DESIGN OF FLEXIBLE THREE PHASE LOAD EMULATOR

LOH SIEW CHOON

A project report submitted in partial fulfilment of the requirements for the award of the degree of Bachelor (Hons.) of Electrical and Electronic Engineering

Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2011

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _________________________

Name : ____LOH SIEW CHOON

ID No. : _____ 08UEB07802

Date : ___ 11 APRIL 2011_____

APPROVAL FOR SUBMISSION

I certify that this project report entitled **“DESIGN OF FLEXIBLE THREE PHASE LOAD EMULATOR”** was prepared by **LOH SIEW CHOON** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electrical and Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ___________________________

Supervisor : MR. CHUA KEIN HUAT

Date : ___________________________

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of University Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© APRIL 2011, LOH SIEW CHOON. All right reserved.

ACKNOWLEDGEMENTS

It is has been a long journey to accomplish this research project. However, it is also a very interesting and informative process for me. With the completion of this completion of this research proposal, I would like to take this opportunity to thank all the parties who assist and cooperate with me through the whole process of this research. Without those assistance and completion, our research proposal could not be competed in smooth and successful.

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, MR. CHUA KEIN HUAT for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement throughout my course of study in UTAR.

Thanks you!

DESIGN OF FLEXIBLE THREE PHASE LOAD EMULATOR

ABSTRACT

A three phase load emulator is a device that can emulate a load changes in the electrical network. The load emulator offers a more flexible platform for testing inverters in a laboratory environment. The load emulator can provide different load characteristics with regard to the typical load variation. There are 5 resistors with same value connected in parallel for each phase. The resistors can be connected and disconnected via the switching of solid state relays (SSR). The load emulator can be switched either by manual or by microcontroller. The solid state relays can be switched and controlled by microcontroller, Programmable Integrated Circuit (PIC) or by manual switch. For manual switching, it can be done by triggering a rocker switch. For microcontroller control, the desired load profiles can be keyed in via the keypad for execution. A LCD is used to display the data keyed in. Variety of load profiles can be generated with the different combination of 15 resistors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

This research seeks to investigate the research challenges associated with the load emulator. This project is to design a flexible three phase load emulator. The main component of load emulation is the algorithm that generates the desired currents to be drawn so as to mimic the actual load. The load emulator may consist of controller, which allows user to fire number of load resistors, solid-state relays and set of resistors.

Using load emulation, the feasibility of connecting a particular machine to an AC grid under various load conditions can be examined without the need for any electromechanical machinery.

The load emulator is composed of switching components, which can be use in producing different load characteristics. This emulator produces a programmable load or source currents. Using load emulation, the feasibility of connecting a particular machine to a grid under various load conditions can be examined. The emulator requires flexible and powerful hardware structures in order to provide switching load characteristics.

1.2 Aims and Objectives

The primary objective is to develope a flexible three phase load emulator power interface for the actual electrical load. Therefore, the behaviour of the load emulator must be close to the actual load when connected to a power supply.

The work described in this thesis focuses on the experimental development and evaluation of this load emulator connected with the power grid systems that behave as the real load. Using load emulation, the feasibility of connecting a particular load to a grid under various conditions can be studied in the absence of any electromechanical machinery.

The load emulator may consist of controller, which allows user to fire number set of load resistors, solid-state relays and set of resistors. For each phase, there will be five set of power resistors connect in parallel and those resistors are control using solid state realys. A load emulator produce different load characteristics can be studied.

1.3 Procedure of Project Development

Load emulator is a project of never investment by senior before. Therefore, this new title of project all is develops form zero. The procedure to develop from beginning to final is important. Follow the procedure to develop the project with step by step can avoid developer away from the point to achieve the goal.

The first step of load emulator development procedure is research the relevant information about switches type in the market, driver for switch and pros and cons of different switch type. The difficult part is to find a switch that control able using low voltage 5-15V DC to turn on a 240AC-415AC resistive load.

The second development procedure step is blueprint design of ‘load emulator’’. After that, electronic part implement will follow with the blueprint design and objective. Under electronic development, software implement and hardware implement is simultaneity. Simulation is necessary perform before practice on the breadboard. The result of circuit design or modify can checking by simulation before construct can avoid component damage because mistake.

When the circuit is successful in simulation, circuit testing on the breadboard to confirm the circuit can be use by practice. If breadboard testing is successful, PCB fabrication implement following. After PCB is successful function, implement process of integration electronic hardware to testing

Troubleshooting and adjustment on electronic part and programming part until load emulator can function well. The procedure development block diagram is providing on Figure 1.1.

