CERTIFICATION OF APPROVAL

Analog and Digital PC Oscilloscope

by

Mohd Fariz bin Othman

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Ms. Salina Mohmad
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

DECEMBER 2006

ABSTRACT

This Final Year Project is about designing an Analog and Digital PC Oscilloscope. To realize this, software has to be design that has the capability of acting as an oscilloscope that can be installed in normal PC. The main focus of this project is to display the waveform of digital and analogue signal. The software that been used for this project was Visual Basic software. Oscilloscope is very useful in determining the response of an electric circuit in terms of finding the signal response. But sometimes for a small / mini project, one did not really need to get or buy an oscilloscope to see the response of the circuit. Since majority of students / people have their own PC, it is very useful to have a program that can turn the PC to an oscilloscope. Hopefully, the design of this PC Oscilloscope will assist students in their learning environment to be a better learner especially in the electronic engineering area.

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Mohd Fariz bin Othman

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

A/D	Analog-to-Digital
ASCII	American Standard Code for Information Interchange
CMOS	Complimentary Metal Oxide Semiconductor
CPU	Central Processing Unit
CRT	Cathode Ray Tube
FYP	Final Year Project
GUI	Graphical User Interface
I/O	Input/Output
IC	Integrated Circuit
PC	Personal Computer
PIC	Programmable Integrated Circuit
SCI	Serial Communication Interface
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
UTP	University Technology Petronas

CHAPTER 1: INTRODUCTION

1.1 Background of Study

Oscilloscope can be categorized as one of the important equipment in the world of electronic. Many purposes can be point out especially in its usage. As a student in engineering university, most of them use it to during experiment, circuit testing and in project. Some can display two waveforms and even more depending on the types and models of the oscilloscope.

1.2 Problem Statement

1.2.1 Problem Identification

Since majority of the students have their own PC, it is quite useful to have their own PC turning into an oscilloscope. It might not have the complete specification like the normal oscilloscope but still can be used for displaying waveform as a response from an electronic circuit.

1.2.2 Significant of Project

This project can bring some advantages to the student including cost saving from buying normal oscilloscope that is quite expensive. Beside that, this project can also reduce time spending at laboratory while doing circuit testing as not all laboratories open for 24 hours. As this work can be done at home, students can spend extra time for their work and research. Additionally, it is also easy to export the data to standard PC software such as spreadsheets and word processor as this software is running in PC. Furthermore, the ability of the software to control the instrument by running a custom program on the PC and easy portability when used with a laptop that can be used everywhere.

1.3 Objective and Scope of Study

1.3.1 Objective and Scope

a. *Understanding the full function of oscilloscope*

In order to design this software, research has to be done especially on how the oscilloscope works. This information has to be mastered to get the full understanding on the equipment

b. *Design a PC oscilloscope that can display analogue and digital signal waveform*

This main challenge to design the software comes where the oscilloscope will have to detect both signals. Since normal PC uses digital signal, it might be the main challenge to ensure the software can detect the analogue signal. There will be some conversion for the signal entering the PC until it is detected by the software. The use of analogue to digital (A/D) converter might be used especially to transfer the signal from the test circuit to the PC. Then, it has to be reverted back to analogue signal again. A lot of research and programming has to be done to make the software works.

c. *Develop a test circuit that can produce analogue and digital signal*

This project not only includes software designing but also hardware. As the software has completed, a test circuit is needed to produce the signal waveform to the PC.

1.3.1 Feasibility of the Project within the scope and Time Frame

a. *Feasibility of Project*

Mostly, the knowledge that will be implemented in the project was related with Electrical & Electronics Engineering courses in UTP. Therefore, this project is fit to the field and will not entertain quite problems to be completed.

b. *Time Frame*

As plan, the software will be designed in the first semester on the Final Year. After finishing the software design, it will continue with the hardware design in the second semester. The project completion will due on time. Gantt Chart have been created for planning schedule and time management that can be found in APPENDIX G and H. Any extra effort will be done to ensure every thing that has been planned is follow.

CHAPTER 2: LITERATURE REVIEW AND THEORY

2.1 What is Oscilloscope?

An oscilloscope or scope is an electronic measuring instrument that creates a visible two-dimensional graph of one or more electrical potential differences. The horizontal axis of the display normally represents time, making the instrument useful for displaying periodic signals. The vertical axis usually shows voltage. The display is caused by a "spot" that periodically "sweeps" the screen from left to right. A typical oscilloscope is a rectangular box with a small screen, numerous input connectors and control knobs and buttons on the front panel. Most oscilloscopes have two or more input channels, allowing them to display more than one input signal on the screen. Usually the oscilloscope has a separate set of vertical controls for each channel, but only one triggering system and time base. [1]

2.2 History of Oscilloscope

Oscilloscopes became a useful tool in 1946 when Howard C. Vollum and Jack Murdock invented the triggered oscilloscope, which would start a horizontal trace when the input voltage exceeded an adjustable threshold. Triggering allows stationary display of a repeating waveform, as multiple repetitions of the waveform are drawn over the exact same trace on the phosphor screen without triggering, multiple copies of the waveform are drawn in different places, giving an incoherent jumble or a moving image on the screen. [2]

Vollum and Murdock went on to found Tektronix, the first manufacturer of calibrated oscilloscopes (which included a graticule on the screen and produced plots with calibrated scales on the axes of the screen). Later developments by Tektronix included the development of multiple-trace oscilloscopes for comparing signals either by time-multiplexing (via chopping or trace alternation) or by the presence of multiple electron guns in the tube. In 1973, Tektronix introduced the memory tube CRT, which allowed observing single pulse waveforms rather than (as previously) only repeating wave forms. By the late 1970s, with transistor components rather than vacuum tubes, Tektronix was selling oscilloscopes on which the signal trace traveled across the screen faster than the speed of light. [2]

Starting in the 1980s, digital oscilloscopes became prevalent. Digital oscilloscopes use a fast analog-to-digital converter to produce a digital representation of a waveform, yielding much more flexibility for triggering, analysis, and display than is possible with a classic analog oscilloscope, and as of 2006 most new oscilloscopes (aside from education and a few niche markets) are digital. [2]

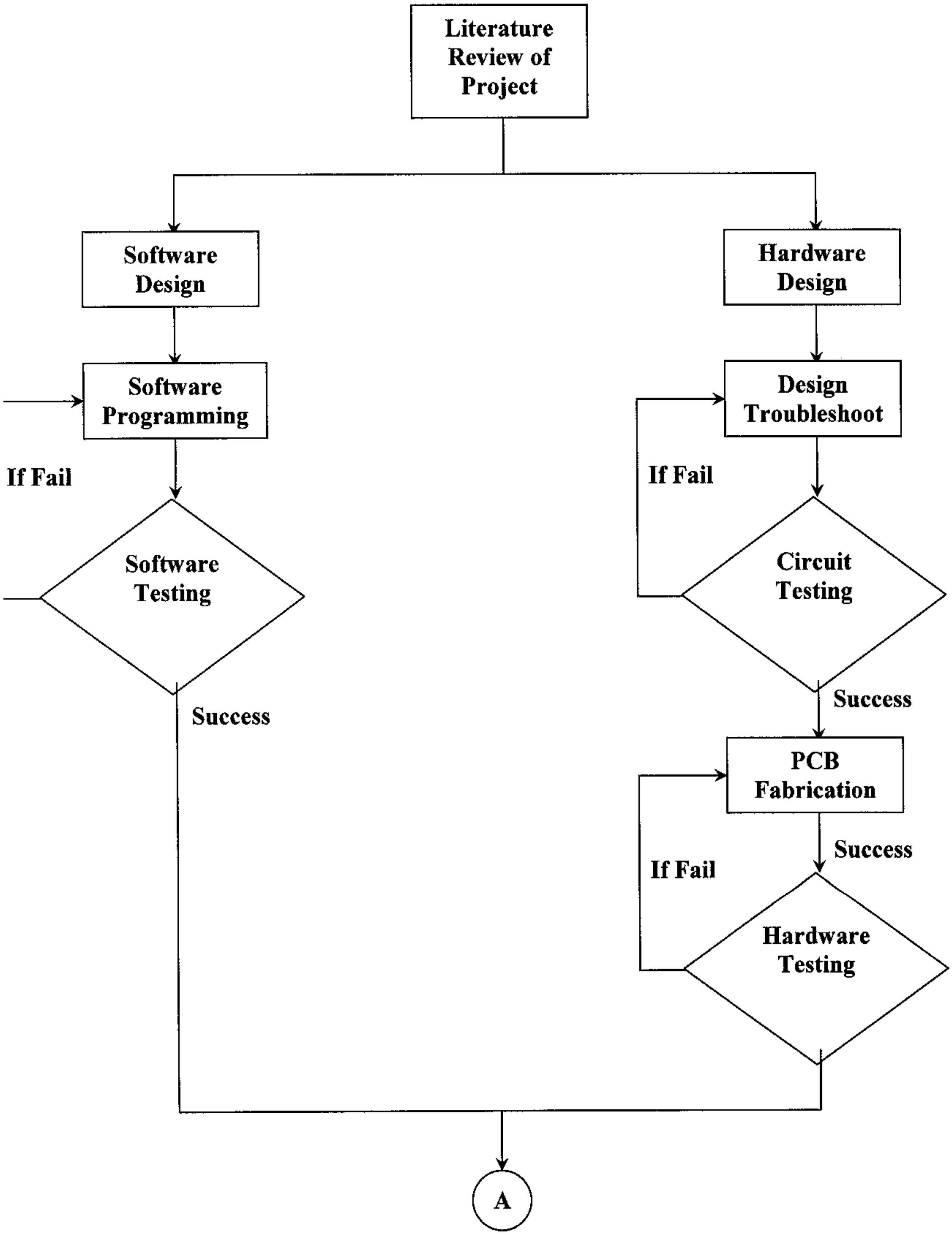
2.3 PC Oscilloscope

Although most people think of an oscilloscope as a self-contained instrument in a box, a new type of "oscilloscope" is emerging that consists of an external analogue-to-digital converter (sometimes with its own memory and perhaps even some data-processing ability) connected to a PC that provides the display, control interface, disc storage, networking and often the electrical power. The viability of these so-called PC-based oscilloscopes depends on the current widespread use and low cost of standardized PCs. This makes the instruments particularly suitable for the educational market, where PCs are commonplace but equipment budgets are often low. [2]

