

Computer Controlled LEDs Display Board

By

Tan Rhu Yee

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

DECEMBER 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

Computer Controlled LEDs Display Board

By

TAN RHU YEE

6013

FINAL YEAR PROJECT REPORT

Submitted to the Electrical & Electronics Programme
in Partial Fulfilment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

© Copyright 2006

by

Tan Rhu Yee

CERTIFICATION OF APPROVAL

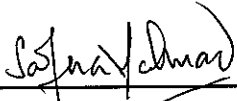
Computer Controlled LEDs Display Board

by

Tan Rhu Yee

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 by,




(Ms. Salina bt. Mohmad)
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
December 2006

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.



TAN RHU YEE

ABSTRACT

LEDs Display Board is a board for dynamically displaying any information in text which is provided. The purpose of this project is to allow the user to enter a message from the PC keyboard and have this message displayed on a 25 x 7 LED display. The LED display will display the last message that has been saved to the on board memory of the micro processor. After the message has started to display, the user has the option of saving a new message to the memory area of the LED display. The message will be installed via the PC keyboard using RS232 serial communication from a terminal program and saved to the EEPROM memory of the Peripheral Interface Controller (PIC). The program will then read the message saved to the memory and put this out to the LED display. The main objective is to develop a LEDs display board on a Printed Circuit Board (PCB), which can be controlled via a computer. The scope will cover more on schematic drawing, circuits building, wiring and software designing.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude and appreciation to several parties who have given me guidance, support and invaluable advice. Most importantly, I would like to express my deepest gratitude to my family, especially my mother and my father. Their love, care and support have always been an encouragement for me to move forward.

Sincere gratitude also goes to my supervisor Ms. Salina bt Mohmad for allocate her time to supervise and guide me throughout the project. It has been a pleasure working under the direction of my supervisor. Thanks for her patience, inspiration, contribution of precious ideas and constant guidance.

Lastly, I would like to thank the lecturers and staffs of the Electrical & Electronics Engineering Department of Universiti Teknologi PETRONAS for their support and assistance in completing my final year project. Despite of their many responsibilities, they have always cheerfully to accommodate our requests, especially Ms. Siti Hawa and Mr. Musa Hassan who had giving untiring commitment to the project. Thanks to those who had directly or indirectly help me in this project. Their willingness to cooperate and assist had helped me to complete the project.

TABLE OF CONTENTS

CERTIFICATION	iii
ABSTRACT	v
ACKNOWLEDGEMENT	vi
CHAPTER 1:	INTRODUCTION	1
	1.1	Background of Study	1
	1.2	Problem Statement	1
	1.3	Objectives and Scope of Study	2
CHAPTER 2:	LITERATURE REVIEW AND THEORY	3
	2.1	Terminology	3
	2.1.1	LEDs Display Board	3
	2.1.2	LEDs Dot Mtrix	3
	2.1.3	Serial Port	5
	2.1.4	USB Port	7
	2.1.5	Printed Circuit Board	8
	2.1.6	Visual Basic6 Soft Ware	8
	2.1.7	PIC16F84 Microcontrollers	8
	2.1.8	Shift Register	10
	2.1.9	ASCII Code	12
	2.1.10	EAGLE Software	13
CHAPTER 3:	METHODOLOGY/PROJECT WORK	14
	3.1	Equipments and Software	14
	3.2	Project Flow	14
	3.3	Research	17
	3.4	Graphical Interface	17
	3.4.1	Program Planning	18
	3.5	Hardware Implementation	20
	3.5.1	Designing and Planning a PCB	20

CHAPTER 4:	RESULT AND DISCUSSION	25
4.1	Result	25
4.1.1	Visual Basic Interface.	25
4.1.2	LEDs Display Board	28
4.2	Discussion	34
CHAPTER 5:	CONCLUSION AND RECOMMENDATION	14
5.1	Conclusion	38
5.2	Recommendation	39
REFERENCES		40
APPENDICES		42
APPENDIX A	Gantt chart of final year project activities	42
APPENDIX B	Main Screen Interface Code.	44
APPENDIX C	Login Screen Interface Code.	45
APPENDIX D	Message Interface Code	47
APPENDIX E	Schematic Diagram	48
APPENDIX F	ASCII Code	49

LIST OF FIGURES

Figure 2.1	Model 1512
Figure 2.2	2in 50 8mm Dot Matrix LED Displays
Figure 2.3	Dot Matrix LED Displays Schematic
Figure 2.4	LED Dot Matrix Top View
Figure 2.5	Serial Cable (DB9)
Figure 2.6	USB Series “A” plug
Figure 2.7	View from Track Side
Figure 2.8	View from Component Side
Figure 2.9	PIC 16F84
Figure 2.10	Pin-out for PIC16F84
Figure 2.11	D Suffix SOIC Package Case 751A–03
Figure 2.12	Logic Diagram
Figure 2.13	Timing Diagram
Figure 3.1	Project Flow Chart (FYP1)
Figure 3.2	Project Flow Chart (FYP2)
Figure 3.3	Methodology of LEDs Display Board design
Figure 3.4	Pictorial Representation of problem Solving
Figure 3.5	Flowchart for graphical Interface
Figure 3.6	Various Test Probes
Figure 3.7	Test-point Connector
Figure 4.1	Main Screen Interface
Figure 4.2	Microsoft Access file with Confidential Username and Password
Figure 4.3	Login Screen Interface
Figure 4.4	Message Interface
Figure 4.5	The Flow Of Whole Process
Figure 4.6	Component layout
Figure 4.7:	PCB Layout
Figure 4.8	Display Board Controller PCB (Before)
Figure 4.9	Display Board Controller PCB (After-Front View)
Figure 4.10	Display Board Controller PCB (After-Back View)
Figure 4.11	LEDs Display Board PCB (Before)
Figure 4.12	LEDs Display Board PCB (After-Front View)

- Figure 4.13 LEDs Display Board PCB (After-Back View)
- Figure 4.14 LEDs Display Board
- Figure 4.15 Properties Window

LIST OF TABLES

- Table 2.1 Model 1512 Details
- Table 2.2 2in 50 8mm Dot Matrix LED Displays
- Table 2.3 Serial Cable (DB9) Pin Function

CHAPTER 1

INTRODUCTION

1.1 Background of Study

LEDs (Light Emitting Diodes) display board is commonly used nowadays to display announcement or advertisement for public view where the messages have been stored earlier in the memory chips. In daily life, board for dynamically display messages and information can be found hang on the wall of lounge of an airport, a main railroad station hall, or a passenger terminal of a harbour.

This Computer Controlled LEDs Display Board can be manually programmed to display different messages as wished, by controlling it via a computer. The computer and the LEDs board are connected together through serial port. In order to make it a user friendly device, an interactive graphical user interface has been created in the computer, whereby the LEDs display board is constructed on a PCB (Printed Circuit Board).

The message board system begins with a software shell. This software is programmed in Visual Basic6 due to its ease of designing for graphical interfaces. For security reason, the software begins by prompting the user to enter an accepted user name and password. This way, the chance of unauthorized access is greatly limited. Upon entrance of an accepted password, the user can enter a text message to be displayed on the message board [1].

1.2 Problem Statement

Many times there are needs to convey information to a large multitude of people which unreachable personally in a university. Under such circumstances, a message display board has the ability to convey message effectively. The display board will most likely be used by professors or lectures wishing to communicate with his or her students.

Currently, LEDs message display boards available in the market comes with pre-set messages. This set-up limits the variation of the product. The consumers do not have the choice to display additional or personalised information according to their needs and desires. In order to add variations to the product, the LEDs display board can be equipped with a computer which enables the consumers to input the message(s) they considered necessary.

The challenges in this project is to connect the LEDs display board to the computer using RS232 serial communication from a terminal program and saved to the EEPROM (Electrically Erasable Programmable Read-Only Memory) memory of the PIC microcontroller. VB6 (Visual Basic6) software is the recommended software to produce a control program, which will control the whole system because it is a user friendly program.

1.3 Objective and Scope of Study

The objective of this project is to construct a users-friendly LEDs display board, and an interactive graphical user interface in normal PC (Personal Computer) to control the LEDs display board. Based on the input given by the users, the LEDs display board will be able to display the required message(s).

The project will focus on the PCB designing of the LEDs display board and VB6 codes generating to control the whole system. This project started with computer control activities familiarization and gathering deep information on its definitions, characteristics, requirements, applications and advantages.

This covers PCB designing where electric and electronics factor must take into consideration and developing a program in VB interface which able to give input to the display board.

CHAPTER 2

LITERATURE REVIEW AND THEORY

2.1 Terminology

2.1.1 LEDs Display Board

LEDs display board is very exciting & excellent media for indoor & outdoor advertising and are clearly visible from very long distance [2]. It can display both text and graphic messages. The display color can be red, green or amber.

In the market, there are several types of LEDs Display Board, which built in pre-set message is programmed during its manufacture process. Below is one of the example:

i. Model 1512 – Red or Tricolor Readerboard

This single line reader board is for a smaller call center or an office that does not require much information to be displayed at one time [3].



Figure 2.1 : Model 1512

Table 2.1 : Model 1512 Details

# of Lines	Character Height	# of Characters	Case Dimensions	Weight	Color
1	2"	15 per line	29"Lx4.5"Dx2"D	6.25lbs	Red/Tricolor

2.1.2 LEDs Dot Matrix

LED dot matrix can display both text and graphic messages. The display colors are red, green and amber. 5x7 dot matrix has high efficiency red LED displays with X-Y

select, end stackable with interlocking moldings and designed for maximum on/off contrast. Connections are along the top and bottom of the display, simplified the wiring in multi-digit applications [4].

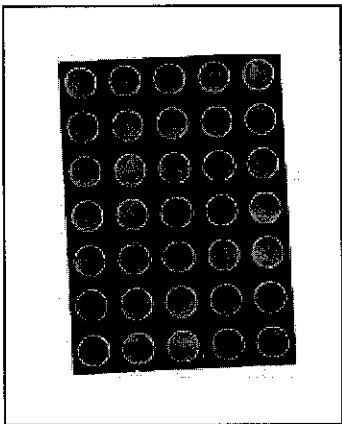


Figure 2.2: 2in 50 8mm Dot Matrix LED Displays

Table 2.2 : 2in 50 8mm Dot Matrix LED Displays

RS Stock No.	589-165
Manufacturer	LEDTECH
Name	ER-LED display, 5x7 dot matrix, common anode, red, 2in, 14 pin

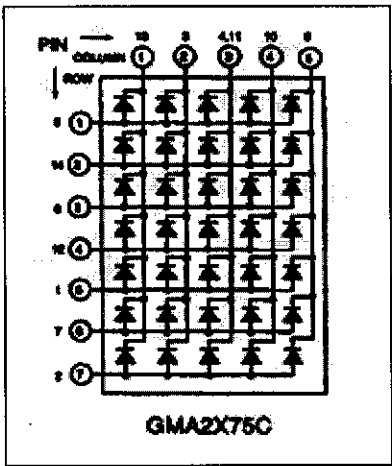


Figure 2.3: Dot Matrix LED Displays Schematic

589-165 Pin No. : Function

Pin 1: Anode Row 5
Pin 2: Anode Row 7
Pin 3: Cathode Column 2
Pin 4: Cathode Column 3
Pin 5: Anode Row 4
Pin 6: Cathode Column 5
Pin 7: Anode Row 6
Pin 8: Anode Row 3
Pin 9: Anode Row 1
Pin 10: Cathode Column 4
Pin 11: Cathode Column 3
Pin 12: Anode Row 4
Pin 13: Cathode Column 1
Pin 14: Anode Row 2

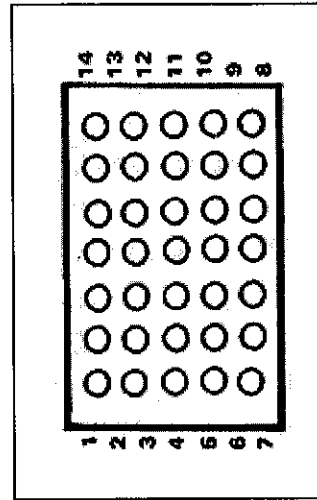


Figure 2.4: LED Dot Matrix Top View

Note: Pin 5 and 12 are Internally Connected.
Pin 5 and 12 are Internally Connected.

2.1.3 Serial Port

Almost every PC has an RS-232C serial interface. RS-232 (Recommended Standard-232) is a telecommunications standard for binary serial communications between devices. It supplies the roadmap for the way devices speak to each other using serial ports. The devices are commonly referred to as a DTE (data terminal equipment) and DCE (data communications equipment); For example, a computer and modem, respectively [5]. The operating system supports up to four interfaces known as COM1: through COM4:

The advantages of using serial data transfer rather than parallel [6]:

- i. 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 where as a 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.