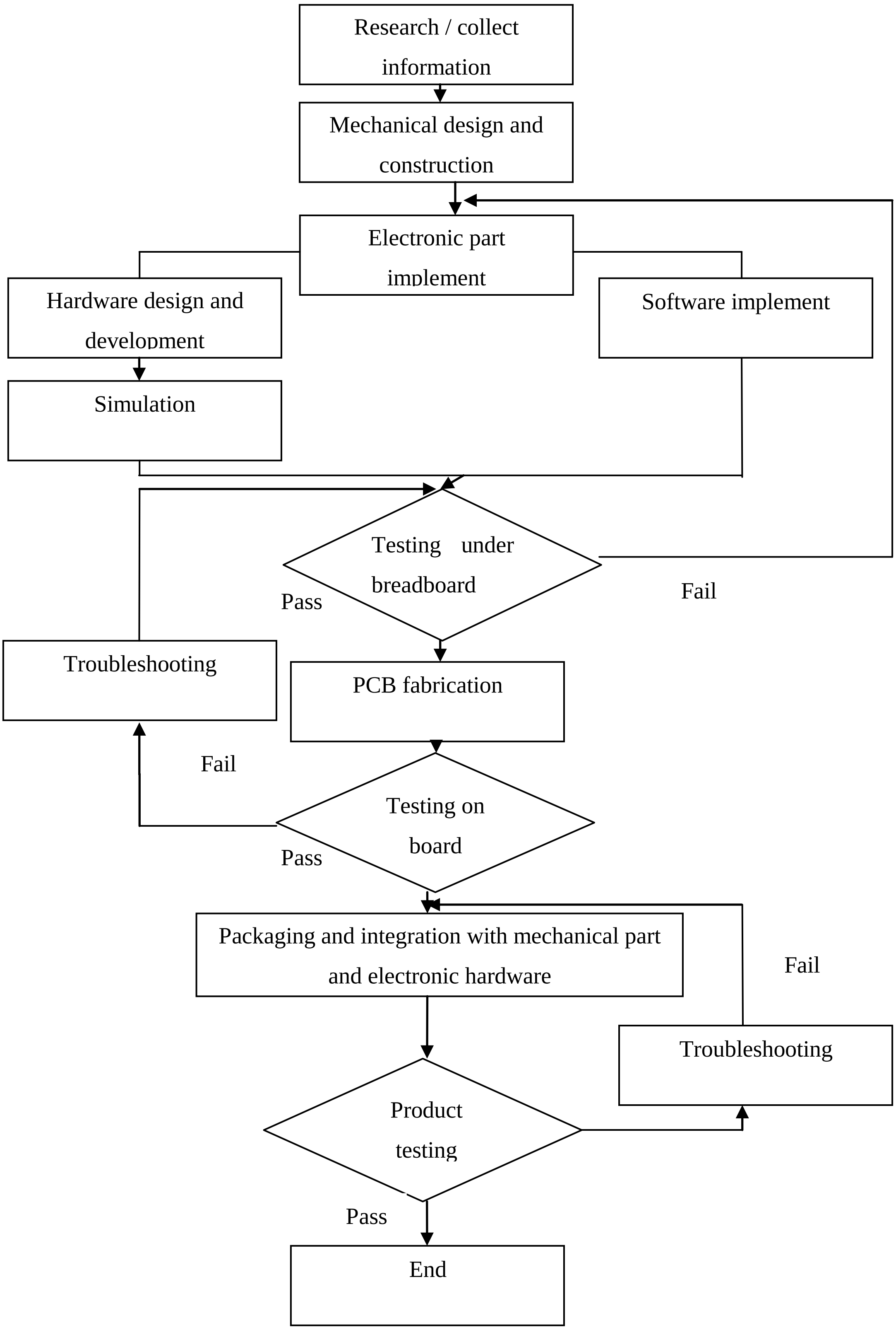

Figure 1.1: Block Diagram of Procedure Development

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This research seeks to investigate the research challenges associated with the load emulator. This project is to design a flexible three phase load emulator. Components that use to develop load emulator are microcontroller PIC, solid state relays, power resistors, LCD, keypad, switches. The PIC is use to interface with the LCD and also the keypad so as to give the signal for switching. User of this load emulator is able to control the resistive load by following the instruction of the displayed screen on LCD and key in value. The instruction may consist of key in set of resistive load for line phase of the three phase system. The solid state relay is act as a switch to turn on/off the HV load side. The solid state relay is control able by using a DC 3V-5V input produce from PIC signal during firing. Software toold is used to design and develop schematic circuit and PCB board.

2.2 Selection of Power Switch

There are many types of power switch that can be use as power switch such as MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor), Power Transformer, Silicon-Controlled Rectifier (SCR)**,** and GTO (Gate Turn-off Thyristors) and solid state relay (SSR). The selection of power switch is important including the consideration of current and voltage rating. This means that whether the power switch can sustain up to a certain level of current and voltage rating and will not easily breakdown.

2.2.1 MOSFET

The metal–oxide–semiconductor field-effect transistor (MOSFET, MOS-FET, or MOSFET) is a device used for amplifying or switching electronic signals. Unlike bipolar transistors that are basically current-driven devices, MOSFETs are voltage-controlled power devices.

In MOSFETs, a voltage on the oxide-insulated gate electrode can induce a conducting channel between the two other contacts called source and drain. The channel can be of n-type or p-type and is accordingly called an nMOSFET or a pMOSFET. The schematic diagram of MOSFET is shown below.