Just like an ordinary bench top oscilloscope, these PC Based Oscilloscopes can read frequencies, voltages, currents and more. These readings can be viewed on the computer screen. These readings can be stored in your computers memory for later viewing or comparisons. Best of all, these PC Oscilloscopes are extremely portable, and weigh no more than a few pounds, making them a great tool for any Electronic Technician who works outside of a traditional repair shop. [3]

CHAPTER 3: METHODOLOGY / PROJECT WORK

3.1 Flowchart of Project Work



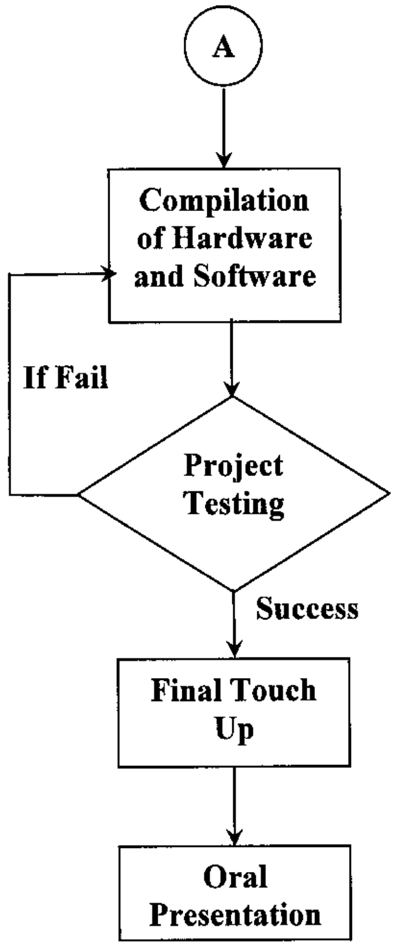


Figure 3.1: Flow Process of Final Year Project

3.2 Overview

Briefly, the system contained many stages of operations necessary in a successful model. First of all, the system contained an external circuit layout that was used to convert the analog to digital sign wave from the signal generator. Then, a serial interface to connect between the hardware and software part. Finally, is inside the computer itself, such as the software interface, and the software programming which displayed the data.

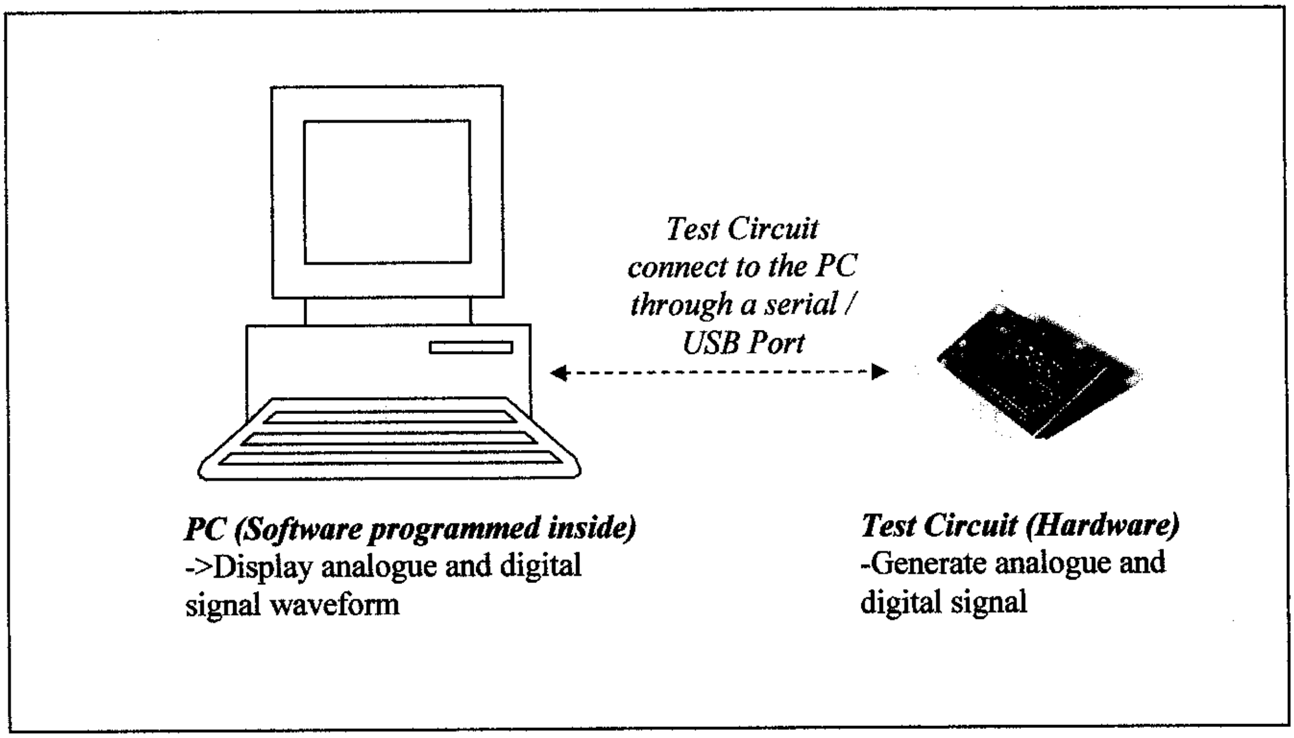


Figure 3.2: Basic Concept of PC Oscilloscope

The basic structure of this system consists of:

1. An input which consists of the External Circuitry which can produce both digital and analog signal.
2. A serial interface to retrieve the waveform from the external circuit to be transmitted to the serial port of a PC.
3. A MS Comm properties in Microsoft Visual Basic to load the input from the serial port of the PC
4. PC Oscilloscope programmed using Microsoft Visual Basic to display both waveforms loaded from the external circuit

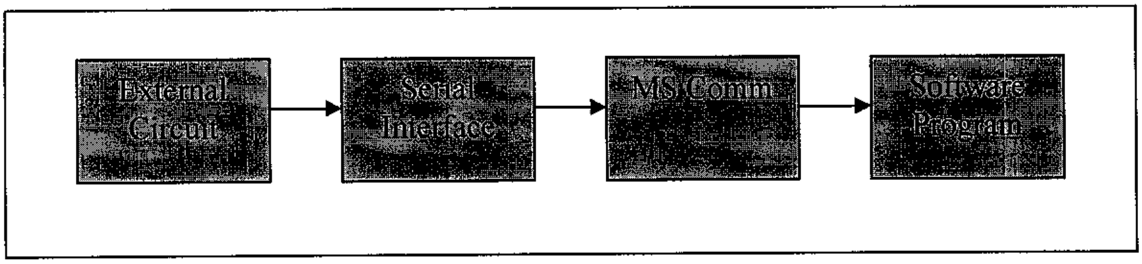


Figure 3.3: Overall Block Diagram

3.3 Hardware Design Process

This project will be describing the system software and the necessary program that allows the computer to interface with the external circuit. When starting working on the PC oscilloscope, firstly an excellent A/D converter is needed. This is to ensure the input signal from the test circuit can be read by the PC before been transmitted to the software. The software used is Microsoft Visual basic. Besides that, to design the software, a good programming skills is also required. This process will be the first step in the whole project. The I/O card was easy to choose because it provided us with the ability to upgrade and it was programmable for; read, write, or bi-directional. Also, the I/O card had the amount of bit locations to be used in our bi-directional interface programming.

