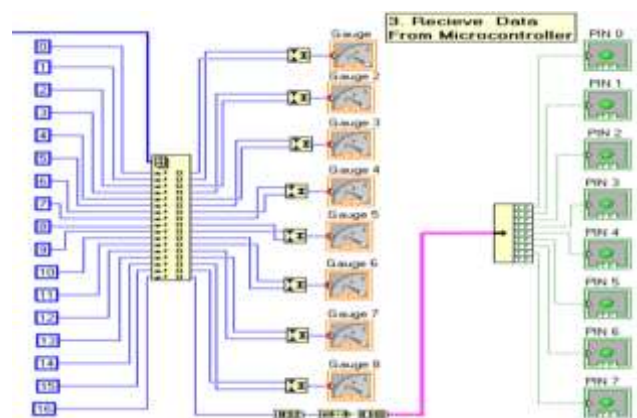
Figure (15): Controlling 8device connected to the microcontroller



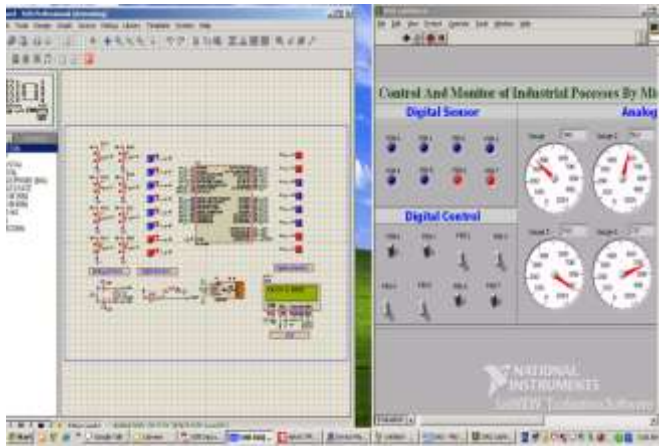
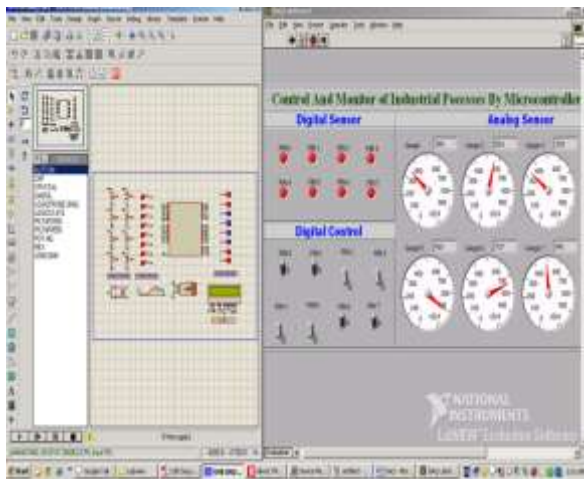
Figure(14)- Controlling the system from PC



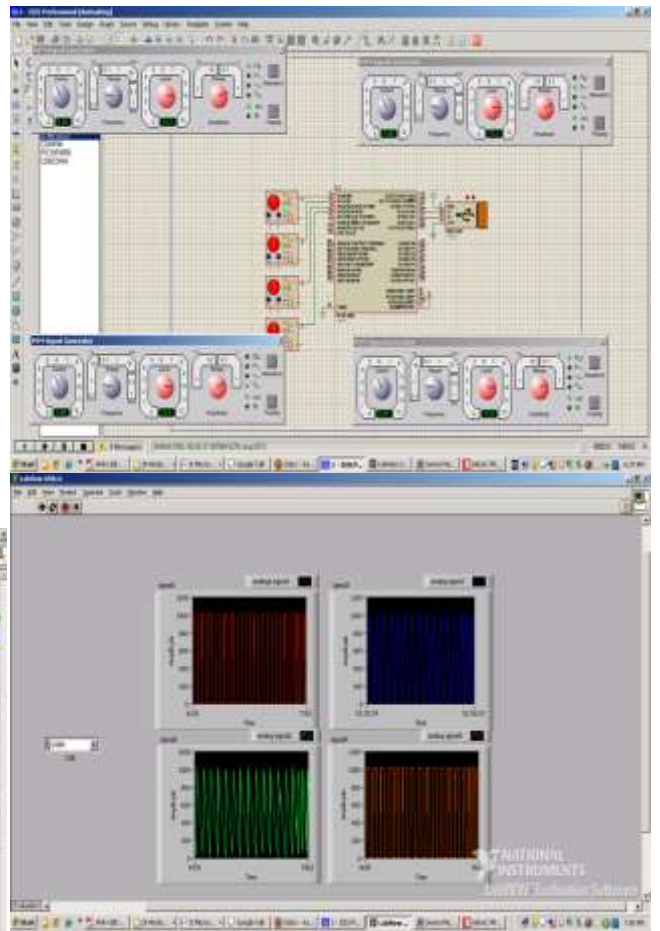
Figure(16): Writing on LCD



Figure(17):Receiving the input signals from microcontroller

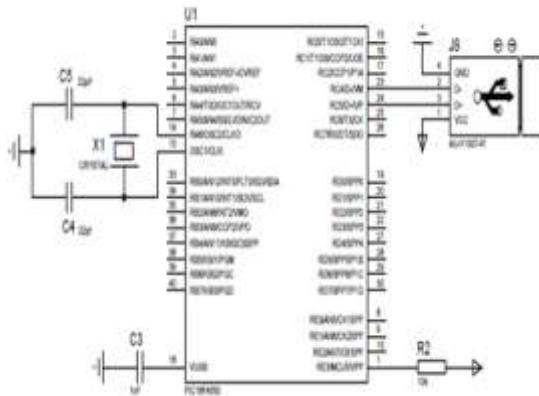


(a) Case1

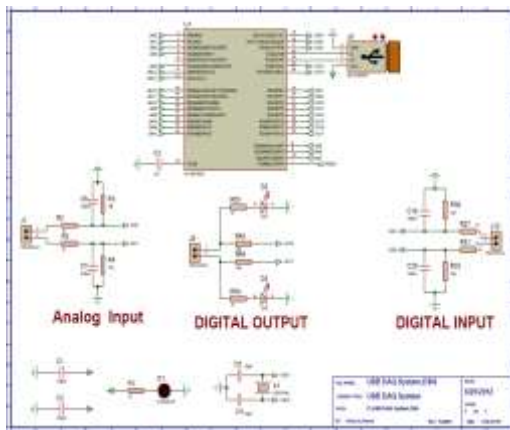


(b) Case2

Figure(18): GUI samples of system under running



(a) USB interface with PIC18F4550



(b): Wiring I/O lines
Figure (19): Wiring diagram



Figure (20): Proto type data acquisition card