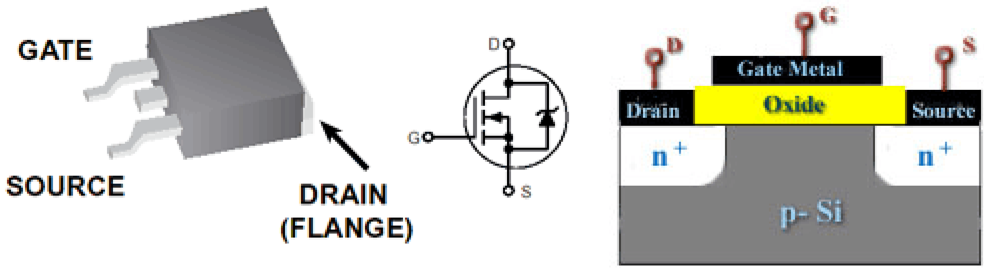

Figure 2.1: Schematic Diagram of MOSFET

To use the power MOSFET as a switch, some of the considerations have to be taken. Because of the extremely high input or Gate resistance that the MOSFET has, it's very fast switching speeds and the ease at which they can be driven makes them ideal to interface with op-amps or standard logic gates. However, care must be taken to ensure that the gate-source input voltage is correctly chosen because when using the MOSFET as a switch the device must obtain a low $R_{DS(on)}$ channel resistance in proportion to this input gate voltage. For example, do not apply a 12v signal if a 5v signal voltage is required. Power MOSFET´s can be used to control the movement of DC motors or brushless stepper motors directly from computer logic or Pulse-width Modulation (PWM) type controllers. As a DC motor offers high starting torque and which is also proportional to the armature current, MOSFET switches along with a PWM can be used as a very good speed controller that would provide smooth and quiet motor operation. The circuits diagram below show the simple power MOSFET motor controller.

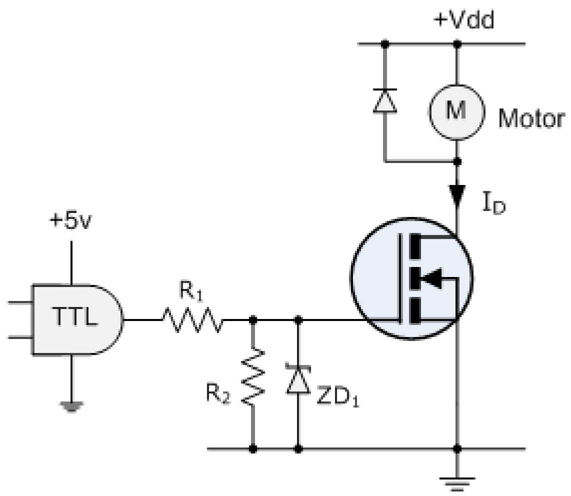

Figure 2.2: Simple Power MOSFET Motor Controller

The operation principle is that as the motor load is inductive, a simple "Free-wheeling" diode is connected across the load to dissipate any back emf generated by the motor when the MOSFET turns it "OFF". The Zener diode is used to prevent excessive gate-source input voltages.

2.2.2 GTO

A gate turn-off thyristor (GTO) is a special type of thyristor, a high-power semiconductor device. GTOs, as opposed to normal thyristors, are fully controllable switches which can be turned on and off by their third lead, the GATE lead. Gate turn-off switch (GTO) is, like an SCR, is a four layer, three junction semiconductor device with three external terminals, namely, the anode, the cathode and the gate, as illustrated in figure. The basic construction, schematic symbol of a GTO are shown in figure below.

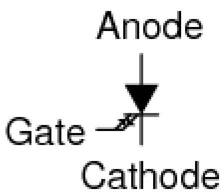

Figure 2.3 : GTO Switch Symbol

The GTO can be turned-on by a gate signal, and can also be turned-off by a gate signal of negative polarity. Turn on is accomplished by a "positive current" pulse between the gate and cathode terminals. As the gate-cathode behaves like PN junction, there will be some relatively small voltage between the terminals. The turn on phenomenon in GTO is however, not as reliable as an SCR (thyristor) and small positive gate current must be maintained even after turn on to improve reliability. Turn off is accomplished by a "negative voltage" pulse between the gate and cathode terminals. Some of the forward current (about one-third to one-fifth) is "stolen" and used to induce a cathode-gate voltage which in turn induces the forward current to fall and the GTO will switch off (transitioning to the 'blocking' state). GTO thyristors suffer from long switch off times, whereby after the forward current falls, there is a long tail time where residual current continues to flow until all remaining charge from the device is taken away. This restricts the maximum switching frequency to approx1kHz.

2.2.3 Silicon-Controlled Rectifier (SCR)

A silicon-controlled rectifier (or semiconductor-controlled rectifier) is a four-layer solid state device that controls current. In the normal “off” state, the device restricts current to the leakage current. When the gate-to-cathode voltage exceeds a certain threshold, the device turns “on” and conducts current. The device will remain in the “on” state even after gate current is removed so long as current through the device remains above the holding current. Once current falls below the holding current for an appropriate period of time, the device will switch “off”. If the gate is pulsed and the current through the device is below the holding current, the device will remain in the “off” state.

If the applied voltage increases rapidly enough, capacitive coupling may induce enough charge into the gate to trigger the device into the “on” state; this is referred to as “dv/dt triggering.” This is usually prevented by limiting the rate of voltage rise across the device, perhaps by using a snubber. “dv/dt triggering” may not switch the SCR into full conduction rapidly and the partially-triggered SCR may dissipate more power than is usual, possibly harming the device.

SCRs can also be triggered by increasing the forward voltage beyond their rated breakdown voltage (also called as break over voltage), but again, this does not rapidly switch the entire device into conduction and so may be harmful so this mode of operation is also usually avoided. Also, the actual breakdown voltage may be substantially higher than the rated breakdown voltage, so the exact trigger point will vary from device to device.