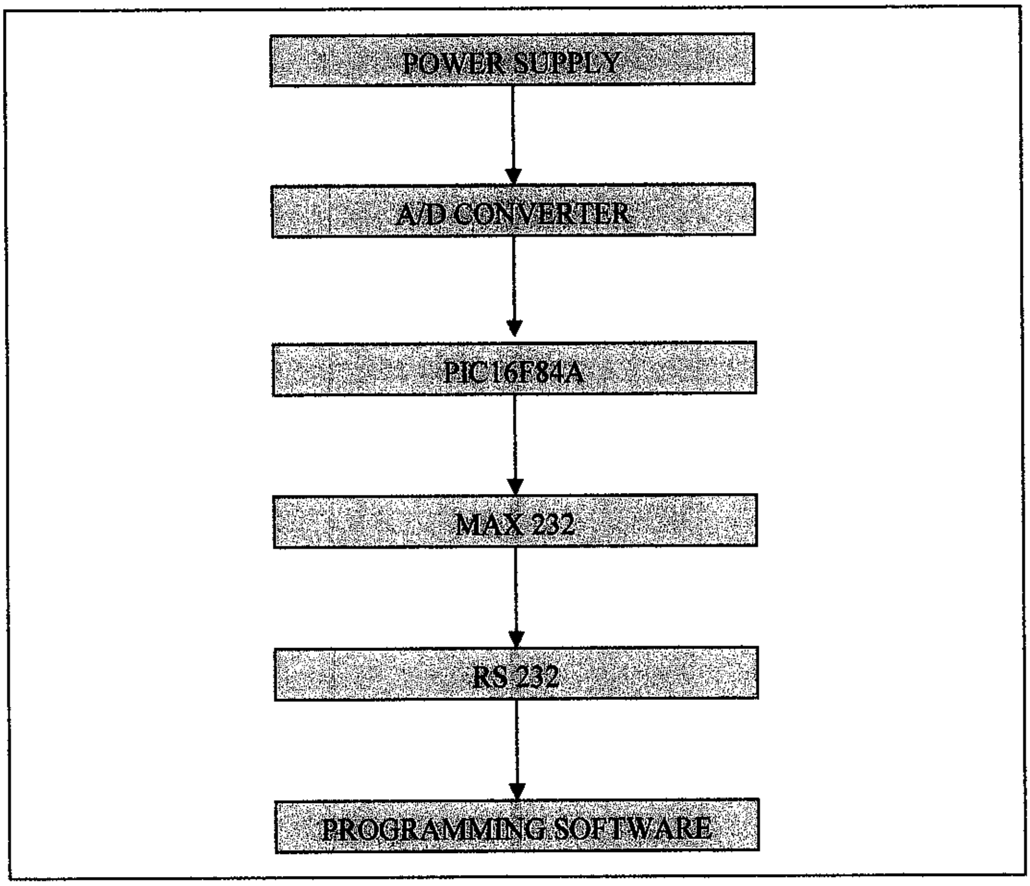


Figure 3.4: Hardware Design Process

3.3.1 Analog to Digital Converter

To display analog signal onto a digital computer, such as PC, the continuous analog voltage needs to be converted into a discrete digital number that the computer can then take and manipulate. The conversion between analog to digital is done using a basic A/D converter chip.

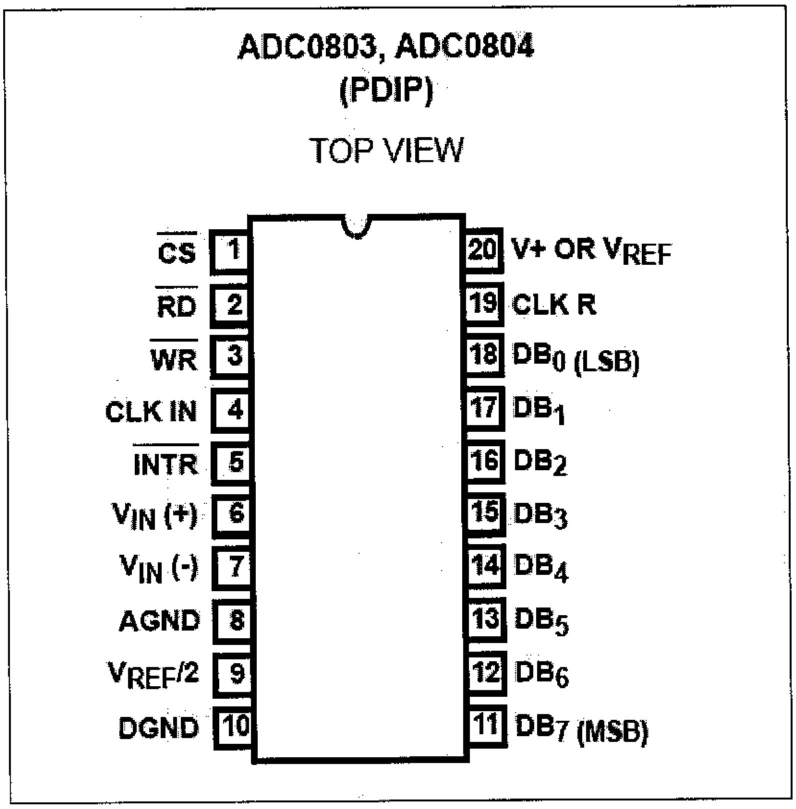


Figure 3.5: Pinout of ADC0803, 0804

In this project, an 8-Bit, Microprocessor-Compatible, A/D Converters was used. The ADC080X family is CMOS 8-Bit, successive approximation A/D converters which use a modified potentiometric ladder and are designed to operate with the 8080A control bus via three-state outputs. These converters appear to the processor as memory locations or I/O ports, and hence no interfacing logic is required. The differential analog voltage input has good common mode-rejection and permits offsetting the analog zero-input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

3.3.2 PIC16F84A

The brain of the hardware circuit is the PIC16F84A microcontroller from Microchip. PIC16F84A is an 18-bit *enhanced* FLASH/EEPROM 8-bit microcontroller. The main features of the PIC is 1024 words of program memory, 68 bytes of Data RAM, 64 bytes of Data EEPROM, 14-bit wide instruction words, 8-bit wide data bytes, 15 special function hardware registers, eight-level deep hardware stack, direct, indirect and relative addressing modes and four interrupt sources. PIC16F84A is also built in with the latest CMOS technology to make the most versatile and easy to use microcontroller for both entry-level and advanced users.

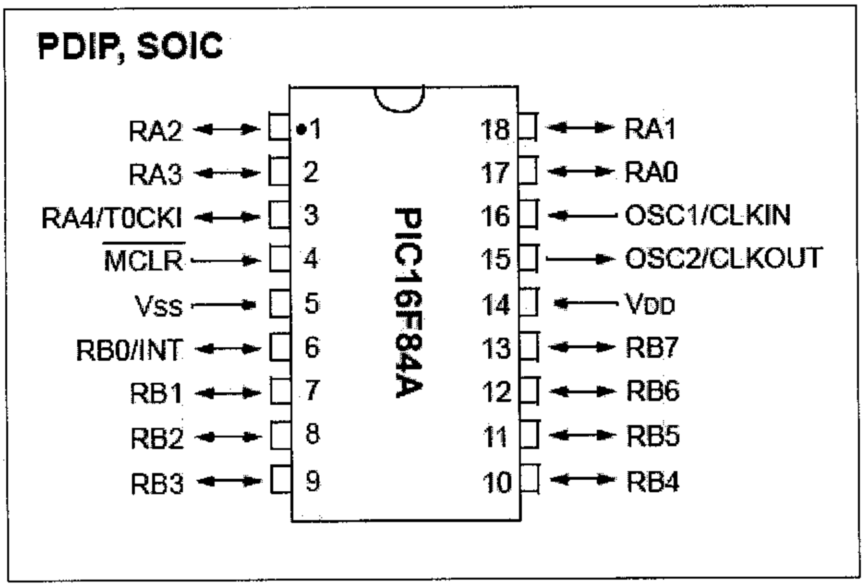


Figure 3.6: Pinout of PIC16F84A

The PIC16F84A belongs to the mid-range family of the PICmicro® microcontroller devices. A block diagram of the device is shown in Figure 3.7. There are also 13 I/O pins that are user-configured on a pin-to-pin basis. Some pins are multiplexed with other device functions. The microcontroller is a programmable device where user need to program the microcontroller prior before use. There are several programmers available on the market to program Microchip's line of products.