- ii. Does not need as many wires as parallel transmission needs. If the device needs to be mounted a far distance away from the computer, it is recommended to use 3 core cable (Null Modem Configuration) instead of 19 or 25 core cable because it is cheaper. However cost of the interfacing at each end must be take into account too.
- iii. Infra Red devices have proven quite popular recently where many electronic diaries and palmtop computers have this capabilities built in. To transmit 8 bits of data at one time across the room, it is difficult to (from the devices point of view) decipher which bits are which. Therefore, serial transmission is used where one bit is sent at a time.
- iv. Microcontrollers have also proven to be quite popular recently. Many of these have built in SCI (Serial Communications Interfaces) which can be used to talk to the outside world. Serial Communication reduces the pin count of these MPU (Microprocessor chips). Only two pins are commonly used, Transmit Data (TXD) and Receive Data (RXD), compared with at least 8 pins if 8 bit Parallel method is used.

In serial communications, data is transferred from sender to receiver one bit at a time through a single line or circuit. The serial port takes 8, 16 or 32 parallel bits from computer bus and converts it as an 8, 16 or 32 bit serial stream. Each bit of information is transferred in series from one location to another.

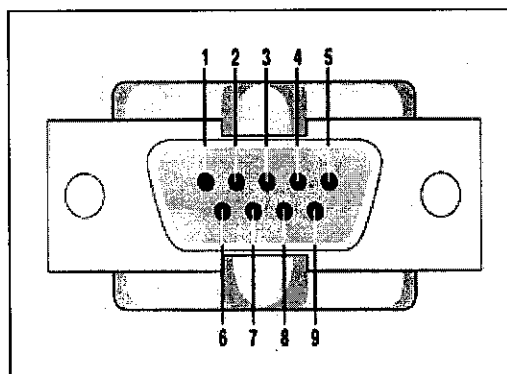


Figure 2.5: Serial Cable (DB9)

Table 2.3 : Serial Cable (DB9) Pin Function

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		

2.1.4 USB Port

USB (Universal Serial Bus), is a hardware bus standard that allows users to plug a peripheral device into a USB port and have it automatically configured and ready to use. Most computers that were manufactured after 1995 have USB ports. Windows 95 had only erratic support for USB, but all newer Windows operating systems have support for USB. It has three great features: speed, power, and ease of use [7].

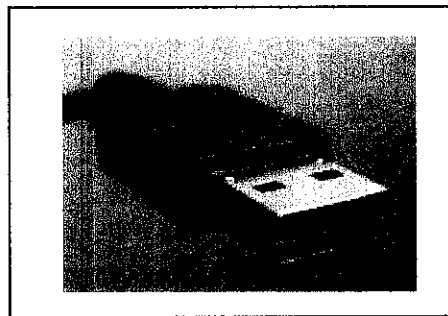


Figure 2.6: USB Series "A" plug

USB is an I/O bus that transfers data at up to 12 megabits per second. The advantages of USB port are it is faster than serial and most parallel ports; can connect up to 127 peripherals at once and widely used. It's flexible, with support for hot plugging and Plug and Play. However it is not suited for high-bandwidth peripherals.

Information transfers over the bus are called transactions [8]. At any time the host controller may have a list of transactions that are waiting to be executed. A transaction begins when the controller sends a packet describing the type and direction of the transaction, the 7-bit USB device address and the endpoint number. This packet is called the Token Packet.

2.1.5 Printed Circuit Board

Printed circuit boards, or PCBs, are used to mechanically support and electrically connect electronic components using conductive pathways, or *traces*, etched from copper sheets laminated onto a non-conductive *substrate* [9]. The boards are made from glass reinforced plastic with copper tracks in the place of wires.

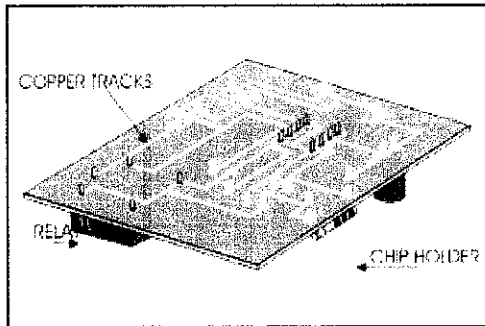


Figure 2.7: View from Track Side

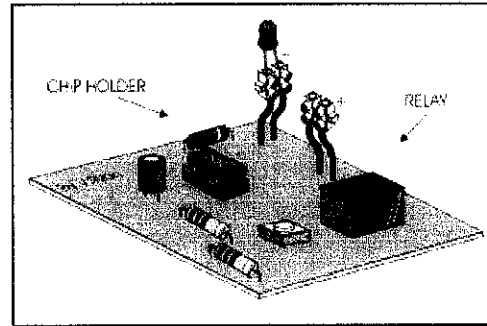


Figure 2.8: View from Component Side

2.1.6 Visual Basic6 Software

Visual Basic was designed to make user-friendly programs easier to develop. Prior to the creation of visual basic, developing a friendly user interface usually requires a programmer to use a language such as C or C++, often requiring hundreds of lines of codes just to get a window to appear on the screen [9]. With visual basic, the same program can be created with much less time and fewer instructions using a language that is a direct descendant of BASIC—the language most accessible to beginning programmers. Visual Basic 6.0 requires the Microsoft Windows operating system.

2.1.7 PIC16F84 Microcontrollers

A PIC is a **micro-controller** (manufactured by Microchip). It is a component which includes RAM, ROM or equivalent, a calculating unit, and inputs/outputs. In fact, there is a whole family of components more or less powerful and with different sizes. They are interesting as being RISC micro-controllers (with few instructions), cheap and thrifty in power [10].

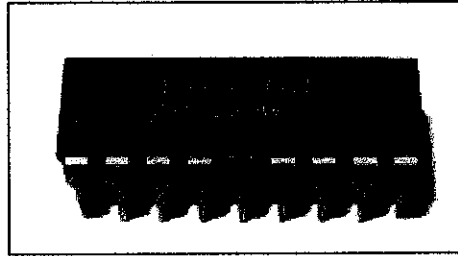


Figure 2.9: PIC 16F84

The microcontroller in this project (PIC16F84) can be programmed and re-programmed up to 1,000 times and this makes it ideal for experimenting [13]. The other advantage of this chip is the ability to program it while it is “in circuit”.

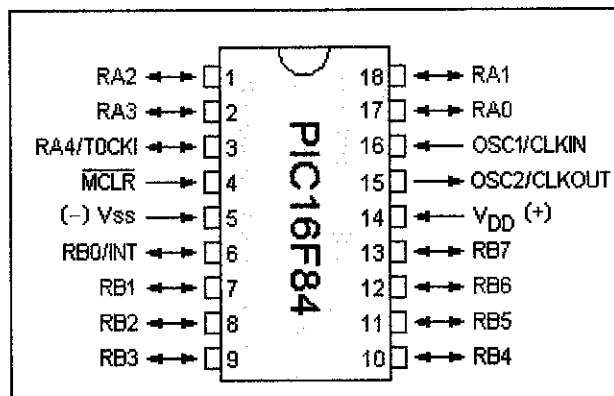


Figure 2.10: Pin-out for PIC16F84

The basics of programming a PICF16F84 [13] are as below:

- i) When /MCLR (pin 4) is taken HIGH (to a voltage called VIH 12v to 14v), the chip turns into programming mode and two pins change from in/out pins to Clock and Data pins.
- ii) Port B bit 6 (pin 12) changes from an in/out pin to **CLOCK**. (to clock data into the chip during programming and clock it out of the chip during "read" mode).
- iii) Port B bit 7 (pin 13) changes from an in/out pin to DATA in/out. The /MCLR pin becomes Vtestmode during programming mode.

- iv) Data books on the PIC16F84 state that the programming voltage (about 13v) is internally generated and the voltage delivered to the Vtestmode pin is purely a reference voltage and no current is required to be delivered to this pin during programming. By using these features, a circuit with the PIC chip in programming mode can be produced.
- v) This is not a functional circuit as the programming voltage (V_{pp}) must be able to be switched from LOW to HIGH to place the chip in programming mode.

2.1.8 Shift Register MC54/74HC164A

Shift registers can have a combination of serial and parallel inputs and outputs, including serial-in, parallel-out (SIPO) and parallel-in, serial-out (PISO) types. There are also types that have both serial and parallel input and types with serial and parallel output. The serial input and outputs of a register can also be connected together to create a circular shift register. One could also create multi-dimensional shift registers, which can perform more complex computation [11].

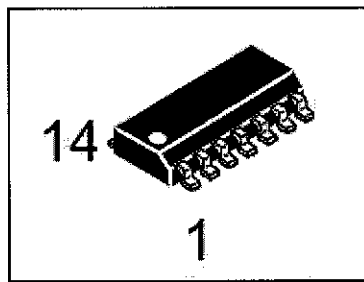


Figure 2.11: D Suffix SOIC Package Case 751A-03

The device inputs are compatible with standard CMOS outputs. With pull up resistors, they are compatible with LSTTL outputs. The C54/74HC164A is an 8-bit, serial-input to parallel-output shift register.

Two serial data inputs, A1 and A2, are provided so that one input may be used as a data enable. Data is entered on each rising edge of the clock. The active-low asynchronous Reset overrides the Clock and Serial Data inputs.

- i) Output Drive Capability: 10 LSTTL Loads
- ii) Outputs Directly Interface to CMOS, NMOS, and TTL
- iii) Operating Voltage Range: 2 to 6 V
- iv) Low Input Current: 1 mA
- v) High Noise Immunity Characteristic of CMOS Devices
- vi) Chip Complexity: 244 FETs or 61 Equivalent Gates

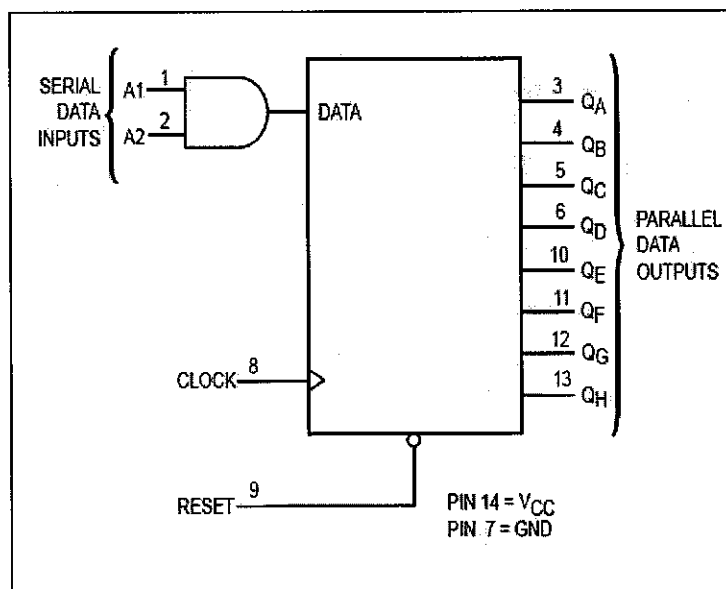


Figure 2.12: Logic Diagram

Pin Descriptions :

INPUT

A1, A2 (Pins 1, 2)

Serial Data Inputs. Data at these inputs determine the data to be entered into the first stage of the shift register. For a high level to be entered into the shift register, both A1 and A2 inputs must be high, thereby allowing one input to be used as a data-enable input. When only one serial input is used, the other must be connected to VCC.

Clock (Pin 8)

Shift Register Clock. A positive-going transition on this pin shifts the data at each stage to the next stage. The shift register is completely static, allowing clock rates down to DC in a continuous or intermittent mode.

OUTPUTS

QA – QH (Pins 3, 4, 5, 6, 10, 11, 12, 13)

Parallel Shift Register Outputs. The shifted data is presented at these outputs in true, or noninverted, form.

CONTROL INPUT

Reset (Pin 9)

Active–Low, Asynchronous Reset Input. A low voltage applied to this input resets all internal flip–flops and sets Outputs QA – QH to the low level state.