2.2.4 Solid State Relay (SSR)

A solid state relay (SSR) is an electronic switching device in which a small control signal controls a larger load current or voltage. It comprises a voltage or current sensor which responds to an appropriate input (control signal), a solid-state electronic switching device of some kind which switches power to the load circuitry either on or off, and some coupling mechanism to enable the control signal to activate this switch without mechanical parts. The relay may be designed to switch either AC or DC to the load. It serves the same function as an electromechanical relay, but has no moving parts. The control signal must be coupled to the controlled circuit in a way which isolates the two circuits electrically. The control voltage energizes an LED which illuminates and switches on a photo-sensitive diode (photo-voltaic); the diode current turns on a back to back MOSFET to switch the load. The optical coupling allows the control circuit to be electrically isolated from the load. This means that a single MOSFET cannot block current in both directions. For AC (bi-directional) operation two MOSFETs are arranged back to back with their source pins tied together.

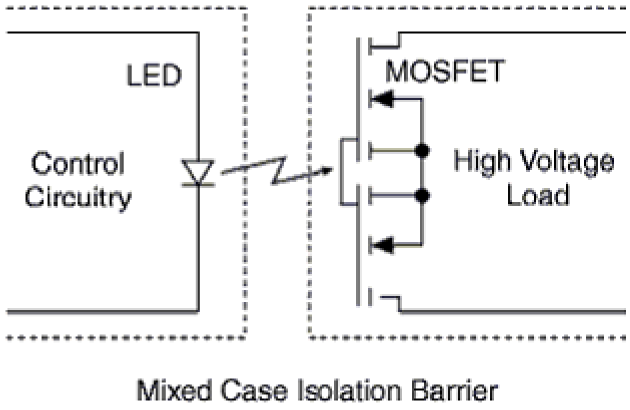

Figure 2.4: Solid State Relay

2.2.5 Comparison Of Switching Devices

Although thyristors are heavily used in megawatt scale rectification of AC to DC, in low and medium power (from few tens of watts to few tens of kilowatts) they have almost been replaced by other devices with superior switching characteristics like MOSFETs. One major problem associated with SCRs is that they are not fully controllable switches. The GTO (Gate Turn-off Thyristor) is related devices which address this problem. In high-frequency applications, thyristors are poor candidates due to large switching times arising from bipolar conduction. MOSFETs, on the other hand, have much faster switching capability because of their unipolar conduction (only majority carriers carry the current).

Solid state relay has the characteristic of MOSFET and provide isolation input control signal and high voltage at output load side. SSR serves the same function as an electromechanical relay, but has no moving parts. SSR is easy to use and it provides safety and protection for the input control side. Hence, the SSR is selected as switching devices in this project.

2.3 Microcontroller PIC 16F877A

PIC16F877A is the chosen one due to high-performance RISC CPU only 35 single-word instructions to learn. All single-cycle instructions except for program branches, which are two-cycle. Operating speed: DC – 20 MHz clock input DC – 200 ns instruction cycle. Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
It has special microcontroller features 100,000 erase/write cycle Enhanced Flash program memory typical and 1,000,000 erase/write cycle Data EEPROM memory typical. It is low-power, high-speed flash/EEPROM technology and Wide operating voltage range (2.0V to 5.5V). Besides that, it can be use in Commercial and Industrial temperature ranges and low-power consumption

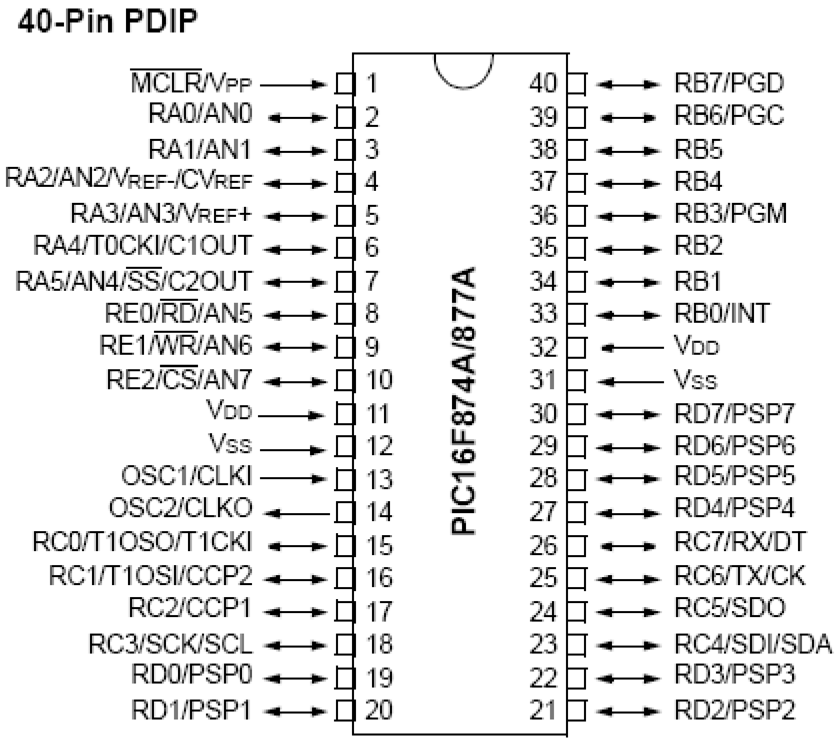

Figure 2.5 : PIC16F877A Pin Diagram

Table 2.1: Key features for PIC16F877A

Key Features	PIC16F877A
Operating Frequency	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	8K
Data Memory (bytes)	368
EEPROM Data Memory (bytes)	256
Interrupts	15
I/O Ports	Ports A, B, C, D, E
Timers	3
Capture/Compare/PWM modules	2
Serial Communications	MSSP, USART
Parallel Communications	PSP
10-bit Analog-to-Digital Module	8 input channels
Analog Comparators	2
Instruction Set	35 Instructions
Packages	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