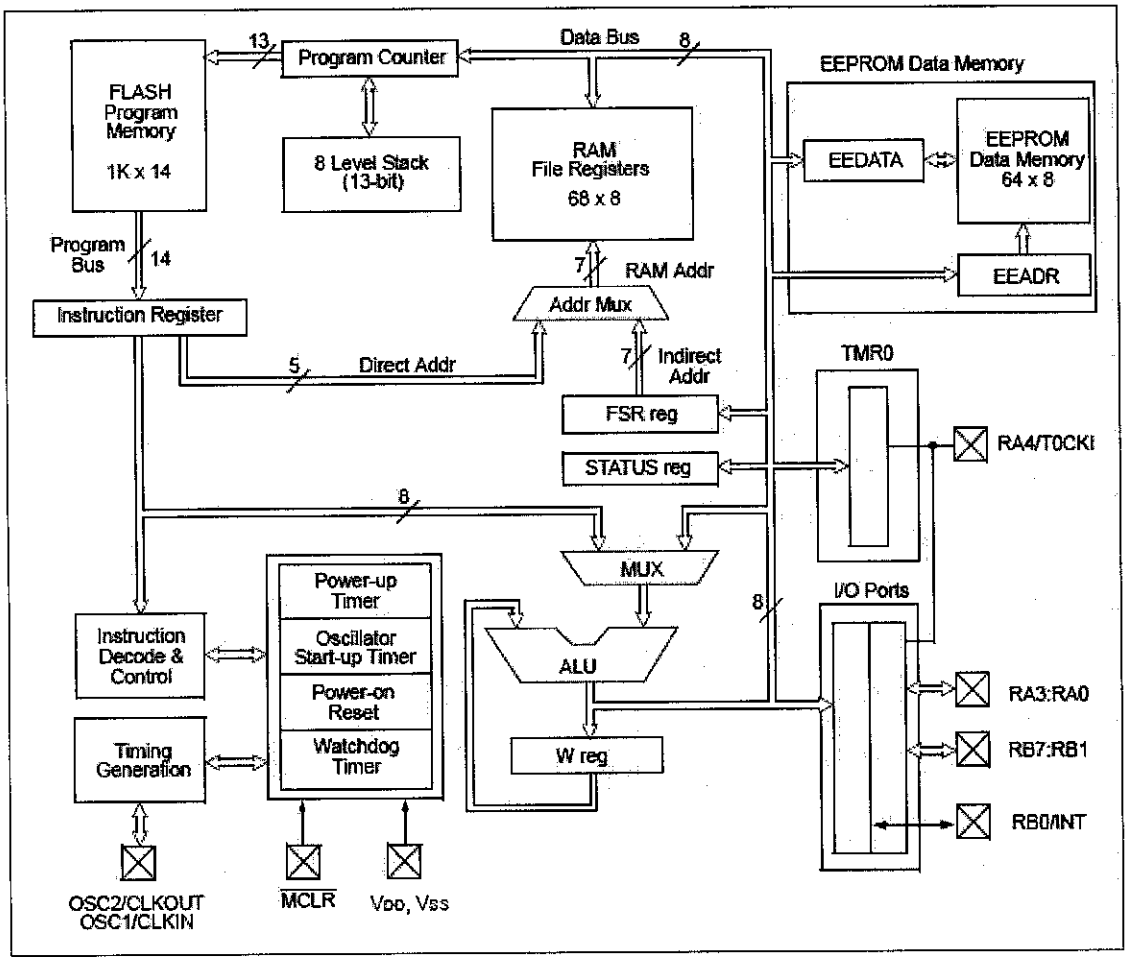
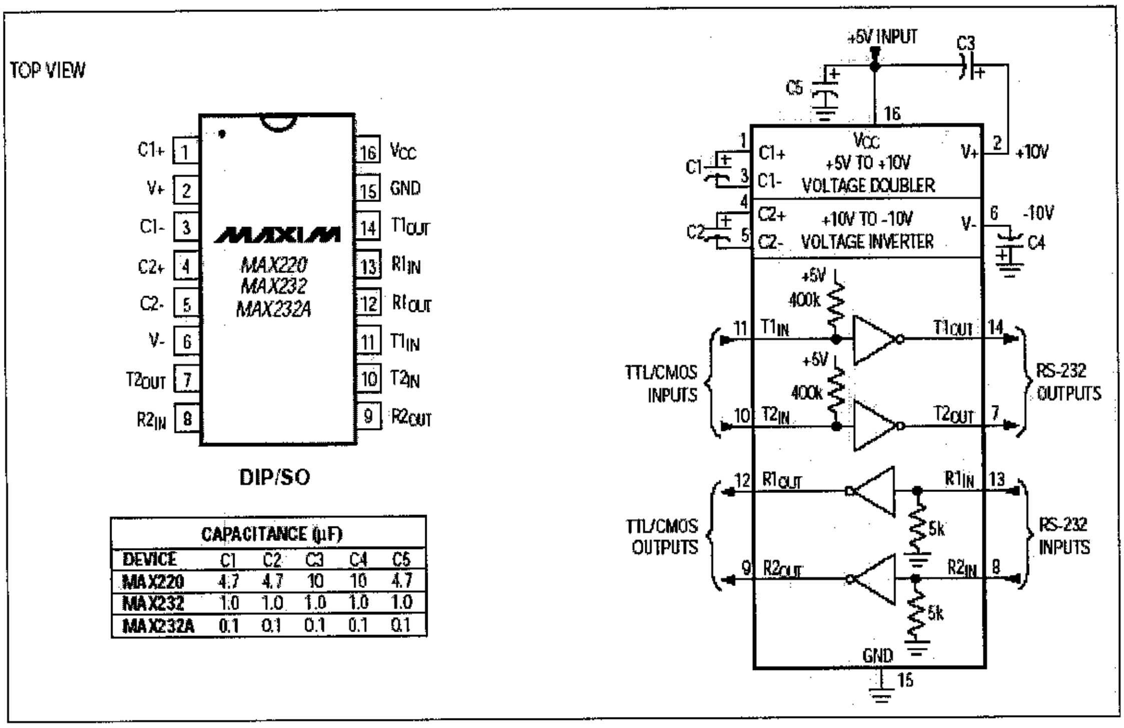


Figure 3.7: PIC16F84A block diagram

The block diagram is explained in detail in the datasheet of the device. For the PIC, the program was obtained earlier and edited so that it conforms to the specification for the application. The pinout description of PIC16F84A can be found in **Appendix A.** Function of the microcontroller in this project is to convert the data from the A/D converter into a format that can be understood by the computer. Data flows in the computer had the specification of bit rate and baud rate. Thus, the microcontroller is will be programmed to do exactly this using the code as in **Appendix C.**

3.3.3 MAX 232

Another essential component for the project is the MAX 232 which is the voltage regulator for the communication port on the computer. It acts as the interface between the PIC and the computer. This is required because the voltage swing provided by the outputs of the PIC does not comply with the voltage required by the COM port in the computer. The MAX 232 has a very high tolerance to variance to its inputs. That is ±30V, and the output of the MAX 232 is at minimum of ±5V. ±5V is the voltage range understood by the COM port while the PIC outputs is only 0 and 5V, that's why MAX 232 is needed. The pin configuration is shown in Figure 3.8 with its typical operating circuit.



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 3.8: MAX 232 pin configuration and its typical operating circuit.

The MAX 232 electrical characteristics are included in **Appendix B**. From the figure above we know that the capacitor works as the charge storage since the input is just 5V and the output is at maximum ±10V. Interface from the MAX 232 only requires two data lines that is transmit and receive. These two data lines are directly buffered from the PIC microcontroller. Interfacing to the COM port requires two wires. They are the receive and ground wires. Ground line is required so that the voltage level has a reference point to where to compare the voltage to. Like measuring voltages, we cannot just check voltage with one probe; we need two probes so that one probe will be the reference to where the other probe will be checked to.

3.3.4 RS 232(Serial port)

The serial port is an I/O device. An I/O device is just a way to get data into and out of a computer. There are many types of I/O devices such as serial ports, parallel ports, disk drive controllers, Ethernet boards, universal serial buses and many others. Most PC's have one or two serial ports. Each has a 9-pin connector or sometimes 25-pin on the back of the computer. Computer programs can send data (bytes) to the transmit pin (output) and receive bytes from the receive pin (input). The other pins are for control purposes and ground.

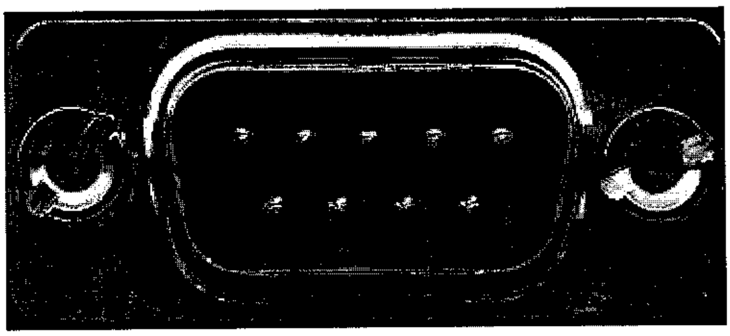


Figure 3.9: Serial Port (9 Pin)

The serial port is much more than just a connector. It converts the data from parallel to serial and changes the electrical representation of the data. Inside the computer, data bits flow in parallel, using many wires at the same time. Serial flow is a stream of bits over a single wire, such as on the Transmit or Receive pin of the serial connector. For the serial port to create such a flow, it must convert data from parallel inside the computer to serial on the transmit pin and conversely.

Most of the electronics of the serial port is found in a computer chip or a part of a chip known as a UART. UART stands for Universal Asynchronous Receiver / Transmitter. It is the little box of tricks found on the serial card which plays the little games with the modem or other connected devices. Most cards will have the UARTs integrated into other chips which may also control the parallel port, games port, floppy or hard disk drives and are typically surface mount devices. The 8250 series, which includes the 16450, 16550, 16650, & 16750 UARTs are the most commonly found type in your PC.

The serial port is harder to interface than the parallel port. In most cases, any device connected to the serial port will need the serial transmission converted back to parallel so that it can be used. This can be done using a UART. On the software side of things, there are many more registers that have to be attended to than on a standard parallel port. The advantages of using serial data transfer rather than parallel are serial cables can be longer than parallel cables. The serial port transmits a '1' as -3 to -25 volts and a '0' as +3 to +25 volts while parallel port transmits a '0' as 0v and a '1' as 5v. Therefore the serial port can have a maximum swing of 50V compared to the parallel port which has a maximum swing of 5 Volts. Therefore cable loss is not going to be as much of a problem for serial cables as they are for parallel.

Besides, less wires are use than parallel transmission. If a device needs to be mounted a far distance away from the computer then 3 core cable (Null Modem Configuration) is going to be a lot cheaper that running 19 or 25 core cable. However the cost of the interfacing at each end must be taken into consideration.

3.4 Microsoft Visual Basic

Microsoft Visual Basic is Windows GUI programming software. Therefore, it is the appropriate software option to extract the input from the serial port and alert the user of any activity via graphical interface.

A component known as the Microsoft Comm Control 6.0 is applied to allow the connection between Visual Basic and the serial port. Among the properties of MSComm that need to be considered are listed in Table below.

Table 1: Microsoft Comm Control 6.0 Properties

Properties	Description
CommPort	Sets and returns the communications port number
Settings	Sets and returns the baud rate, parity, data bits and stop bits as a string
PortOpen	Sets and returns the state of a communication port. Also opens and closes a port
Input	Returns and remove character from the receive buffer
Output	Writes a string of characters to the transmit buffer
InputLen	Sets the maximum number of characters that will be returned when the input property is accessed

CHAPTER 4: RESULT AND DISCUSSION

4.1 Microsoft Visual Basic

After getting familiarize with some of the software to program the Digital and Analog PC Oscilloscope, the Visual Basic has been chosen to program the Digital and Analog PC Oscilloscope. The decisions to choose Visual Basic software are due to the following reasons:

i. *Easy for implementation with real oscilloscope*

   There has been some research and practical work related to this project. This project has been proved easy to implemented especially using Microsoft Visual Basic rather than C++ or MATLAB.

ii. *Easy understanding to code the program*

   In order to write a proper Visual Basic project, there several important elements to learn and understand. The two vital steps are:

   1. *Planning*

   - Design the user interface
   - Plan the properties
   - Plan the Basic code - procedures are associated with the events, actions written in first

   2. *Programming*

   - Define the user interface - define objects
   - Set the properties
   - Write the Basic code

iii. *More graphical interface to be created*

The Microsoft Visual Basic is known to be among the most popular choice to create Windows GUI. In Visual Basic, new windows created are called forms. Elements (such as text boxes and buttons) that are placed inside a form are called controls. The Visual Basic allows event-driven programming, where the user's actions cause events, and each event in turn triggers a procedure that is associated with it.