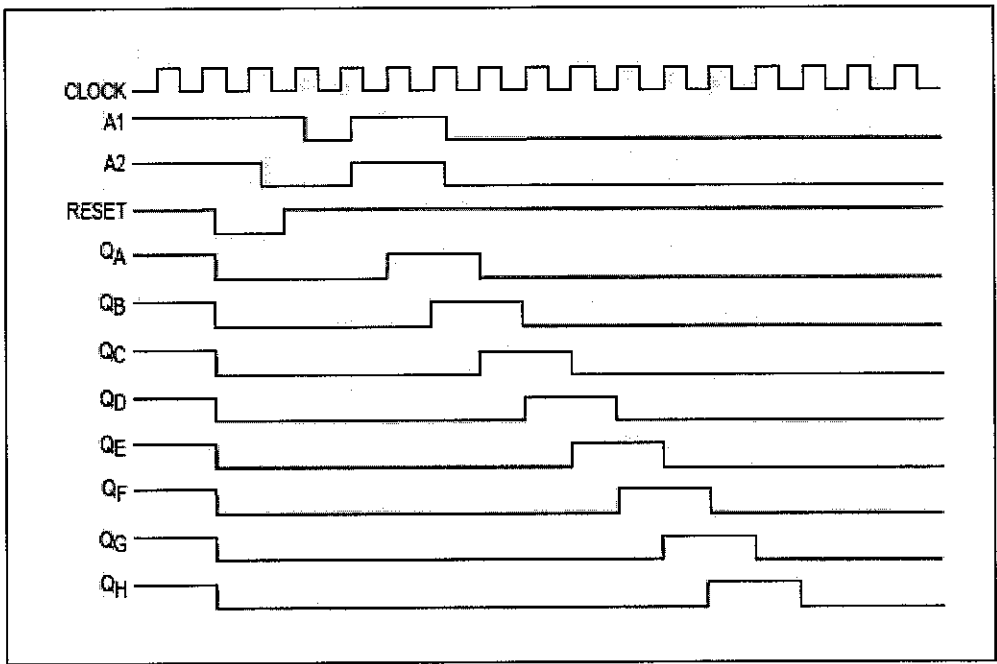


Figure 2.13: Timing Diagram

2.1.9 ASCII Code

Like other character representation computer codes, ASCII specifies a correspondence between digital bit patterns and the symbols / glyphs of a written language. Thus, this allows digital devices to communicate with each other and to process, store, and communicate character-oriented information.

ASCII is, strictly, a seven-bit code. Meaning that it uses the bit patterns representable with seven binary digits (a range of 0 to 127 decimal) to represent character information as shown in **Appendix F**. At the time ASCII was introduced, many computers dealt with eight-bit groups (bytes or, more specifically, octets) as the smallest unit of information; the eighth bit was commonly used as a parity bit for error checking on communication lines or other device-specific functions [12].

2.1.10 EAGLE Software

The EAGLE Layout Editor is an easy to use, yet powerful tool for designing printed circuit boards (PCBs). The name EAGLE stands for Easily Applicable Graphical Layout Editor [14]. Eagle software is one of the examples of CAD (Computer Aid Design). CAD does for graphic documentation what word processing does for written documentation.

CHAPTER 3

METHODOLOGY/PROJECT WORK

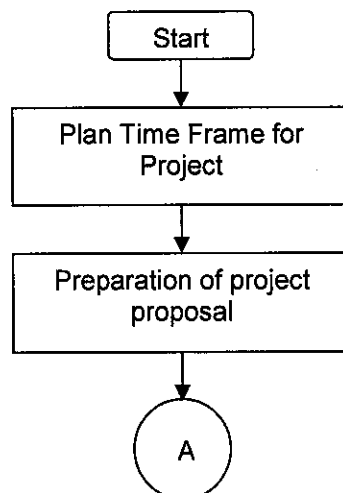
3.1 Equipments and Software

The followings are the equipments that will to be used as part of the Computer Controlled LEDs Display Board:

- i. PIC 16F84 Microchip Microcontroller
- ii. RS232 Transceiver
- iii. Shift Registers
- iv. 5x7 LED Dot Matrix Displays
- v. Serial Cable
- vi. Normal Personal Computer
- vii. Visual Basic 6 software
- viii. Oscilloscope

3.2 Project Flow

This part consist of the process flow for the first half (FYP1) and second half (FYP2) of the final year project.



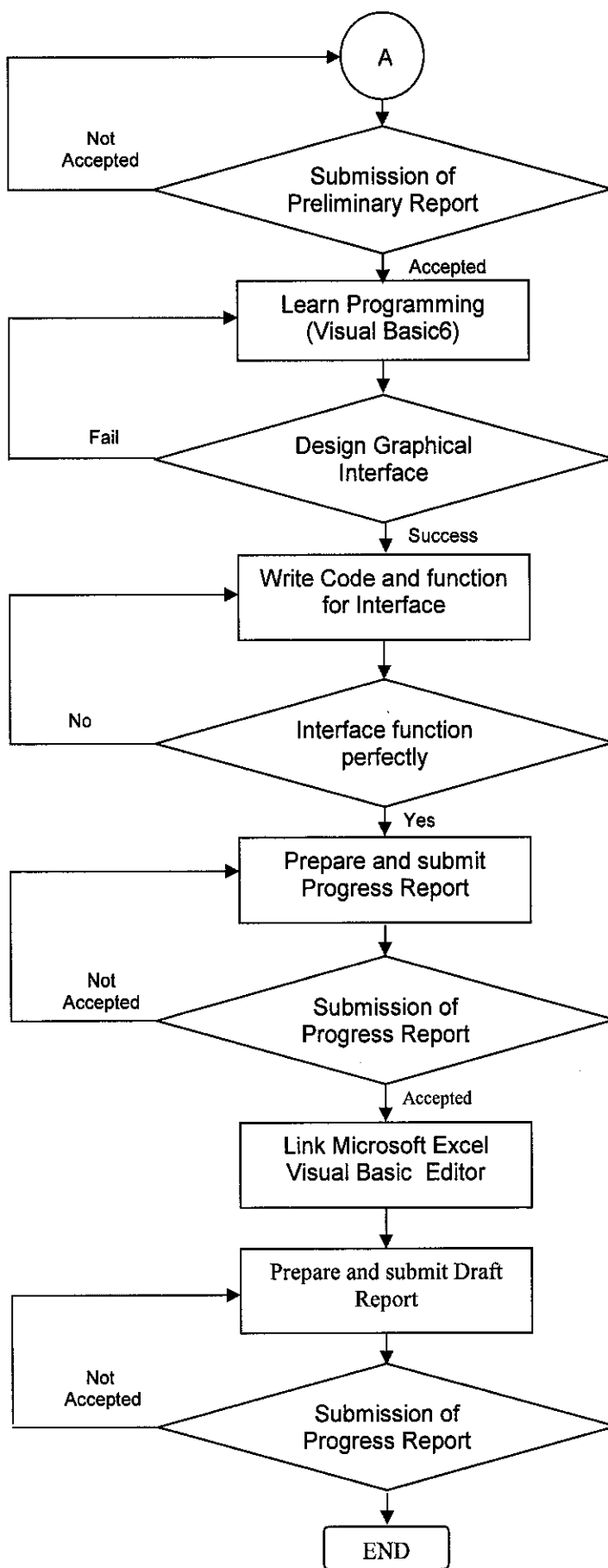


Figure 3.1: Project Flow Chart (FYP1)

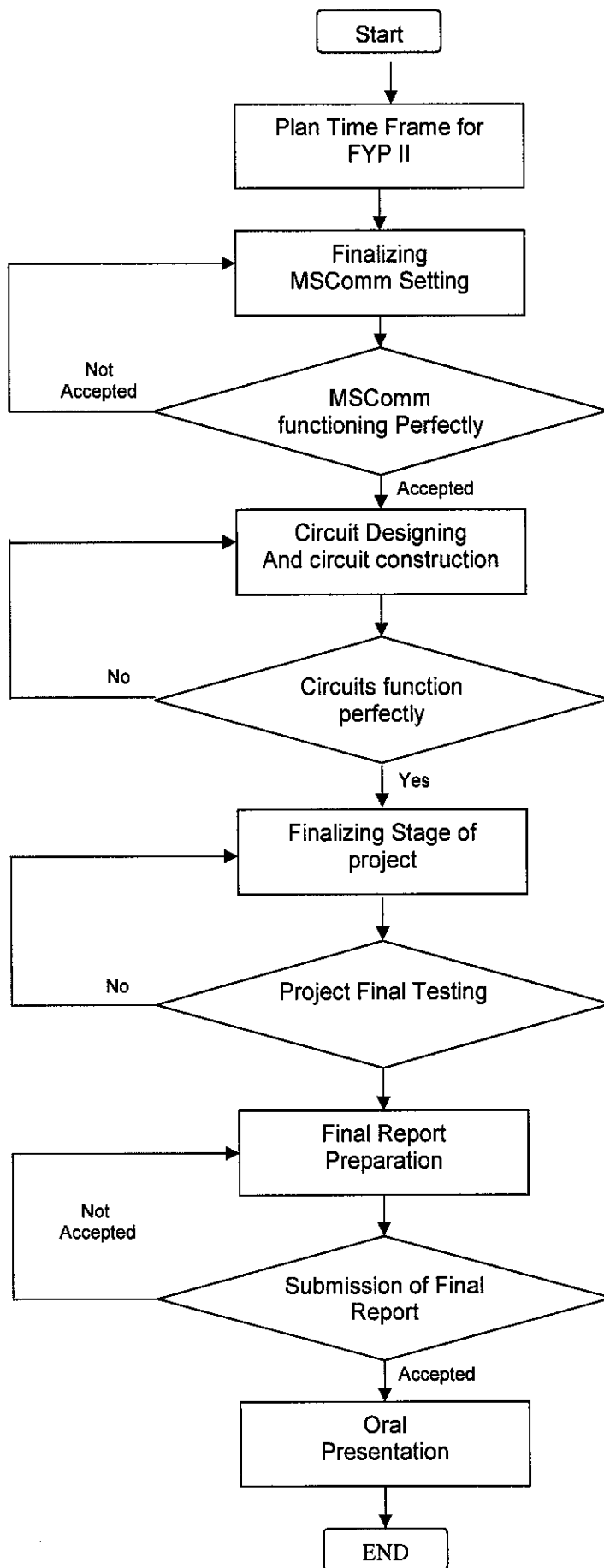


Figure 3.2: Project Flow Chart (FYP2)

The methodology of the project can be divided into three main parts, which include research, Visual Basic Interface and hardware implementation as shown in Figure 3.3.

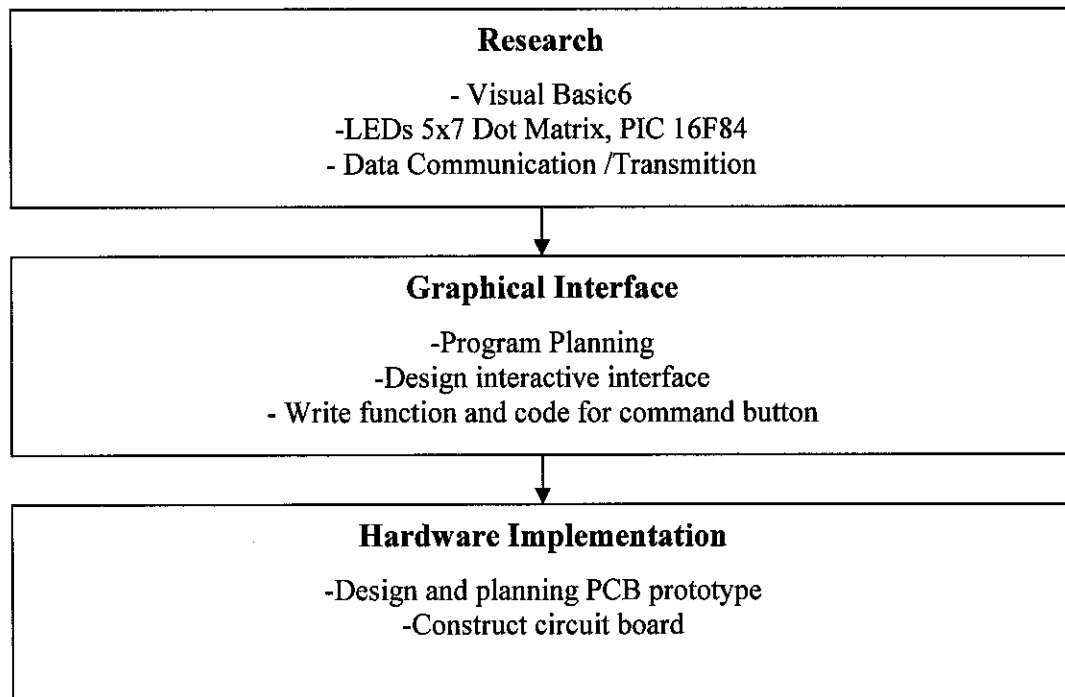


Figure 3.3: Methodology of LEDs Display Board design

3.3 Research

The project starts with research on LEDs 5x7 Dot Matrix and PIC16F84, which covers the characteristic, connections and pin function in the LED matrix Display itself. Most of the research was on how to design an interactive graphical interface through Visual Basic6. Furthermore the research also includes data transmission and serial communication that helps to send data from computer to display board.

3.4 Graphical Interface

This section includes program planning and how to design interactive graphical interface using Visual Basic6 software. The first step in writing instructions to carry out a task is to determine what the output should be (exactly what the task should produce). The second step is to identify the data, or input, necessary to obtain the output. The last step is to determine how to process the input to obtain the desired

output (to determine what formulas or ways of doing things can be used to obtain the output) [20].