2.4 Interfacing PIC16F877A with Hitachi 44780-Based LCD

PIC16F877A is used here to display message on the Hitachi HD44780-based character LCD module. PIC16F877A is 8-bit microcontroller based on reduced instruction set computer (RISC) architecture. It has 8kx14-bits flash program memory, 368 bytes of RAM. Here PIC16F877A microcontroller is connected to HD44780 LCD in 4-bit interface data, only four bus lines (DB4 to DB7) are used for data transfer. Bus lines DB0 to DB3 are having no connection with microcontroller. The data transfer between the HD44780U and the PIC16F877A is completed after the 4-bit data has been transferred twice. The most common used for the 44780-based LCD is 16 pin in a row.

Table 2.2: Hitachi 44780 Based LCD Pinout

Table 2.2 Hitachi 44780 Based LCD Pinout	
Pin	Description
1 Vss	Ground
2 Vcc	+5V supply
3 Vee	Contrast Voltage
4 R/S	Instruction / Data Mode select
5 R/W	Read / write
6 E	Enable
7-14 D0-D7	Data lines
15	+Led
16	-Led

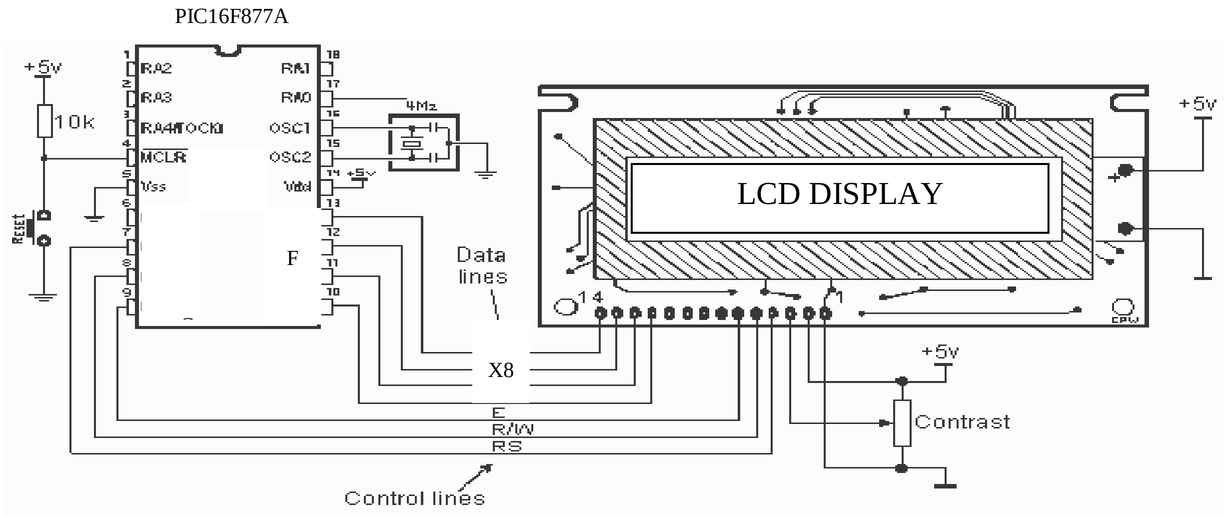

Figure 2.6: Interfacing PIC With LCD

Table 2.3 : 3 "control" lines of LCD

Enable (E)	This line allows access to the display through R/W and RS lines. When this line is low, the LCD is disabled and ignores signals from R/W and RS. When (E) line is high, the LCD checks the state of the two control lines and responds accordingly.
Read/Write (R/W)	This line determines the direction of data between the LCD and microcontroller. When it is low, data is written to the LCD. When it is high, data is read from the LCD.

Register select (RS)	With the help of this line, the LCD interprets the type of data on data lines. When it is low, an instruction is being written to the LCD. When it is high, a character is being written to the LCD.

R/S	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Instruction/Description
4	5	14	13	12	11	10	9	8	7	Pins
0	0	0	0	0	0	0	0	0	1	Clear Display
0	0	0	0	0	0	0	0	1	*	Return Cursor and LCD to Home Position
0	0	0	0	0	0	0	1	ID	S	Set Cursor Move Direction
0	0	0	0	0	0	1	D	C	B	Enable Display/Cursor
0	0	0	0	0	1	SC	RL	*	*	Move Cursor/Shift Display
0	0	0	0	1	DL	N	F	*	*	Set Interface Length
0	0	0	1	A	A	A	A	A	A	Move Cursor into CGRAM
0	0	1	A	A	A	A	A	A	A	Move Cursor to Display
0	1	BF	*	*	*	*	*	*	*	Poll the "Busy Flag"
1	0	D	D	D	D	D	D	D	D	Write a Character to the Display at the Current Cursor Position
1	1	D	D	D	D	D	D	D	D	Read the Character on the Display at the Current Cursor Position