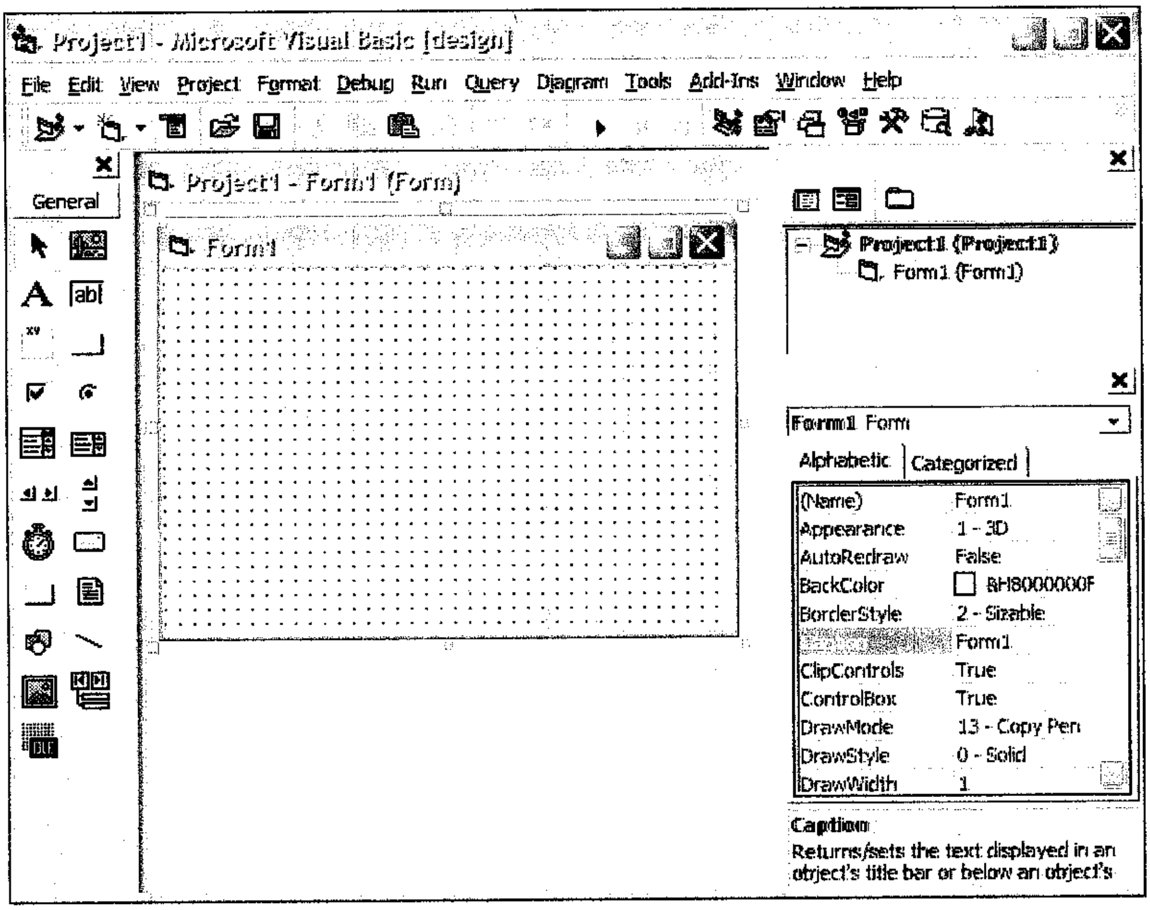


Figure 4.1: Visual Basic Editor

4.2 Microsoft Comm Control 6.0

In order for Visual Basic to establish connection with the serial port, a component called the Microsoft Comm Control 6.0 is applied. This component is represented by an icon of a phone.

Figure 4.2: Microsoft Comm Control 6.0 Icon

The portion of the source codes that initializes the serial port using the Microsoft Comm Control 6.0 is presented below:

```
Private Sub Form_Load()
  'Initialize
  Comm.InputMode = 0 'take ASCII codes as inputs
  Comm.CommPort = 1
  Comm.Settings = "9600,N,8,1"
  Comm.PortOpen = True 'open port
  Comm.InputLen = 1 'limitation for input
  Comm.RThreshold = 1

End Sub
```

The comm settings will coincide with the serial port settings which are:

- Baud rate: 9600
- Parity bits: None
- Data bits: 8
- Stop bit: 1

4.3 Serial Port

Naturally, the output from the external circuit needs to be transmitted to the PC so that the PC Oscilloscope software can detect the input signals. Among the transmission medium available include parallel and serial communication. As explained in earlier, the serial port is chosen as it proves to be more feasible.

As far as interfacing between the external circuit are concerned, initialization is the first consideration by modifying the settings on both ends to accommodate each other. These settings include:

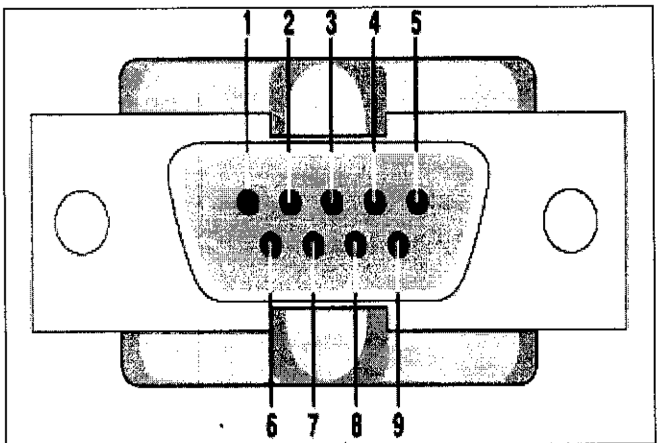
- Baud rate: 9600
- Parity bit: None
- Data bits: 8
- Stop bits: 1

The pin from external circuit that interacts with the serial port is:

- Receive pin (Pin 2)

Interface between the computer and the PIC circuit is the COM 1 port. COM 1 port uses RS 232 connector. This connector is also known as the DB-9 connector. It is called DB-9 because it is constructed with nine pins. Old PC's used 25 pin connectors but only about 9 pins were actually used. So today, most connectors only have 9-pin. Each of the pins usually connects to a wire. Besides the wires used for receiving data, another pin (wire) is signal to ground.

The voltage on any wire is measured with respect to this ground. Thus the minimum number of wires to use for 1-way transmission of data is 2. Despite this method, it has also been known to work with no signal ground wire but with degraded performance and sometimes with errors. But in this project, only two are used. They are pin number 2 and 5. Pin number 2 is the receive pin and pin number 5 is the ground pin.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Figure 4.3: RS-232 Pin Out on a DB-9 Pin

Almost all digital devices used nowadays require either TTL or CMOS logic levels. Therefore the first step to connecting a device to the RS-232 port is to transform the RS-232 levels back into 0 and 5 Volts. This is done by RS-232 level converters. Two common RS-232 level converters are the 1488 RS-232 driver and the 1489 RS-232 receiver. Each package contains 4 inverters of the one type, either drivers or receivers. The driver requires two supply rails, +7.5V to +15V and -7.5V to -15V. As expected, this may pose a problem in many instances where only a single supply of +5V is present. However the advantage of these IC's is that, they are cheap.

4.4 Hardware Part

After the A/D converter, PIC and the MAX 232 circuits are constructed and programmed, the circuit will be tested using Hyper Terminal. HyperTerminal is a program that you can use to connect to other computers, Telnet sites, and bulletin board systems (BBSs), online services, and host computers, using either a modem or a null modem cable. The configuration for the Hyper Terminal will only be done after the completion of the circuit.

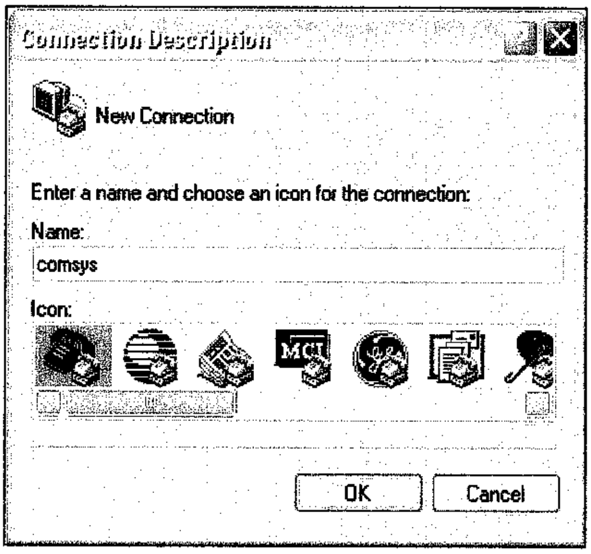


Figure 4.4: Hyper Terminal

If all the transmitter, receiver, PIC and MAX 232 are working correctly, the Hyper Terminal will display the character of A, B, C, D, E, F, G and H. These eight inputs triggered by the PIC circuit. This method can recognize and verify whether the hardware has send data successfully to the Comm port 1 that was used.