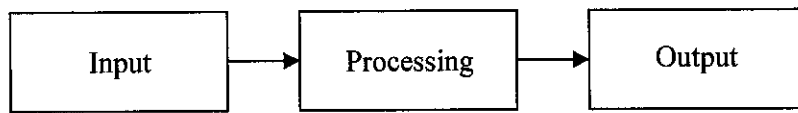


Figure 3.4: Pictorial Representation of problem Solving

3.4.1 Program Planning

An effective program planning will enable programmer to manage time efficiently and help to design error-free programs that produce the desired output.

i. Analyze

Define problem. To make sure what the output should be and have a clear idea of what input data are given, and the relationship between the input and the desired output.

ii. Design

Plan the solution for the problem. To find a logical sequence of precise steps that solves the problem. Such a sequence of steps is called an algorithm.

iii. Choose the interface

Select the objects (text boxes, command buttons, etc.). To determine how the input will be obtained and how the output will be displayed. Then create objects to receive the input and display the output. Also, create appropriate command buttons and menus to allow the user to control the program.

iv. Code

Translate the algorithm into a programming language. Coding is the technical word for writing the program. During this stage, the program is written in Visual Basic and entered into the computer.

v. Test and debug

Locate and remove any errors in the program. Testing is the process of finding errors in a program, and debugging is the process of correcting errors that are found. As the program is typed, Visual basic points out certain types of program errors. Others types of errors will be detected when the program is executed.

However, many errors due to typing mistakes, flaws in the algorithm, or incorrect usages of Visual Basic language rules can be uncovered and corrected only by careful detective work.

vi. Complete the documentation

Organize all the material that describes the program. Documentation is intended to allow another person to understand the program. Documentation might also consist of a detailed description of what the program does and how to use the program.

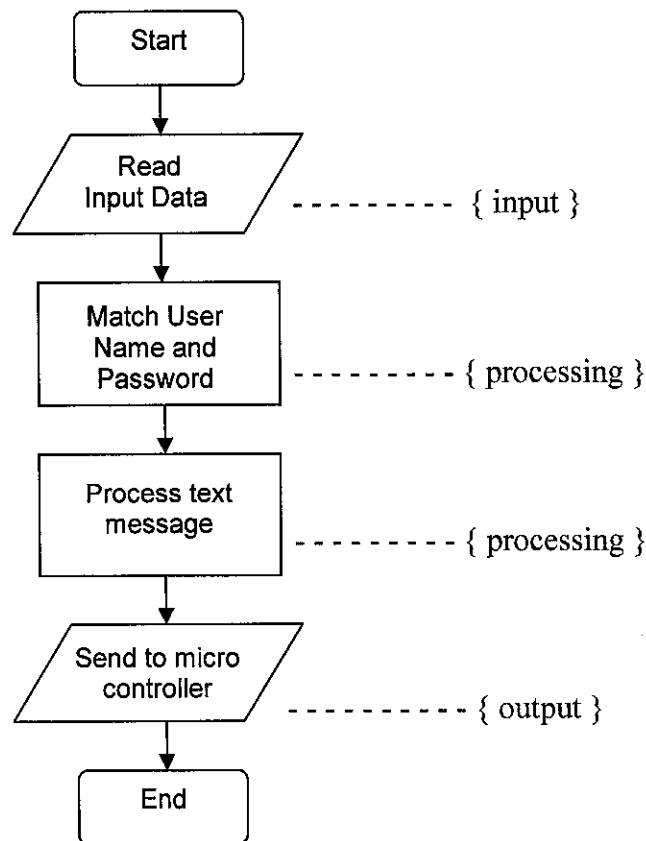


Figure 3.5: Flowchart for graphical Interface

3.5 Hardware Implementation

The last part of the project is the hardware construction. This stage start with the PCB design using Eagle software then follow by constructing of the PCB itself.

3.5.1 Designing and Planning a PCB

Evaluation Printed circuit board prototype design is a key step in evaluating, testing , and proving novel circuit concepts [15-16], especially when considering IC (integrated circuit) implementations are costly, time-consuming, and difficult to debug, which is why PCBs precede ICs in the experimental and developmental process, whenever possible. The popularity of PCB prototype design has therefore spread from industry to academia. However, evaluation time, measurements, and results, without a robust test plan, can be adversely affected, ultimately delaying the schedule and sacrificing reliability. Careful planning will avoid most of these drawbacks by addressing them early in the PCB design phase. For example choosing the most appropriate equipment and test probes for the kinds of signal to be tested, by carefully placing additional test points, connectors.

From a testing perspective, a robust test plan adds redundancy to the PCB prototype and thus enables signal extraction with minimal interference and maximum fidelity: in other words, the test plan ensures the signal measured is a true representation of its source signal and the extraction method is benign to the system. Moreover, a robust test plan is practical, simple, convenient, and cost-effective. Recognizing that various test signal and specifications require specific equipment, test probes, connectors, and circuits is important because they determine they validity of the experiments and respective results.

i. Equipment

Based on the type of signals to be tested (e.g. analogue or digital, voltage or current, Dc or Ac, high or low frequency, large or small signal) and the specifications (e.g. unity-gain frequency, common-mode range, power supply rejection ration, etc.), suitable equipment must be identified to reveal and measure these signals accurately

and effectively. Power supplies, for example, provide DC and bias currents, and their accuracy will vary with vendor. Function and signal generators produce various types of waveforms and signals but their output power is limited. Multi-meters measure Dc and Ac voltages, currents, and resistances, and like power supplies, their accuracy will vary with vendor.

Oscilloscopes capture transient responses and Ac signals but they have finite input resistance and parasitic input capacitance which will affect the signal measured and its degree of change will depend on the circuit being tested. Spectrum analyzers capture the frequency composition of a given signal including distortion, system noise and environmental noise which is why low noise experiments may be warranted. Network/impedance analyzers extract S-parameters in amplitude and phase which are useful for examining impedance-matching and loop-stability performance. Semiconductor parameters analyzers are versatile and programmable and capable of characterizing discrete semiconductor devices and ICs (integrated circuits).

Probe stations are very useful for debugging ICs for accessing signals embedded in the die and exposing and severing metallic signal paths with a laser beam. The electrical characteristics of the probes used are critical given that they have limited bandwidth, resistance, and physical resolution. As seen, the electrical characteristics of the equipment alter the signals being measured and it is up to designer to appropriately choose the equipment so as to minimize these effects to acceptable and possibly negligible levels. More detailed equipment information about equipment can be found in [17-18].

ii. Test Probes

Most equipment includes its own set of test probes for a basic configuration but unfortunately their electrical characteristics are not universally benign to a measurement, to the signal being tapped especially for advanced measurements in electrically harsh environments.



Figure 3.6: Various Test Probes

As noted, selecting the proper test probe for a given experiment is critical because its loading resistance and capacitance, bandwidth, and frequency response alter the fidelity of the signal and measurement.

iii. Connectors and sockets

The physical constraints of on-board connectors and chip sockets are many times important considerations in designing a PCB and testing a batch of silicon dies for convenience and like test probes their electrical characteristics also affect signal fidelity. Test point connectors, like through-hole points, SMT (surface mount technology) test points, SMT ball and socket connectors, switches and jumpers are selected to match the test probes and signal requirements of the experiment. Through-hole test points are for PCBs with through-hole chips like SIP and/or DIP (single-and/or dual-in line packages). Switches and jumpers are convenient means of enabling and disabling parts of a system which is especially useful in debugging phase of a PCB design [19].

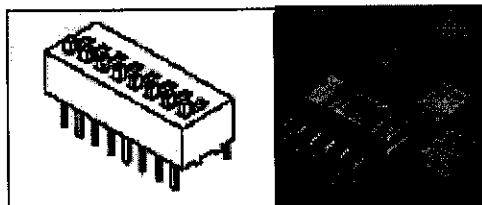


Figure 3.7: Test-point Connector

On the other hand, sockets for chip-packaged prototypes (e.g., plastic leadless chip carrier (PLCC), quad flat no-lead (QFN), small outline transistor (SOT), small shrink outline package (SSOP), thin small outline package (TSOP) and so on) are useful for easily testing several similar prototypes without having to solder or de-

solder components on the board which would otherwise damage the PCB. In all, connectors and sockets not only extend but also define the physical constraints of a given test probe link and silicon chip prototype.

iv. Circuits

During the circuit construction, testing and measuring low power, low energy signals is difficult because any parasitic in the equipment, test probe, connector, and PCB will unavoidably affect the particular circuit. Application of specific circuits like low noise and low input-offset amplifiers, drivers, and buffers can be used to isolate and contain these signals before exposing them to the parasitic elements as mention above.

v. Test Plan

There are certain complexities and extensive set of considerations of designing a PCB (i.e., equipment, test probes, connectors, circuits, and measurement) as discussed above therefore a test plan has to be devised carefully. The plan must start with circuit testing. The parameters have to be ascertained first then only followed by the electrical sensitivity of each signal to parasitic resistance, capacitance, and inductance. The signal's accessibility has to defined and comprehended into the PCB. Once these constrains are defined and redundancy explored, equipment, test probes, connectors, and circuits for each signal can be identified at which point the simulation process of the PCB schematic can commence.

PCB simulation must emulate and evaluate all the experimental procedure to be performed on the circuit. Only after carefully and fully simulating the projected PCB design and test plan only then layout process begin. Simulating the test procedure ensures that the proper nodes to be accessed are defined. Predicting problems before the layout process is important because redundancy and access to miscellaneous nodes in the system can potentially ease the debugging effort.

Finally when testing the PCB, the procedure must start by testing the connectivity of the PCB without the device under test (DUT) to ensure the PCB is operational then

followed by adding the DUT and testing only DC measurements to verify the device is properly biased. The debugging phase will be start if problem exist before any other measurements are make. If the system is properly biased, functional measurements will be performed to test the operational status of the various system blocks.

The test plan which is always being undervalued is a pivotal part of the project, especially when managing design risk and test time. With a robust test plan, evaluation time can be decreased, complex measurements eased and accurate results verified, thereby increase the chances of meeting project deadlines.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Result

4.1.1 Visual Basic Interface

By using reference book and some related website as guidance, the Visual Basic Interface for the project is created.

i. Main Screen Interface

This interface had been changed for a better work sheet from previous design. Only ***Lable*** control being used in this interface and the command button are ready to be use with the code written for it. The code for the interface is in **Appendix B**.

- Login → Once click will go to *Login Screen Interface*.

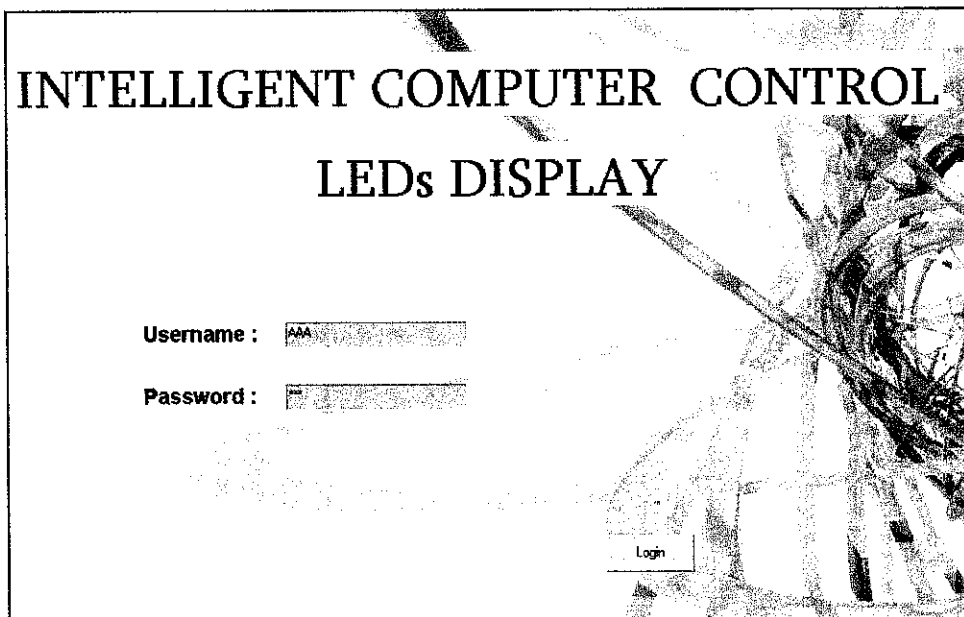


Figure 4.1: Main Screen Interface

ii. Login Screen Interface

For security reason, the software begins by prompting the user to enter an accepted user name and password. This is to limit the unauthorized access.

Confidential username and password is stored in the Microsoft Access file.