Figure 2.7: The instructions available for 44780 Hitachi LCD

Table 2.4: The initialization steps of LCD

1	Wait for about 45ms delay after power is applied
2	Clear RS and R/W to select instruction mode and writing mode
3	Write the instruction set to set the interface to 8bit, 2 line and 5x7 dot i.e. write b'00111011' to LCD.
4	Set the Enable line then wait for 5ms for the instruction to complete.
5	Write the same data, b00111011 to LCD and wait >160us for instruction to complete
6	Write the same data again, b00111011 to LCD and wait >160us for instruction to complete
7	Set the operating Characteristics of the LCD - Write 0x08 to turn off the display, E = 1 for 3us - Delay 50us - Write 0x01 to clear the display, E = 1 for 3us - Delay 1.8ms - Write 0x02 for cursor home, set E = 1 for 3us - Delay 1.8ms

	- Write 0x0F to turn on the display, set E = 1 for 3us - Delay 50us - Write 0x06 for entry mode set, set E = 1 for 3us - Delay 50us
8	Set RS to 0, and R/W to 0

2.5 Keypad

The 4X4 keypad is consisting of a matrix of switch. This means that it have 16 buttons. With the LCD display, it can rapidly emulate the function of a calculator by setting the suitable mode. The user-defined input function allows the user to set and detect different keys through the software to activate the module for versatile operations. The diagram of 4X4 keypad is shown below.

Figure 2.8: Diagram of 4X4 Keypad

The following is about the connection of the 4x4 Key Matrix to PORTC of the PIC16F877A. The four columns are connected to RC0-RC3 and the four rows are connected to RC4- RC7. The keypad is sampled every 5 ms. The seven segment is connected to PORTB. The schematic diagram of keypad connection is shown below.

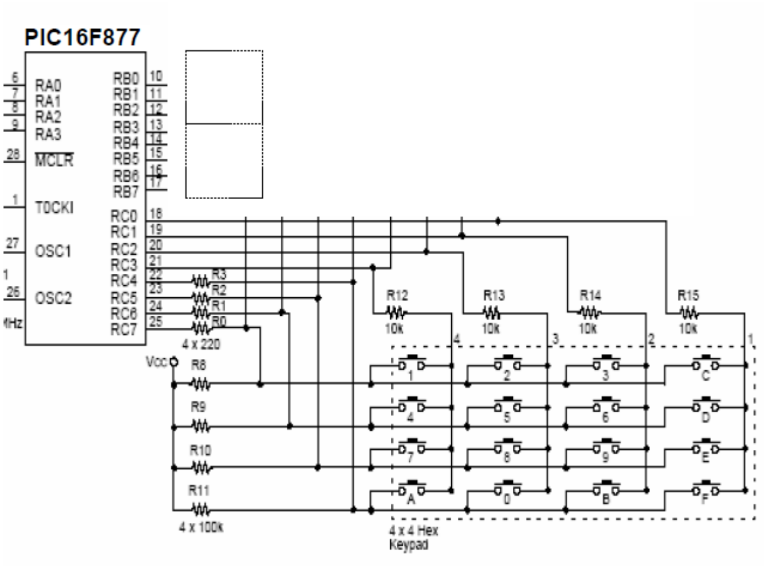

Figure 2.9: Keypad Schematic Diagram

2.6 MPLAB Software

MPLAB software is used to develope code for PIC16F877A to interact with peripheral devices such as keypad, LCD screen and SSR. The program code consists of defining I/0 ports and pins for PIC16F877A.

MPLAB IDE is a software program that runs on a PC to develop applications for Microchip microcontrollers. It is called an Integrated Development Environment, or IDE, because it provides a single integrated environment to develop code for embedded microcontrollers.

An embedded system is typically a design making use of the power of a small micro-controller, like the Microchip PICmicroÆ MCU or dsPICÆ Digital Signal Controller (DSCs). These microcontrollers combine a microprocessor unit (like the CPU in a desk-top PC) with some additional circuits called peripherals, plus some additional circuits on the same chip to make a small control module requiring few other external devices.

2.7 Real PIC Simulator Software

Real Pic Simulator is a Microchip PIC microcontroller simulator capable of real-time simulation. An integrated disassemble allows examining and exporting the code to assembler code. Debugger allows execution of the program in real-time, at selected speed or step-by-step, using breakpoints. RAM and EEPROM viewer allows the user to inspect RAM and EEPROM memory content. Processor viewer allows users to view the microcontroller's pin allocation and characteristics. Visual simulator enables visual simulation of the program with visual components (LEDs and Keypads).

2.8 Dip Trace Software

This software is to design a schematic, simply select and place components onto your document and connect them together using the wire and bus tools. Multisheet and hierarchical schematics are supported. Then select the menu option 'Convert to PCB' to convert the schematic to PCB. Layout can be updated from Schematic in a few clicks at anytime. DipTrace PCB software includes 2 automatic routers (Shape-based and Grid-based). Grid Router can also make single-layer boards with jumper wires. With Specctra DSN/SES interface you can use external shape-based or topological autorouter. Intelligent manual routing tools allow you to create and edit traces by 90, 45 degree or without any limitations. Curved traces are supported. Through, blind or buried vias can be used in automatic and manual routing. Board size is not limited.