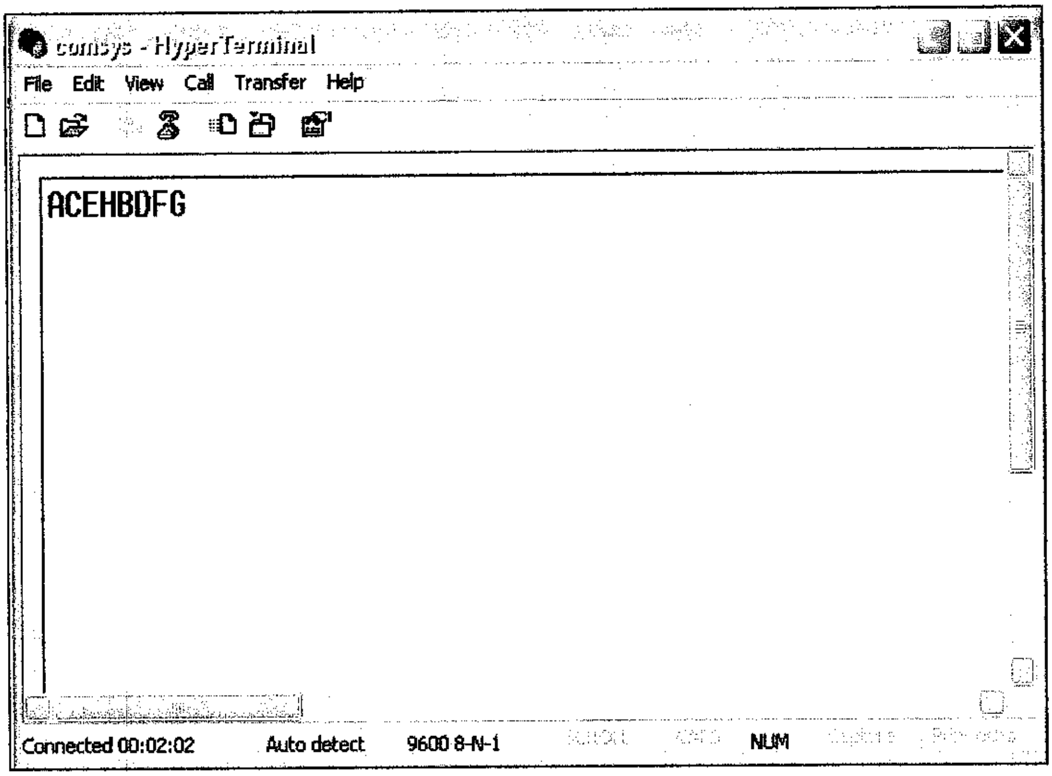


Figure 4.5: Hardware Test Verification

Using this concept, the application to be controlled need to be the active window, like controlling application using the original keyboard. So the applications are controlled like someone is operating the keyboard but actually the Visual Basic is simulating the keyboard.

4.5 PC Oscilloscope Software

The three main windows identified in the program are:

4.5.1 Login window

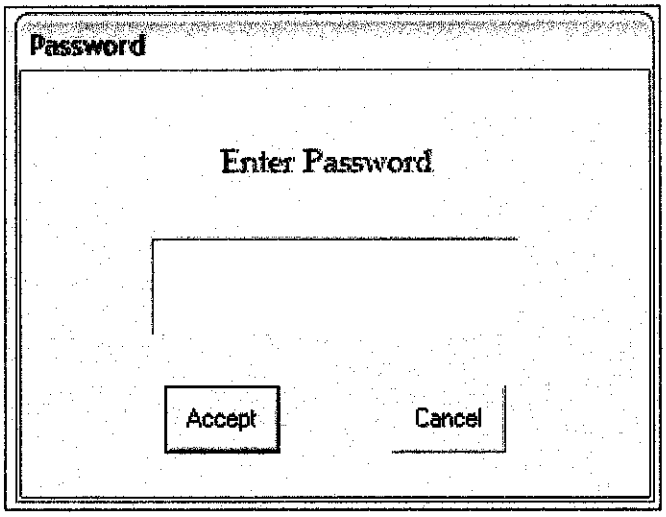


Figure 4.6: Login Window

The user will need to enter the exact password set in the PC Oscilloscope software. The Password can include any characters including numbers. If the password entered was failed, warning will be shown to the user as a caution and the user is requested to try again.

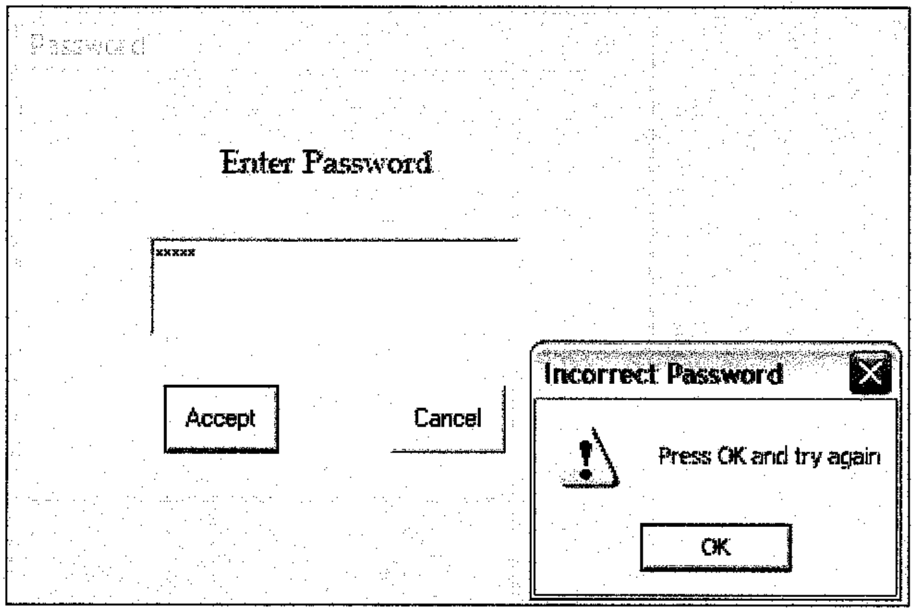


Figure 4.7: Warning for Password Error

However, the user has only three attempts to enter the correct password. If errors still occur, the user is not allowed any more attempts and the program will be terminated immediately. The correct password will link directly to form1, the main window of PC Oscilloscope.