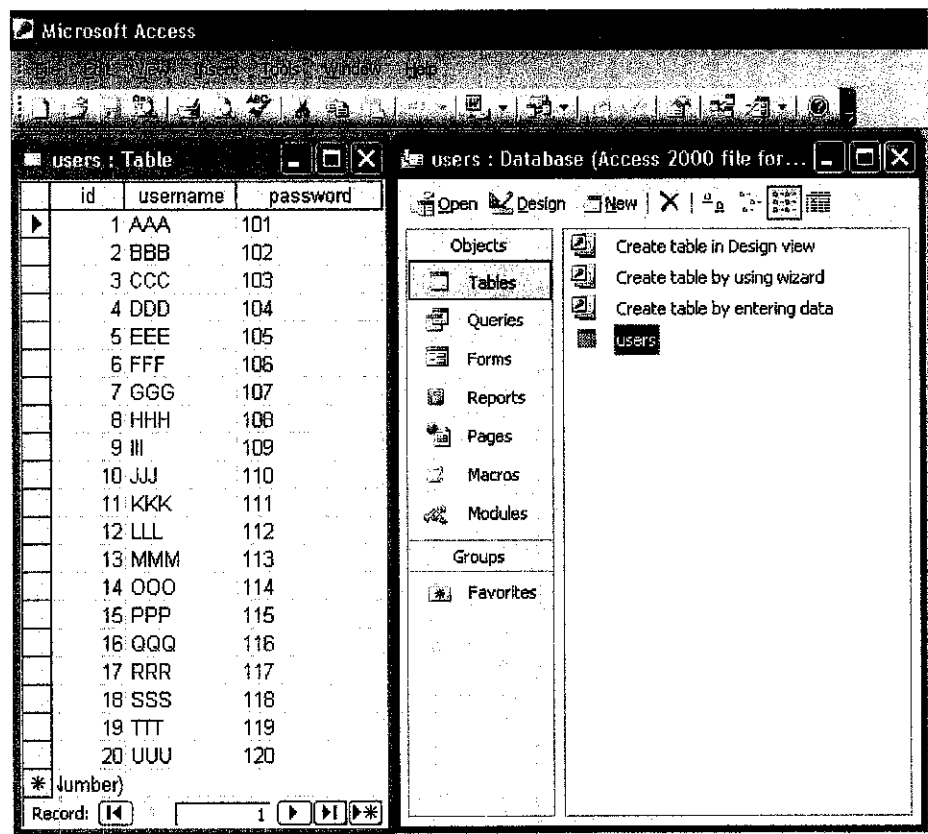


Figure 4.2: Microsoft Access file with Confidential Username and Password

This Microsoft Access file is link with the *Login Screen Interface*. System will start checks and match the username and password given by user with the one stored in Microsoft Access file. User only are allows to proceed to the next interface if it is match.

The *Login Screen Interface* is designed to protect the system from non- authorised people from using it. The command buttons are ready to be used with the code been write for it. The code for the interface is in **Appendix C**.

- Send → Once click will go to *Main form*.
- Exit → Once click will end the program.

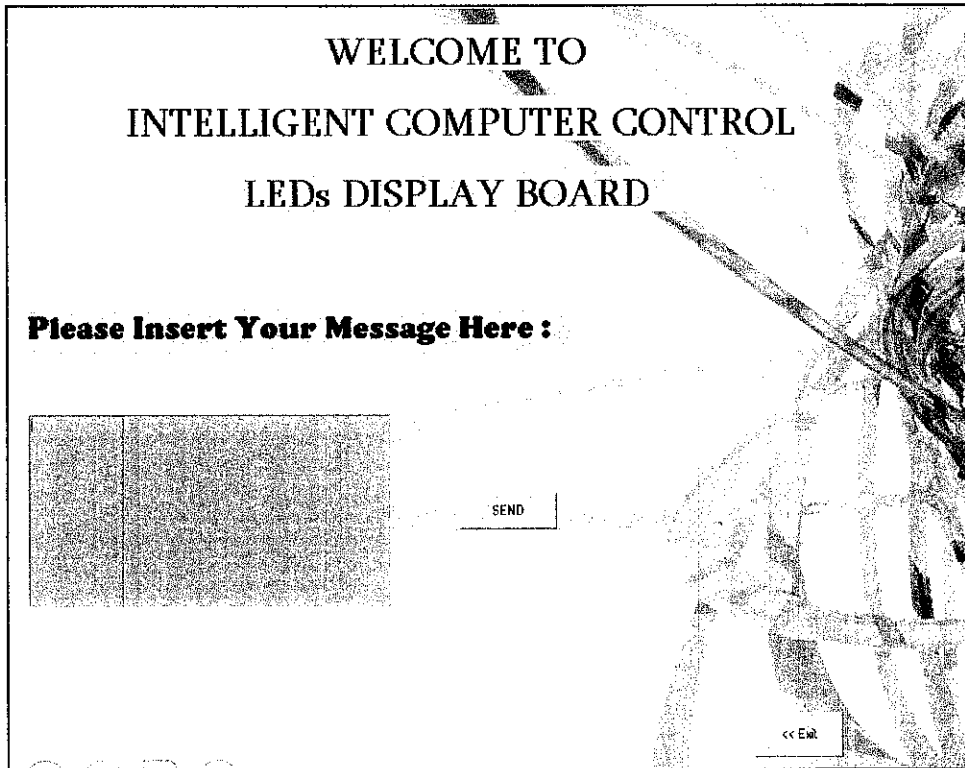


Figure 4.3: Login Screen Interface

iii. Message Interface

This Message Interface will be appeared once the user name and password is match. The command button are ready to be use with the code been write for it. The code for the interface is in **Appendix D**.

- Back → Once click will back to previous form.

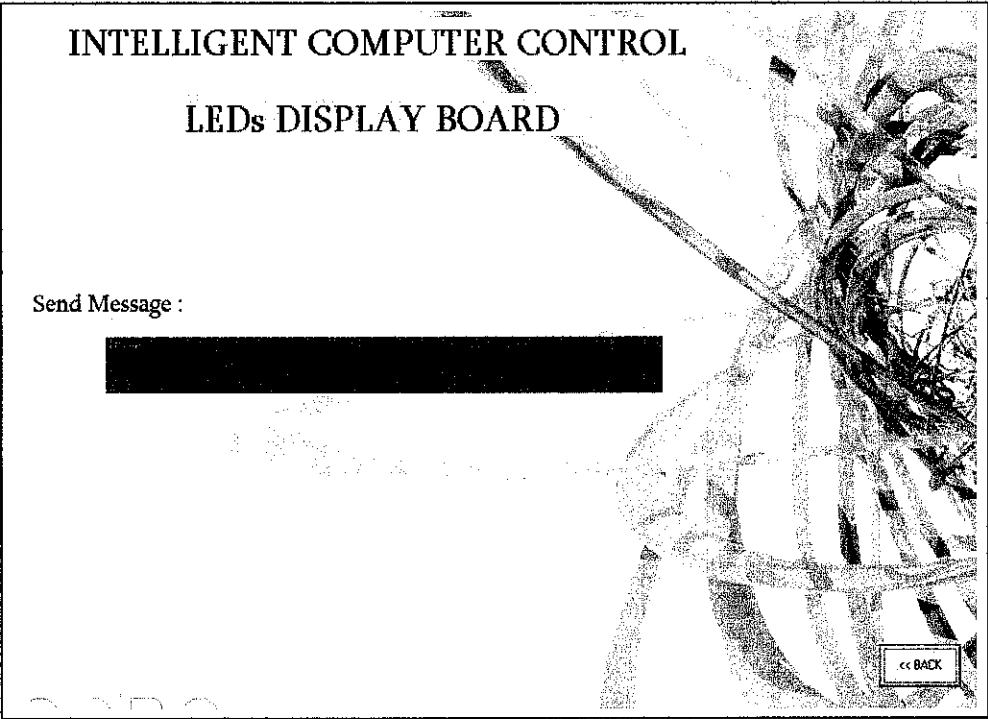


Figure 4.4: Message Interface

4.1.2 LEDs Display Board

The flow of the whole project begins at user stage and end at LEDs Display Board:

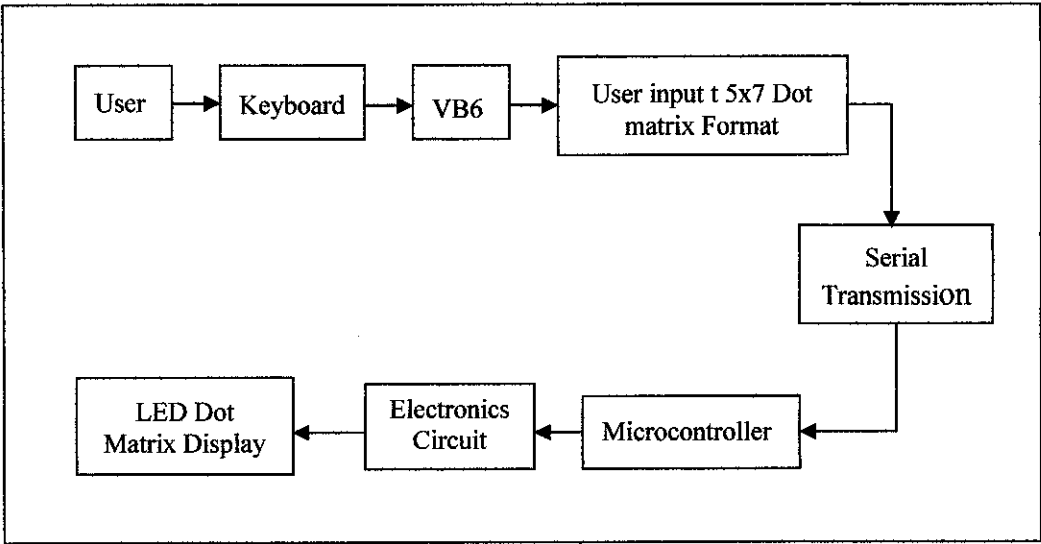


Figure 4.5: The Flow Of Whole Process

The message board system begins with a software shell. This software is programmed in Visual Basic6 due to its ease of implementation for graphical interfaces. For security, the software begins by prompting a user to enter an accepted

password. Through this way, the chance of unauthorized access is greatly limited. Upon entrance of an accepted password, this will allows the user to enter a text message to be displayed on the message board.

The formatted message is transmitted to Maxim’s MAX232 integrated circuits via serial communication. The MAXs provide an interface between the microcontroller and the 5 x 7 LED matrices. One MAX232 interfaces five LED matrices by connecting the pins on the integrated circuit to the respective pins of the five LED displays. The schematic diagram is shown in **Appendix E**.

These are the circuit layout which needed to produce PCB.