Powerful copper pour system can help to reduce your manufacturing costs by minimizing the amount of etching solution required. To use it, all you have to do is insert a copper area on your board in the PCB Layout program and any pad or trace inside the selected area will be automatically surrounded with a gap of the desired size. Using copper pour you can also create planes and connect them to pads and vias (different thermal types are supported).

2.9 PSCAD

It allows the user to efficiently construct a circuit schematic, run a simulation, analyze the results, and manage the data in a completely integrated graphical environment.

PSCAD is fast, accurate and easy-to-use power system simulation software for the design and verification of all types of power systems. PSCAD is most suitable for simulating time domain instantaneous responses, also known as electromagnetic transients or instantaneous solutions, in both electrical and control systems.

PSCAD becomes an indispensable tool for a variety of power system designs and studies. It is a multi-purpose tool. It is equally capable in the areas of power electronic design and simulation, power quality analysis, protection and electrical utility system planning studies. It is easier and much less expensive to design and optimize electrical devices and systems prior to prototyping or manufacturing. Thus, PSCAD is becoming a true Power System Computer-Aided Design tool for a variety of industry applications. PSCAD users include engineers and technologists from energy utilities, electrical equipment manufacturers, engineering consulting firms, and research and academic institutions. PSCAD is used in the planning, design, and operational phases of power systems. It is also very prevalent in power system research around the world.

CHAPTER 3

METHODOLOGY

3.1 Introduction of Load Emulator

Figure 3.1 shows the block diagram of load emulator. The load emulator is equipped with PIC 16F877A, LCD Hitachi 44780, solid state relays D4825x15, keypad 4x4, switches x15 and power resistor(288Ω)x15 and ac to dc inverter.

Switching power supply and AC adapter are using together in power supply unit. In additional, two steps down voltage circuit are built on power supply part by using component voltage regulator 5V from 9V DC switching power supply. Lastly, solid state relays play a role to active the resistive load from order of control unit. The hardware block diagram is provided on Figure 3.1.

For this load emulator there are two ways to control the solid state relays one is manual switching another is auto switching. Therefore, user is able to choose switching type of resistive loads. For each phase, there will be five load resistors, these five load resistors are connected to those solid-state relays. For manual switching will be manual switches. For auto switching there will be PIC and keypad for user to key in number of resistors for each phase. The C programming develops using MPLAB software is use for PIC16F877A interacts with peripheral devices such as keypad, LCD screen and solid state relays. The program code consists of defining I/0 ports and pins for PIC16F877A. For this load emulator user is able to select number of resistive load for each of the phase voltage. Figure 3.1 shows diagram of load emulator.

The primary objective of load emulation is to design the power interface for the actual electrical load. To allow this user of this load emulator are able to control the resistive load by following the instruction of the displayed screen on LCD and key in number of resistive load. The instruction may consist number of resistive load for each phase that wanted to apply into the system.