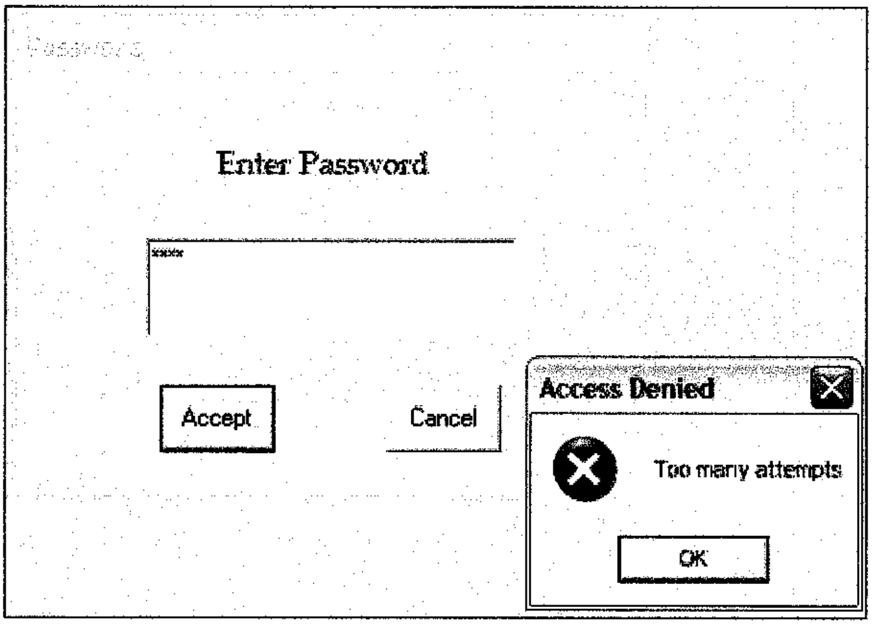


Figure 4.8: Further Attempts Failed

4.5.2 Main window (Welcome)

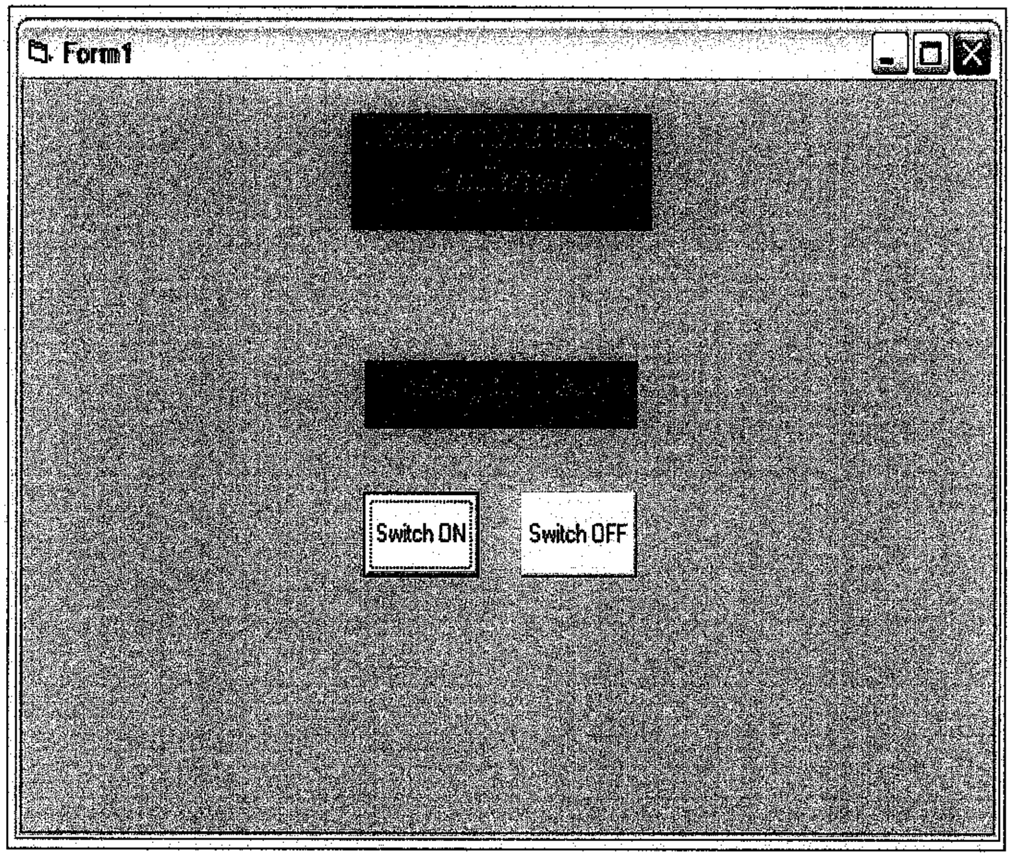


Figure 4.9: Menu Option of PC Oscilloscope

If the user decides to quit they can press the Switch OFF button. PC Oscilloscope window will be appear as soon as Switch ON is pressed.

4.5.3 PC Oscilloscope window

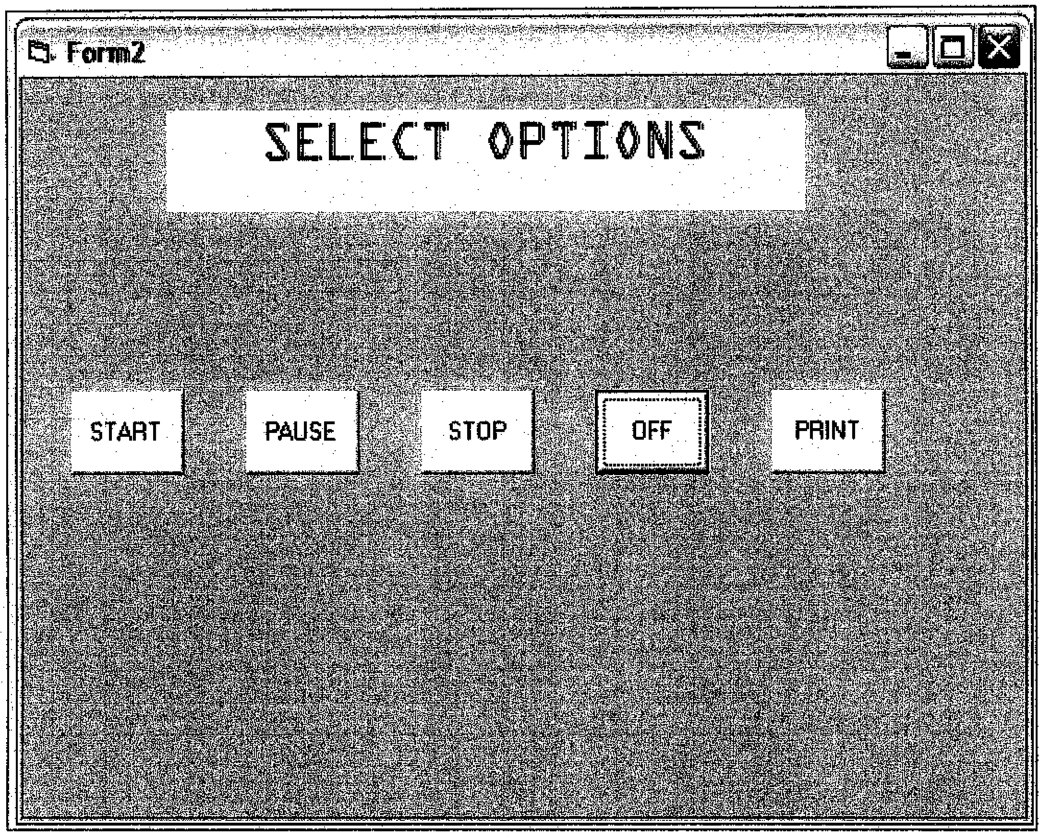


Figure 4.10: PC Oscilloscope Window

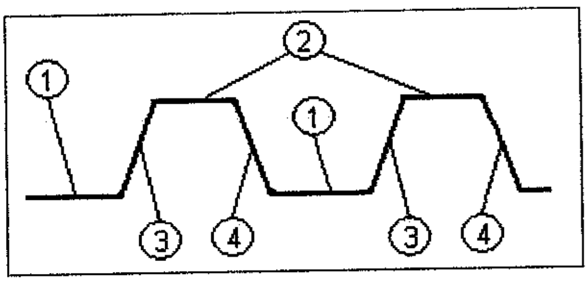
The details of all the windows including the mentioned in the windows are presented in Appendices. The complete source codes are in Appendix D,E and F. The main operations performed by the program include:

- request for password to login
- display error if password is not valid
- alert user of any input from the serial port
- log off from system

4.6 Discussion

Along the FYP way, there were some difficulties encountered in the middle of the process, there were the difficulties of coding program. As programming is the hardest part of the project, everything seemed to be lost if the process was stuck in the middle of the project. To overcome this problem, alternative has been made by working with the hardware part instead finishing the whole programming part.

Theoretically the digital waveform signal should be displayed like picture below. The picture shows switches between two voltage levels representing the two states of an output (HIGH and LOW). Due to the lack of programming skills the waveform could not be achieved as the programming works meet dead ends.



1) Low level
2) High level
3) Rising edge
4) Falling edge.

Figure 4.11: Digital signal waveform

The same thing goes to the analog waveform where the solution to programming display cannot be reached. This is the hardest part in this project where conversion of signal takes place in most of the transition from the input signal to the PC oscilloscope display.

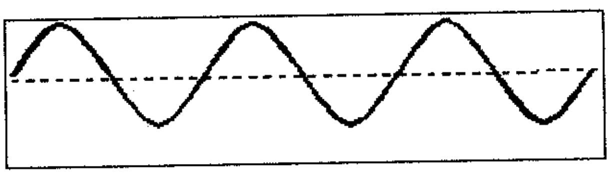


Figure 4.12: Analog signal waveform

However, although the result of this project did not meet the main objective, but lots have been learned along the process of this Final Year Project. New skills have been learned and can expand one knowledge rather than doing it without any positive outcome. There is nothing in this world that is impossible. Maybe, it is just a matter of time to get the result.

Hopefully it can be continued in the future as it brings lot of benefits in terms of positive way especially to the Electronic Students. Moreover there are also several free sample of this PC Oscilloscope software in the internet. These prove that it can be achieved through high extra effort and good programming skills.

CHAPTER 5: CONCLUSION

The Final Year Project's concern is to accomplish the previous mentioned objectives. In performing the first objectives, fact-finding is carried out to understand the function of oscilloscope. Literature reviews have been refined in order to get more understanding regarding PC Oscilloscope. In relating to the above, first objective has been accomplished. Then, the programming work has been started as scheduled in the Gantt Chart.

The project was continued with the hardware part. The conversion of analog to digital signal has been achieved in order to get a successful input source. After two semesters of hard work, the project only manages to display the Digital Signal only. Due to limitation of programming knowledge, the display of both Analog and Digital Signal is still unachievable. But still, the effort to complete the project is still carried away to get the best result.

Based on the project progress, there are some recommendations that have been identified as a future enhancement for this project such as adding more Graphical User Interface (GUI) to the software. This will help other user to understand the software easier if the design is simple and not too complexity. Beside that, suggestion to use PIC16F877 instead of PIC16F84A because it has a built in A/D in it. This can save space on the hardware part. On the other hand, adding channels to the oscilloscope for displaying more waveform was also an option. This can display up to two and even three waveforms simultaneously at the software. Finally, to provide warning sound if error occur in the test circuit and also intrusion by an authorized user to the software.

References

[1] October 2004 < http://www.sightings-in1.com/oscilloscopes/history-of-oscilloscope.htm>

[2] Feb 2006 <http://en.wikipedia.org/wiki/Oscilloscope>

[3] Boehm, Mark. 8 Feb 2006 <http://ezinearticles.com/?PC-Oscilloscopes,-a-Technological-Breakthrough&id=141964>

[4] Patrick G. McKeown, Learning to Program with Visual Basic, John Wiley & Sons,Inc., USA, 1998

[5] Sams Teach Yourself Visual Basic 6 in 21 days, Sams Publishing,USA, 1998

[6] How to Check RS-232 Port to Verify Operation

http://www.bb-europe.com

[7] MAX232 and Serial Communications

http://www.microchipc.com/PIC16bootload

[8] Real Time Surveillance AND Tracking System, Final Year Project November 2003, Raja Danir Ikram Bin Raja Shahrome, Universiti Teknologi Petronas.