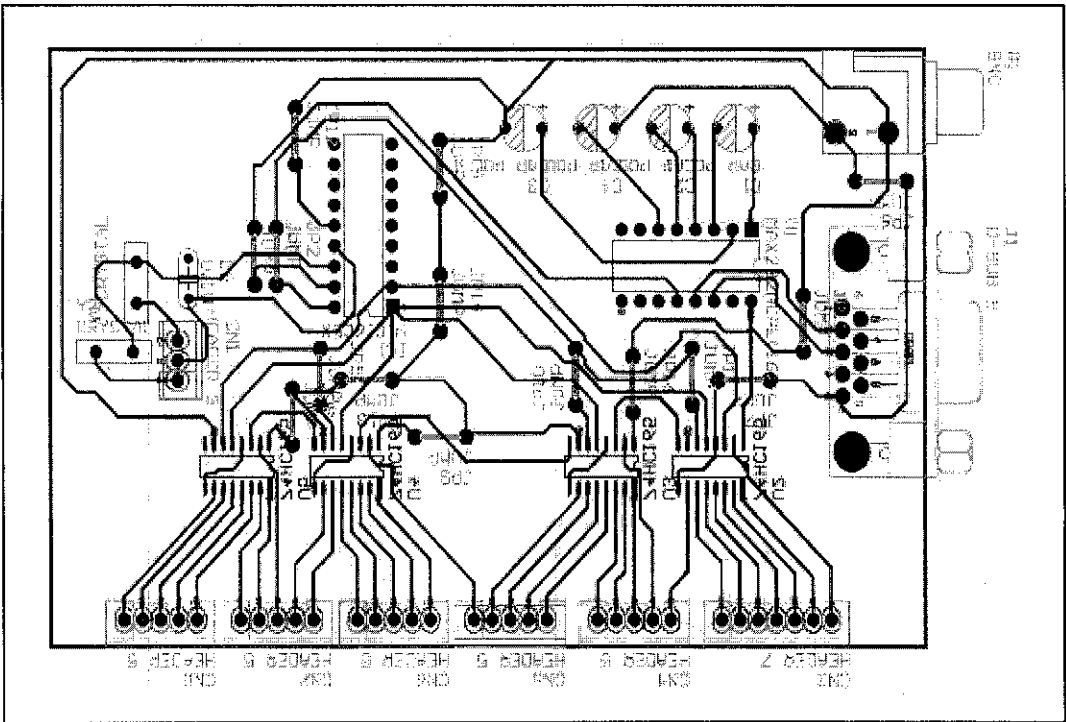


Figure 4.6: Component layout

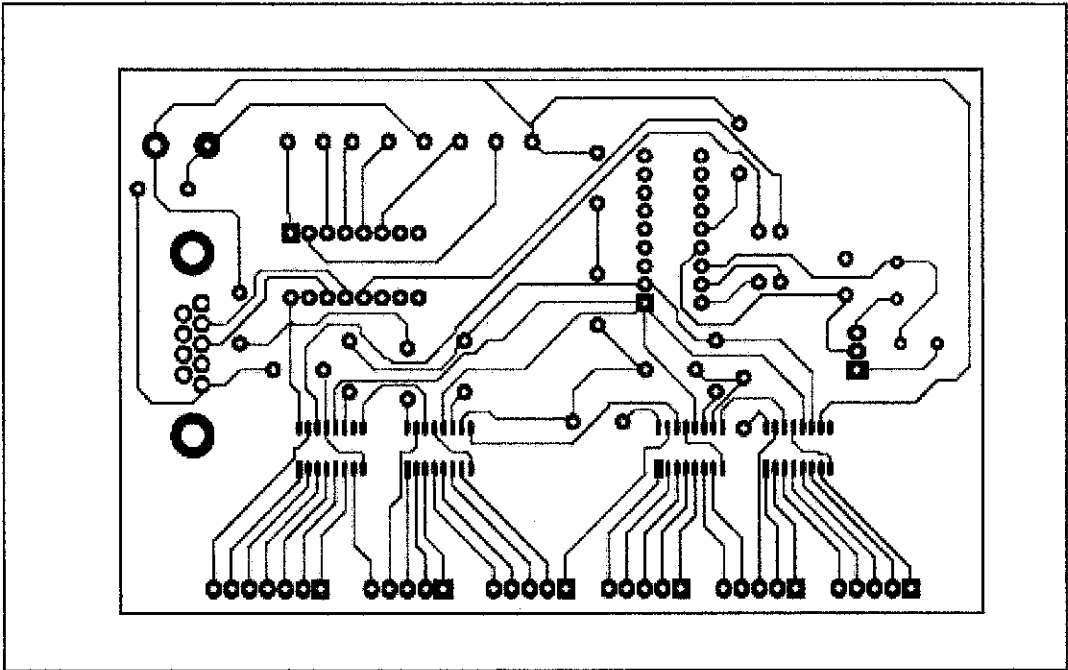


Figure 4.7: PCB Layout

These are the Printed Circuit Board of LEDs Display Board and LEDs Display Board controller.

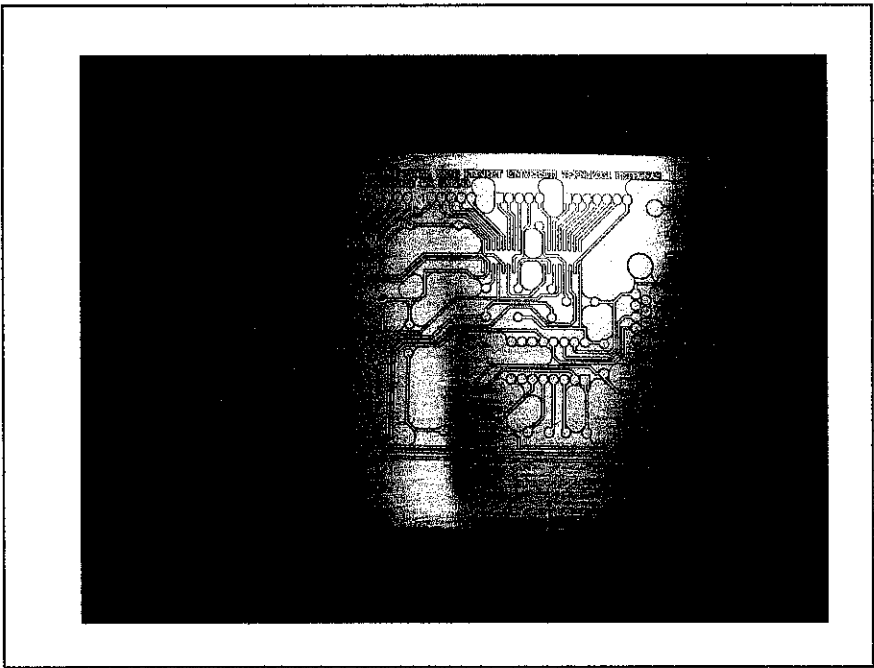


Figure 4.8: Display Board Controller PCB (Before)

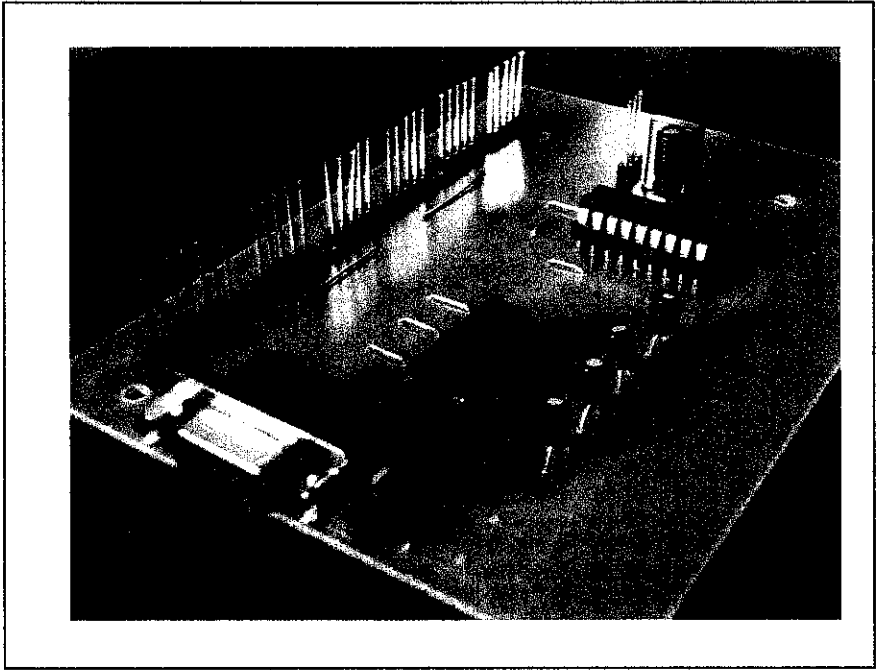


Figure 4.9: Display Board Controller PCB (After-Front View)

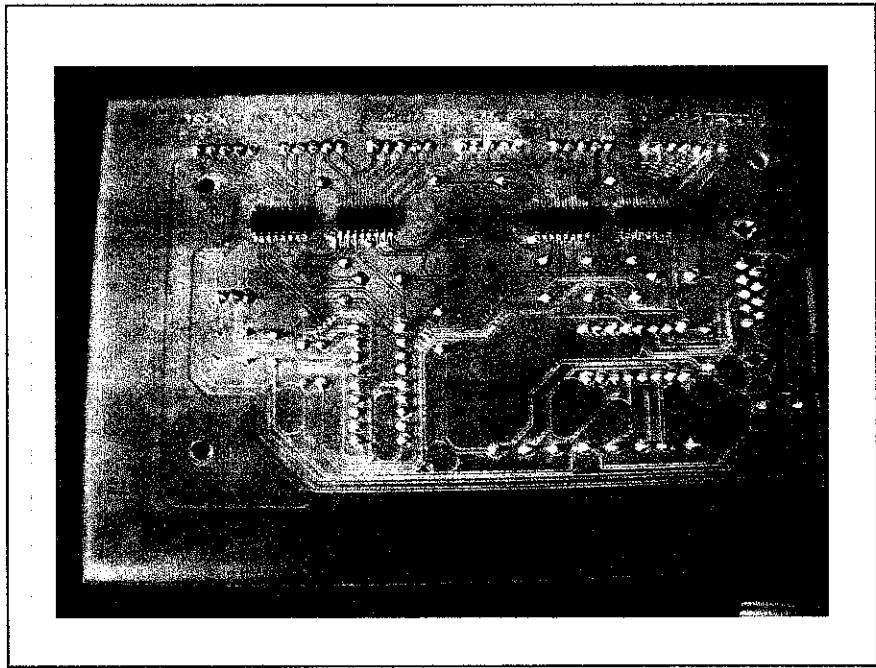


Figure 4.10: Display Board Controller PCB (After-Back View)

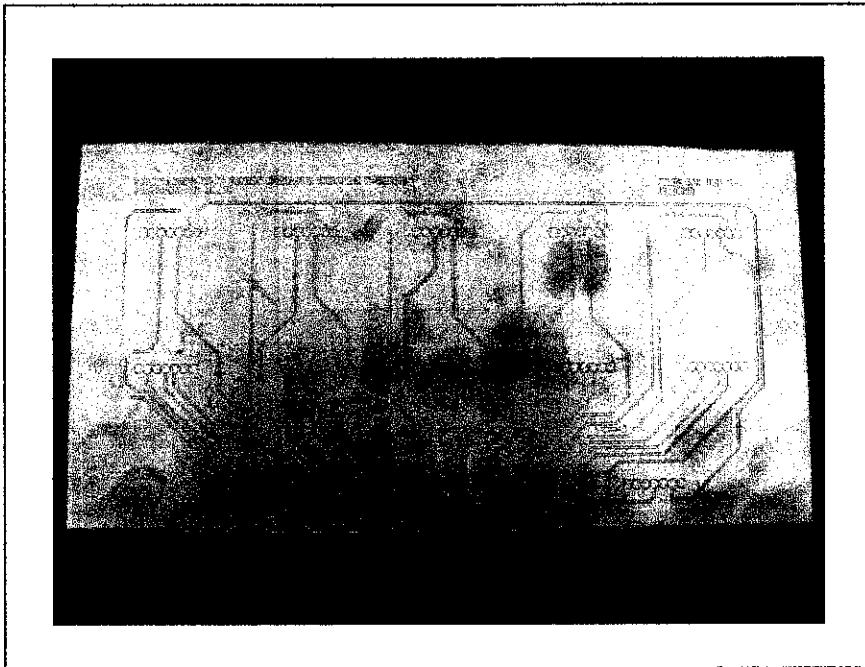


Figure 4.11: LEDs Display Board PCB (Before)

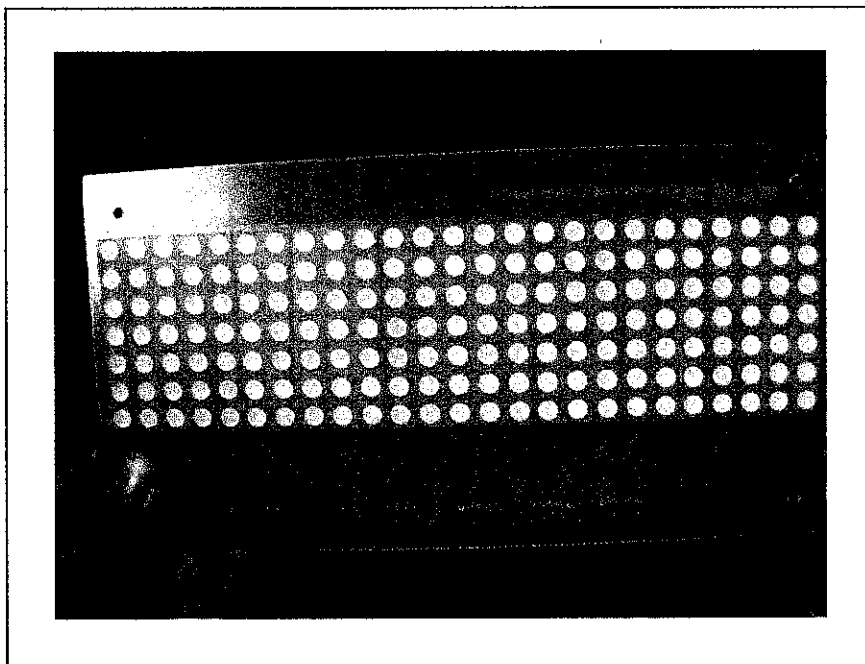


Figure 4.12: LEDs Display Board PCB (After-Front View)

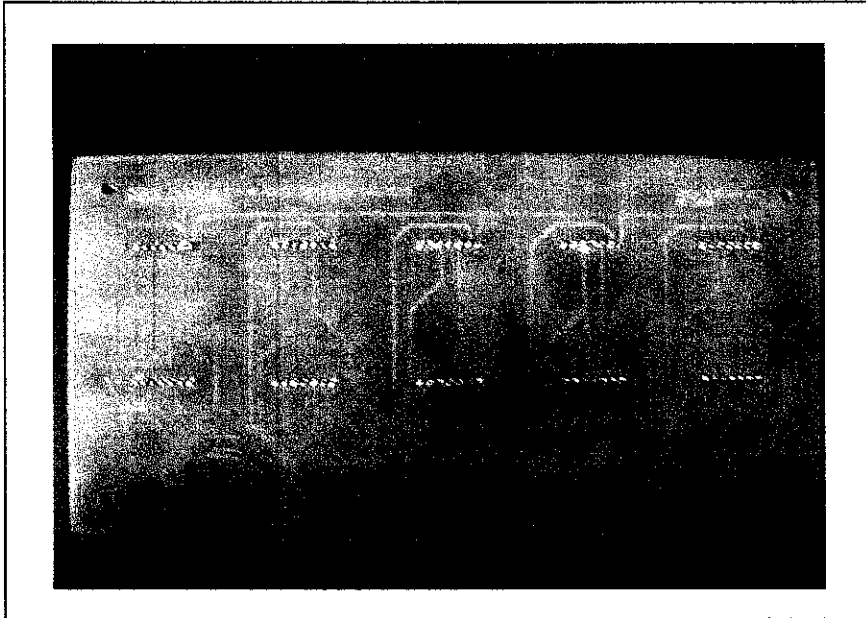


Figure 4.13: LEDs Display Board PCB (After-Back View)

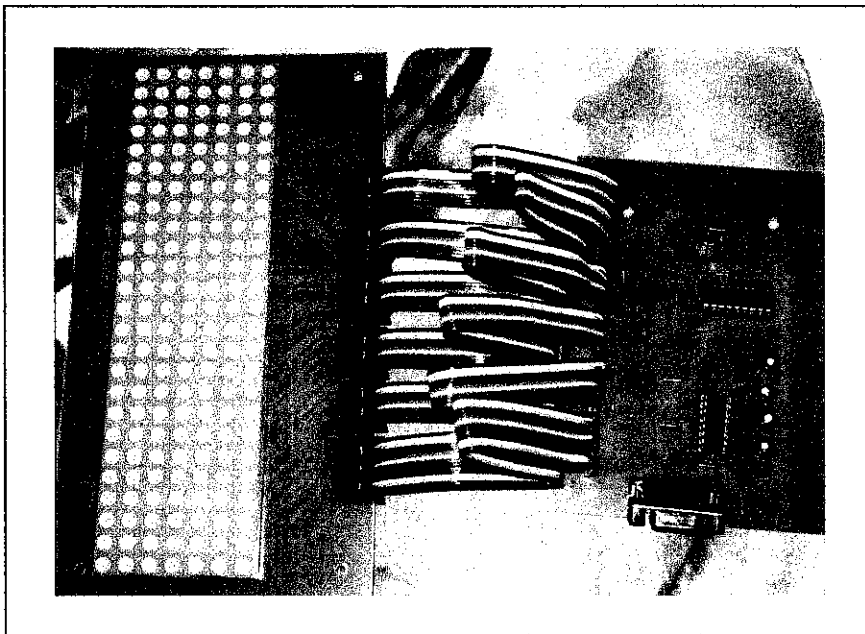


Figure 4.14: LEDs Display Board

4.2 Discussion

Visual Basic is a visual and events driven programming language. In Visual Basic programming is done in a graphical environment and because of users may click on a certain object randomly, so each object has to be programmed independently to be able to response to those actions (events), each has its own program codes, and each can be executed independently and at the same time each can be linked together in one way or another.

The major element of Visual Basic has been introduced. The process to build the application consists of three steps.

- i. Creating the interface – Drawing the form and controls.
- ii. Setting properties – Setting the values of properties for the controls added to the form.
- iii. Code writing – Creating the event procedures that will occur when application runs.

First step to start Visual Basic6 is to design form and add ***Lable***, ***TextBox***, and ***CommandButton*** controls to a form according to the needs. The process for changing the properties of controls by setting Caption, Text, Name, Locked, TabStop properties and some additional properties of the controls is a bit complicated. By adding the ***ActiveX*** control to the Toolbox and used it in an application, can make the project even nicer.

Before writing an event procedure for the control to response to a user's input, certain properties have to be set for the control to determine its appearance and how it will work with the event procedure. The properties of the controls can be set in the properties window or at runtime.

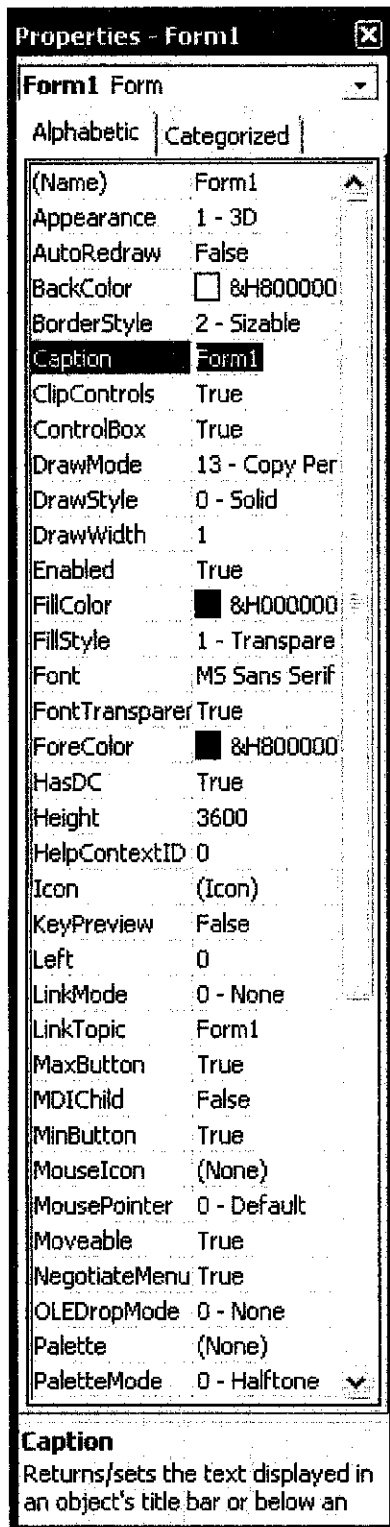


Figure 4.15: Properties Window

- i. In the properties window, the item appears at the top part is the object been selected in form.
- ii. At the bottom part, the items listed in the left column represent the names of various properties associated with the selected object while the items listed in the right column represent the states of the properties.
- iii. Properties is set by highlighting the items in the right column then change them by typing or selecting the options available. For example, in order to change the caption, just highlight Form1 under the name Caption and change it to other names.
- iv. Other things that changeable are the foreground and background color, font type and font size, enable or disable minimize and maximize buttons.
- v. The properties can also be changed at runtime to give special effects such as change of color, shape, animation effect and so on.

Microsoft Access is the Visual Basic's native format for database file. In database terminology, a *file* consists of *table*, which consists of *rows* or records, which consist of *columns* or *fields*. Each record is identified by its *key* field. One record is always the *current* record. A data control is placed on a form to link it to a database file and

a particular table within that file. In the other hand, data-bound controls link to a data control for the table name and to a particular field within the table.

While for the hardware part, in order to accomplish the receiving of the message, a MAX232 transceiver is connected to a Com Port of the PC via a serial cable and to the Rx1 pin 13 of the MAX232. The message was then transmitted to the PIC from RxD1 (pin 12 of the MAX232) to PORTA 0 RA0 (pin 17 of the PIC16F84). The message is sent via asynchronous serial communication. The message was saved internally on the PIC (EEProm memory area) and read back to be displayed to the LED sign.

The LED display has 32 input lines to show a message. These input lines consist of 7 rows for each 5x7 LED and five units columns for each 5x7 LED. The Rows of the LEDs were tied together so that only 7 inputs were needed to control all of the Rows. Seven units of current limiting resistors were put in place between the outputs from the PIC to the Rows of the LEDs to prevent overloading (burnout) of the display. The Columns of the LED display now needed 25 inputs (5 x 5x7).

In order to handle the 32 input lines four units of 8bit Motorola shift registers were used. By using the shift registers the data for the display is loaded into the shift registers serially with one data line, one clock shift line and one latch line from the PIC. When the latch line is enabled the data is then put out in a parallel manner (all 32 bits at once). This display is using one of the shift registers to enable the 7 rows and the first column of the display. The other three shift registers will control the columns. By using the shift registers only three output lines from the PIC were needed. This was done by using PORTA 1 RA1 (pin18 of the PIC16F84) for the data in line pin 14 (serial in of the Shift Register), PORTA 2 for the latch line pin 12 (latch of the Shift Register), and PORTA 3 for the shift clock line pin 11 (clock of the Shift Register). The entire display was controlled by only five output and one input lines of the PIC16F84.

After several attempt of Visual Basic Interface circuit designing, the Visual Basic coding for PIC16F84 and circuit testing has been considered the toughest part of all while the circuit schematic, components layout, and PCB layout were the challenges part as well as the circuit testing. This project had gave a greater understanding of the manipulation that can be done with only a small amount of instructions, and improved immensely understanding of the language. This project included the understanding of RS232 serial communications and the bit manipulation to store data, to read data and also output this data to the LEDs using shift registers.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The objective to enter message from the PC keyboard by authorise user and at the same time having this message scrolling on a 25 x 7 LED display is achieved. However the microprocessor programming part needs to be improved to have a better result. The literature review has been carried out in order to obtain a better understanding on message sending via PC keyboard using RS232 serial communication from a terminal program and saved to the PIC memory. Research and case studies helped made easier to understand how data can be send to a microprocessor.

According to the result obtained, the purpose of designing the interface is to make sure the message can be send effectively to the microprocessor using RS232 serial communication. At the same time to understand the flow of design and the usage of Visual Basic6 application for the purpose. Furthermore, chances to improve interpersonal skills by learning to communicate and interact directly with other personnel are also achievable.

5.2 Recommendation

There are few recommendations proposed to improve the Computer Controlled LEDs Display Board. The display board can be improved with some interesting functions. It can be improved by adding time display, animation display with multi color to make it as a multi-functional display board. Instead of using the serial communication as the interface between the controller and the display board, wireless communication system or SMS system can be implemented in order to obtain more effective result.

For those who interest to continue this project, wireless communication system can be implemented into this project to produce Wireless LED Display Board. Due to time limitation, this project was able to produce only running text display board. Hence, several attempts can be tried such as multi colour LEDs display board with reminder function can be try to make it more attractive.