APPENDIXES

Appendix A

PIC16F84A pinout description

n Name	PDIP No.	SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
1/CLKIN	16	16	18	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
2/CLKOUT	15	15	19	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
R	4	4	4	I/P	ST	Master Clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
	17	17	19	I/O	TTL	
	18	18	20	I/O	TTL	
	1	1	1	I/O	TTL	
	2	2	2	I/O	TTL	
TOCKI	3	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
INT	6	6	7	I/O	TTL/ST(1)	RB0/INT can also be selected as an external interrupt pin.
	7	7	8	I/O	TTL	
	8	8	9	I/O	TTL	
	9	9	10	I/O	TTL	
	10	10	11	I/O	TTL	Interrupt-on-change pin.
	11	11	12	I/O	TTL	Interrupt-on-change pin.
	12	12	13	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming clock.
	13	13	14	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming data.
	5	5	5,6	P	—	Ground reference for logic and I/O pins.
	14	14	15,16	P	—	Positive supply for logic and I/O pins.

nd: I= input O = Output I/O = Input/Output P = Power
— = Not used TTL = TTL input ST = Schmitt Trigger input

1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

Appendix B

MAX 232 electrical characteristics

= +5V ±10%, C1–C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
232 TRANSMITTERS						
ɔut Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
t Logic Threshold Low				1.4	0.8	V
t Logic Threshold High			2	1.4		V
c Pull-Up/Input Current	Normal operation			5	40	µA
	$\overline{SHDN}$ = 0V, MAX222/242, shutdown			±0.01	±1	
ɔut Leakage Current	V_{CC} = 5.5V, $\overline{SHDN}$ = 0V, V_{OUT} = ±15V, MAX222/242			±0.01	±10	µA
	V_{CC} = $\overline{SHDN}$ = 0V, V_{OUT} = ±15V			±0.01	±10	
ı Rate	All except MAX220, normal operation			200	116	kbits/ sec
	MAX220			22	20	
smitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300	10M		Ω
ɔut Short-Circuit Current	V_{OUT} = 0V		±7	±22		mA
232 RECEIVERS						
:32 Input Voltage Operating Range					±30	V
:32 Input Threshold Low	V_{CC} = 5V	All except MAX243 $R2_{IN}$	0.8	1.3		V
		MAX243 $R2_{IN}$ (Note 2)	-3			
:32 Input Threshold High	V_{CC} = 5V	All except MAX243 $R2_{IN}$		1.8	2.4	V
		MAX243 $R2_{IN}$ (Note 2)		-0.5	-0.1	
:32 Input Hysteresis	All except MAX243, V_{CC} = 5V, no hysteresis in shdn.		0.2	0.5	1	V
	MAX243			1		
:32 Input Resistance			3	5	7	kΩ
CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
CMOS Output Short-Circuit Current	Sourcing V_{OUT} = GND		-2	-10		mA
	Shrinking V_{OUT} = V_{CC}		10	30		
CMOS Output Leakage Current	$\overline{SHDN}$ = V_{CC} or $\overline{EN}$ = V_{CC} ($\overline{SHDN}$ = 0V for MAX222), 0V ≤ V_{OUT} ≤ V_{CC}			±0.05	±10	µA

Appendix C

```
#include <16f84a.h>
#fuses XT,NOPROTECT,NOWDT

#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_A0, rcv=PIN_A1)

#define ALL_OUT 0
#define ALL_IN 0xFF


void respond() {

  output_low(PIN_A2);
  output_low(PIN_A3);
  output_low(PIN_A4);

}

main () {

  int i;

  set_tris_B(0xFF);

  do {

  i=0;
  if (input(PIN_B0)!=0) {

    output_high(PIN_A2);
    output_high(PIN_A4);
    delay_ms(1);
    putch(65);
    delay_ms(1);
    output_high(PIN_A3);


  }

  if (input(PIN_B1)!=0) {

    output_high(PIN_A2);
    output_high(PIN_A4);
```

```
    delay_ms(1000);
    putch(66);
    delay_ms(1);
    output_high(PIN_A3);


  }

  if (input(PIN_B2)!=0) {

    output_high(PIN_A2);
    output_high(PIN_A4);
    delay_ms(1);
    putch(67);
    delay_ms(1);
    output_high(PIN_A3);


  }

  if (input(PIN_B3)!=0) {

    output_high(PIN_A2);
    output_high(PIN_A4);
    delay_ms(1);
    putch(68);
    delay_ms(1);
    output_high(PIN_A3);


  }

  if (input(PIN_B4)!=0) {

output_high(PIN_A2);
    output_high(PIN_A4);
    delay_ms(1);
    putch(69);
    delay_ms(1);
    output_high(PIN_A3);


  }

  if (input(PIN_B5)!=0) {
```

```
   output_high(PIN_A2);
   output_high(PIN_A4);
   delay_ms(1);
   putch(70);
   delay_ms(1);
   output_high(PIN_A3);


  }

  if (input(PIN_B6)!=0) {

output_high(PIN_A2);
   output_high(PIN_A4);
   delay_ms(1);
   putch(71);
   delay_ms(1);
   output_high(PIN_A3);


  }

  if (input(PIN_B7)!=0) {

   output_high(PIN_A2);
   output_high(PIN_A4);
   delay_ms(1);
   putch(72);
   delay_ms(1);
   output_high(PIN_A3);


  }

  } while (TRUE);

  }
```

Appendix D

Login Source Code

```
Private Sub cmdAccept_Click()
   Static NumTries
   If UCase(txtPassword.Text) = UCase(txtPassword.Tag) Then
      Form1.Show 'Go to main form
      Unload Me
   Else 'No match found
      NumTries = NumTries + 1 'Increment number of attempts
      If NumTries >= 3 Then 'Too many attempts
         MsgBox "Too many attempts", vbCritical, "Access Denied"
         End
      Else 'Try again
         MsgBox "Press OK and try again", vbExclamation, _
         "Incorrect Password"
         txtPassword.Text = ""
         txtPassword.SetFocus
      End If
   End If
End Sub


Private Sub cmdCancel_Click()
   Unload Me
End Sub
```

Main Window (Welcome)

```
Private Sub Command1_Click()
  Form2.Show
  Unload Me
End Sub

Private Sub Command2_Click()
   Unload Me
End Sub
```

Appendix E

MSComm Source Code

```
'declare integer
Public num As Integer
Public prev_num As Integer
Public total As Integer
Public inc As Integer
Public stopnow As Integer
Public realtotal As Integer
Public t As Integer

Private Sub Comm_OnComm()

'reading

    Select Case comm.Input
      Case "A"
        Data1A.BackColor = &H80000002
        inc = 1
        prev_num = num
        num = 1
        Call process
        If stopnow > 5 Then
          Data1.Caption = realtotal

        End If

      Case "B"
        Data1B.BackColor = &H80000002
        inc = 2
        prev_num = num
        num = 2
        Call process
        If stopnow > 5 Then
          Data1.Caption = realtotal

        End If

      Case "C"
        Data1C.BackColor = &H80000002
        inc = 4
        prev_num = num
        num = 3
        Call process
```

```
    If stopnow > 5 Then
        Data1.Caption = realtotal

    End If

Case "D"
    Data1D.BackColor = &H80000002
    inc = 8
    prev_num = num
    num = 4
    Call process
    If stopnow > 5 Then
        Data1.Caption = realtotal

    End If

Case "E"
    Data1E.BackColor = &H80000002
    inc = 16
    prev_num = num
    num = 5
    Call process
    If stopnow > 5 Then
        Data1.Caption = realtotal

    End If

Case "F"
    Data1F.BackColor = &H80000002
    inc = 32
    prev_num = num
    num = 6
    Call process
    If stopnow > 5 Then
        Data1.Caption = realtotal

    End If

Case "G"
    Data1G.BackColor = &H80000002
    inc = 64
    prev_num = num
    num = 7
    Call process
    If stopnow > 5 Then
        Data1.Caption = realtotal
```

```
        End If

      Case "H"
        Data1H.BackColor = &H80000002
        inc = 128
        prev_num = num
        num = 8
        Call process
        If stopnow > 5 Then
          Data1.Caption = realtotal


        End If


    End Select


  t = t + 1
  If t > 500 Then
    Data1A.BackColor = &H8000000F
    Data1B.BackColor = &H8000000F
    Data1C.BackColor = &H8000000F
    Data1D.BackColor = &H8000000F
    Data1E.BackColor = &H8000000F
    Data1F.BackColor = &H8000000F
    Data1G.BackColor = &H8000000F
    Data1H.BackColor = &H8000000F


    t = 0
  End If

End Sub
```

Appendix F

Graph Display Code

```
'graph
Public Function process()
   If num <= prev_num Then
      realtotal = total
      total = inc
      stopnow = 10
      Call draw_graph
   End If
   If num > prev_num Then
      total = total + inc
      stopnow = 0
   End If
End Function

Public Function draw_graph()
   Chart.Column = 1
   Chart.Row = 1
   temp_data1 = Chart.Data
   temp_data2 = Chart.Data

   Chart.Data = realtotal

   For i = 2 To 1000
   Chart.Row = i
   temp_data2 = temp_data1
   temp_data1 = Chart.Data
   Chart.Data = temp_data2

   Next
End Function
```

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of Project Topic														
	-Propose Topic														
	-Topic assigned to students														
2	Preliminary Research Work														
	-Introduction														
	-Objective														
	-List of references/literature														
	-Project planning														
3	Submission of Preliminary Report														
4	Project Work														
	-Reference/Literature														
	-Software Part														
5	Submission of Progress Report														
6	Project work continue														
	-Software Part														
7	Submission of Interim Report Final Draft														
8	Oral Presentation														
9	Submission of Interim Report														

Appendix H: Suggested Milestone for the Second Semester of 2 Semester Final Year Project

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Project Work Continue														
	-Hardware Part														
2	Submission of Progress Report 1														
3	Project Work Continue														
	-Hardware Part														
4	Submission of Progress Report 2														
5	Project work continue														
	-Compilation of Software & Hardware														
6	Submission of Dissertation Final Draft														
7	Oral Presentation														
8	Submission of Project Dissertation														

Appendix I: Hardware Circuit