REFERENCES

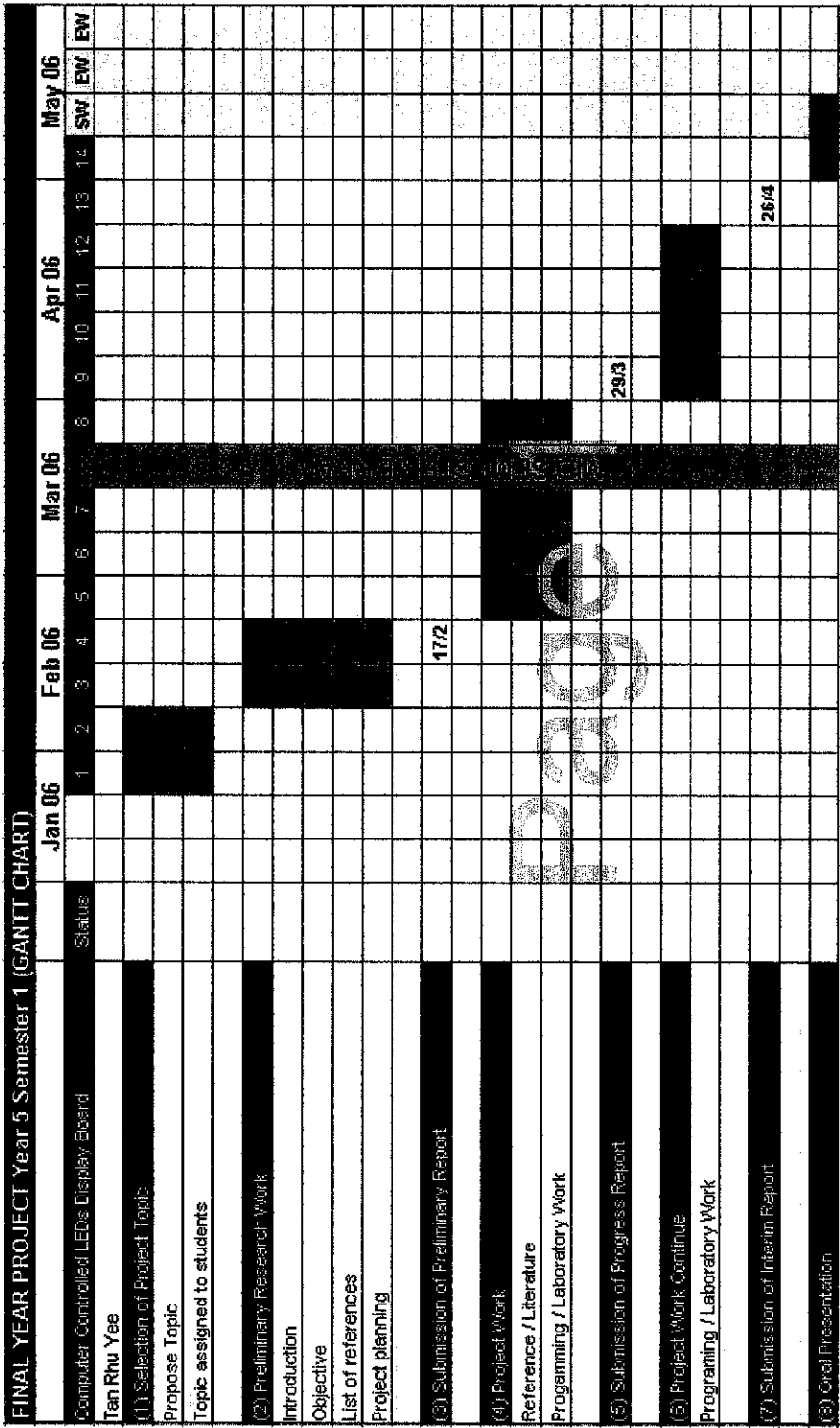
- [1] UCONN. (2005). *Senior Design Project*. University of Connecticut, School of Engineering. Retrieved on 19th Feb 2006 from:
<http://www.engr.uconn.edu/ece/SeniorDesign/projects>
- [2] Ultima. (2006). *LEDs Display Board*. Imaging System Pvt. Ltd. Retrieved on 10th Feb 2006 from: <http://www.ultimaimaging.com/>
- [3] BusinessOL. (2006). *Spectrum Scoreboards and Outdoor Message*. Retrieved on 10th Feb 2006 from: <http://www.specorp.com>
- [4] RS Components Ltd. (2005). *LEDs Dot Matrix*. Retrieved on 2nd March 2006 from: <http://www.rshongkong.com/cgi-bin/bv/rswww/>
- [5] Conjecture Corporation. (2006). *What is RS-232*. Retrieved on 21 Jan 2006 from: <http://www.wisegeek.com/what-is-rs-232.htm>
- [6] Craig Peacock. (2005). *Interfacing the Serial / RS232 Port*. Retrieved on 27 Feb 2006 from: <http://www.beyondlogic.org/serial/serial.htm>
- [7] Dave Berger. (2006). *USB Explained*. Compu-Kiss. Retrieved on 30 Jan 2006 from:
<http://www.compukiss.com/sandyclassroom/tutorials/article793.htm>
- [8] Tech-Pro. (2006). *How USB Works*. Tech-Pro Limited. Retrieved on 3rd Feb 2006 from: http://www.tech-pro.net/intro_usb.html
- [9] Coombs, Clyde F., Jr. (Ed.) 1995: *Printed Circuits Handbook, Fourth Edition*, McGraw-Hill.
- [10] Fribottes team. (2000). *Introduction : What is a PIC*. Retrieved on 13 Jun 2006 from: http://fribotte.free.fr/bdtech/pic/pic_en.html
- [11] Wikipedia. (2006). *Shift register*. Wikimedia Foundation, Inc. Retrieved on 10 Jun 2006 from: http://en.wikipedia.org/wiki/Shift_register
- [12] Wikipedia. (2006). *ASCII*. Wikimedia Foundation, Inc. Retrieved on 10 Jun 2006 from: <http://en.wikipedia.org/wiki/ASCII>
- [13] Colin Mitchell. (2003). *PIC Programing*. Retrieved on 15 Jun 2006 from:
http://users.tpg.com.au/users/talking/Burning%20a%20Chip_Page1.html
- [14] EAGLE Layout Editor. (2006). *CadSoft Online*. CadSoft Computer, Inc. Retrieved on 15 Aug 2006 from: <http://www.cadsoftusa.com/>

- [15] C.T.Robertson, *Printed Circuit Board Designer's Reference: Basic*, Upper Saddle River, NJ: Printice Hall Professional Technical Reference, 2004.
- [16] M. Montrose, *Printed Circuit Board Design Techniques for EMC Compliance: A Handbook for Designers*, 2nd Edition, New York, NY: IEEE Press, 2000
- [17] Agilent Technologies. (2005). *Test and measurement equipment*. Agilent Technologies, Inc. Retrieved on 22 Jun 2006 from:
<http://www.home.agilent.com/agilent/home>.
- [18] Tektronix. (2006). *Test and Measurement*. Tektronix, Inc. Retrieved on 22 Jun 2006 from: <http://www.tek.com/>
- [19] Mill-Max. (2004). *Maximum Connections*. Mill-Max Mfg. Corp. Retrieved on 12 Sept 2006 from: <http://www.mill-max.com/index.cfm>
- [20] David I. Schneider . 2004: *An introduction To Programming Using Visual Basic 6.0, Fourth Edition*, Prentice Hall.

APPENDIX A

Gantt chart of final year project activities

A1. Planning activities for first semester



A2. Planning activities for second semester

FINAL YEAR PROJECT Year 5 Semester 1 (GANTT CHART)															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Computer Controlled LEDs Display Board															
Tan Rhu Yee															
(1) Project Work Continue															
MSComm Control Setting															
(2) Submission of Progress Report 1				18/8											
(3) Project Work Continue															
Practical/Laboratory Work															
Programming / Model project															
(4) Submission of Progress Report 2								22/8							
(5) Project work continue															
Practical/Laboratory Work															
(6) Submission of Draft Report										9/10					
(7) Submission of Dissertation Final Draft												18/10			
(8) Project work continue															
Finalizing the project															
(9) Oral Presentation														4/12	
Finalizing the project															
(10) Submission of Project Dissertation (Hardbound)															22/12

APPENDIX B

Main Screen Interface Code

Code for Main Screen Interface

```
Private Sub cmdLogin_Click()
    Dim rs As New ADODB.Recordset
    Dim cn As New ADODB.Connection
    Dim sql As String
    Dim uname As String
    Dim pword As String

    cn.ConnectionString = "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
App.Path & "\users.mdb;Uid=Admin;Pwd=123;"

    cn.Open (cn.ConnectionString)
    sql = "SELECT [username],[password] FROM [users] WHERE [username]=" &
txtUN.Text & " AND [password]=" & txtPW.Text & ";"
    Set rs = cn.Execute(sql)

    If rs.EOF = True Then
        MsgBox ("Login Is Incorrect!")
        txtUN.Text = ""
        txtPW.Text = ""

        Exit Sub
    End If

    If ((rs("username") = txtUN.Text) And (rs("password") = txtPW.Text) And
(rs("username") <> "")) Then
        frmMassage.Show
        txtUN.Text = ""
        txtPW.Text = ""
    Else
        MsgBox ("Login Is Incorrect!")
    End If

    cn.Close

End Sub
```

APPENDIX C

Login Screen Interface Code

Code for Login Screen Interface

```
Private Sub cmd_cancel_Click()
cmd_cancel.Enabled = True
On Error GoTo ErrHandler

    MSComm1.Output = CStr(Text1)
    frmMain.Label3.Caption = CStr(Text1)
    cmd_cancel.Enabled = False
    frmMain.Show

Exit Sub
ErrHandler:
MsgBox Err.Description, vbInformation, "Output Error"
Err.Clear
End Sub

Private Sub Command1_Click()
Me.Hide
frmLogin.Show
cmd_cancel.Enabled = True
End
End Sub

Private Sub Form_Load()
On Error GoTo ErrHandler
    MSComm1.CommPort = 4

    If Not MSComm1.PortOpen Then
        MSComm1.PortOpen = True
    End If

Exit Sub
ErrHandler:
MsgBox Err.Description, vbInformation, "Open Port Error"
Err.Clear
End Sub

Private Sub Form_Unload(Cancel As Integer)
On Error GoTo ErrHandler
    If MSComm1.PortOpen Then
        MSComm1.PortOpen = False
    End If
```

```
Exit Sub  
ErrorHandler:  
MsgBox Err.Description, vbInformation, "Close Port Error"  
Err.Clear  
End Sub  
  
Private Sub MSComm1_OnComm()  
MsgBox MSComm1.Input, vbInformation, "On Comm"  
End Sub  
  
Private Sub Text1_Change()  
cmd_cancel.Enabled = True  
End Sub
```


APPENDIX D

Message Interface Code

Code for Message Interface

```
Private Sub CmmndNext_Click()  
'Dim frmLogin As New frmLogin  
'Me.Hide  
'frmMessage.Show
```

```
End Sub
```

```
Private Sub Command1_Click()  
Me.Hide
```

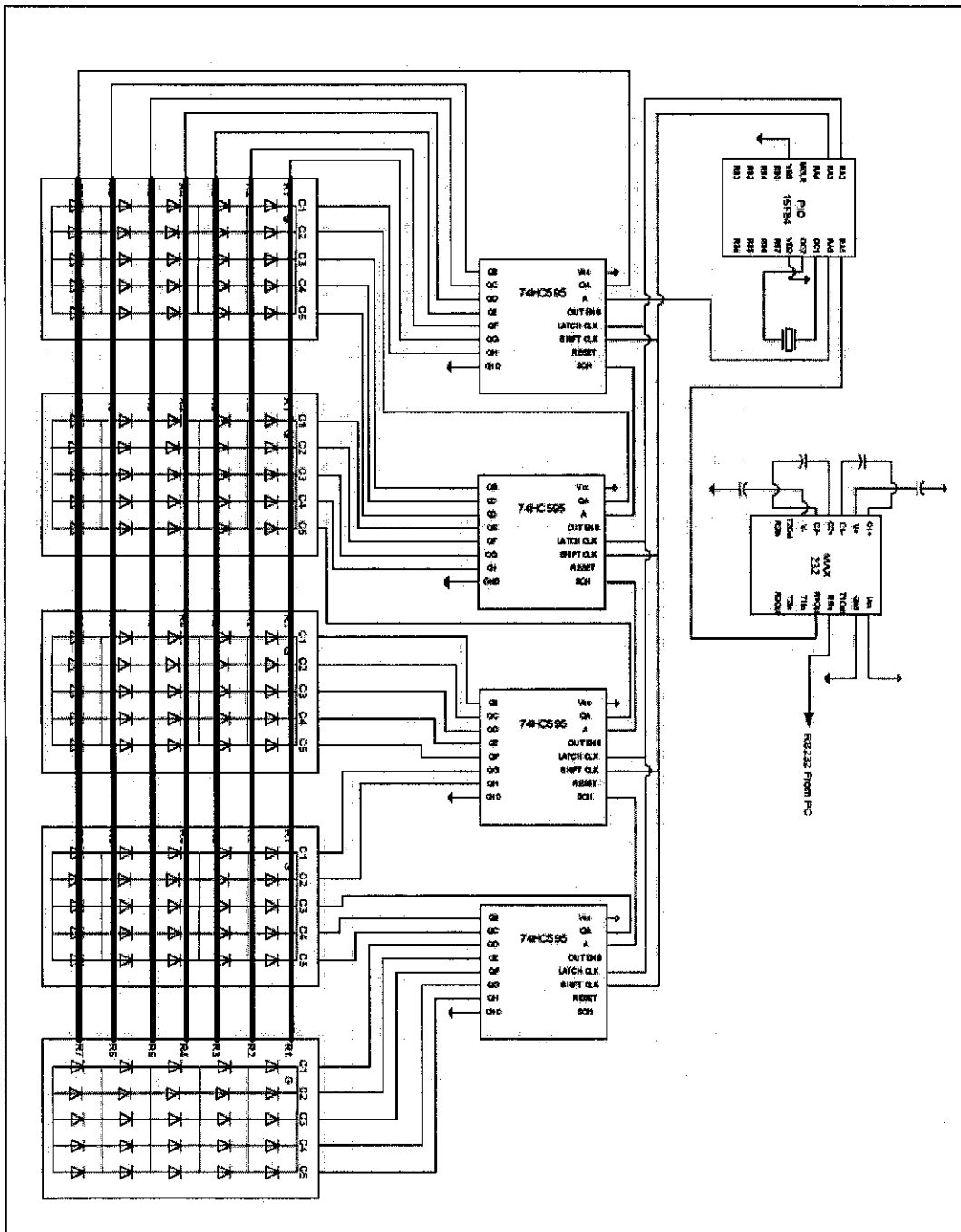
```
frmMessage.Show  
End Sub
```

```
Private Sub Command2_Click()
```

```
Me.Hide  
frmLogin.Show  
cmd_cancel.Enabled = True  
End
```

```
End Sub
```

Schematic diagram



APPENDIX F

ASCII Code

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_

Source: www.asciitable.com