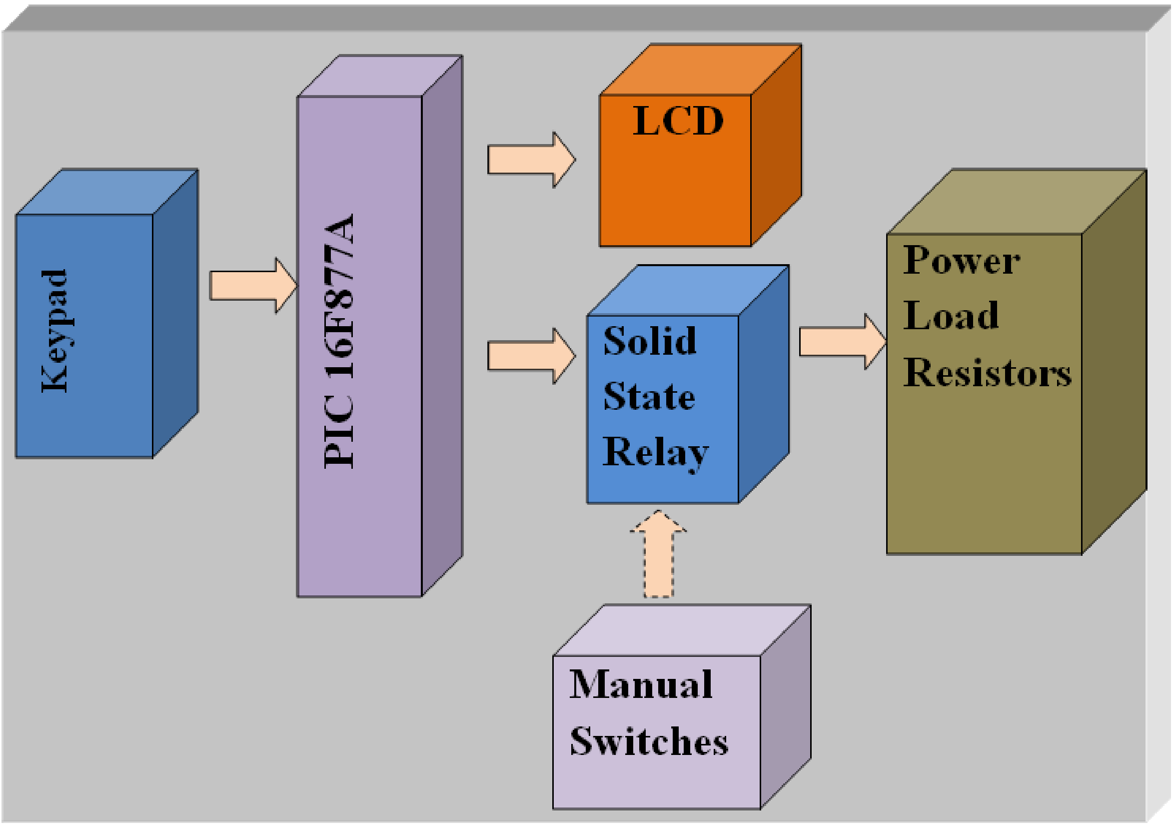

Figure 3.1: Block Diagram of Load Emulator

3.2 Selection Relay

Item that has to consider in this project is switch or relay and protection circuit because the specification of it must be able to turn on using 3-9V dc and the output of it must be able to sustain 240-415ac. Moreover the input and output of the switch must be isolated to prevent reverse current that will spoil sensitive devices and to protect human from getting electrical shock. The project is dealing with 240-415ac if there is no protection circuit that will be dangerous.

The most suitable relay is Crydom solid state relayD4825-10.

Figure 3.2: D4825-10 Solid State Relay

The control signal must be coupled to the controlled circuit in a way which isolates the two circuits electrically. The control voltage energizes an LED which illuminates and switches on a photo-sensitive diode (photo-voltaic); the diode current turns on a back to back MOSFET to switch the load. The optical coupling allows the control circuit to be electrically isolated from the load. This means that a single MOSFET cannot block current in both directions. For AC (bi-directional) operation two MOSFETs are arranged back to back with their source pins tied together.

3.3 Select Resistor Value

By using voltage value and power value substitutes to $Power(w) = \frac{V^2}{R}$, therefore resistor value can be obtain. Table 3.1 shows the resistor values have been obtained. For this project 15 pieces of 288ohm resistors are use as load profile.

Table 3.1: Obtained Resistor Value

Voltage	Power		
	200	250	300
	Resistor Value		
200	200	160	133.33
210	220.5	176.4	147
220	242	193.6	161.33
230	264.5	211.6	176.33
240	288	230.4	192

3.4 Power Supply for The Circuit

User can choose either use the AC to DC adaptor or 12V battery to power up the circuit. Higher input voltage will produce more heat at LM7805 voltage regulator. Typical voltage is 12V. Anyhow, LM7805 will still generate some heat at 12V. Normally AC to DC adaptor can be plugged to J1 type connector. Shown in Figure 3.3, the D1 is use to protect the circuit from wrong polarity supply. C1 and C3 is use to stabilize the voltage at the input side of the LM7805 voltage regulator, while the C4 and C2 is use to stabilize the voltage at the output side of the LM7805 voltage supply. The single pole double throw switch (spdt) is to allow user to switch manual switch or switch the supply to voltage regulator part as a source to power up PIC. Green LED to indicate the power status of the PIC controller circuit. R6 is resistor to protect LED from over current that will burn the LED.

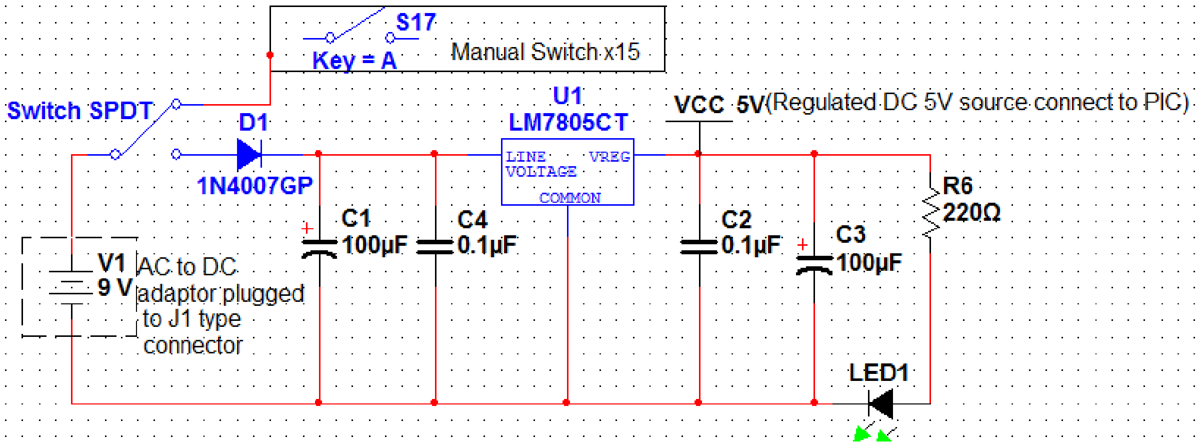

Figure 3.3: Power Supply for The Circuit

3.5 PIC Controller Board

In this load emulator there are two separated boards one is controller board and other is LED indicator board. Table 3.2 shows a microcontroller input pin assignment. Table 3.3 shows microcontroller output pin assignment. Figure 3.5 shows the schematic diagram of microcontroller interacts with keypad and LCD.

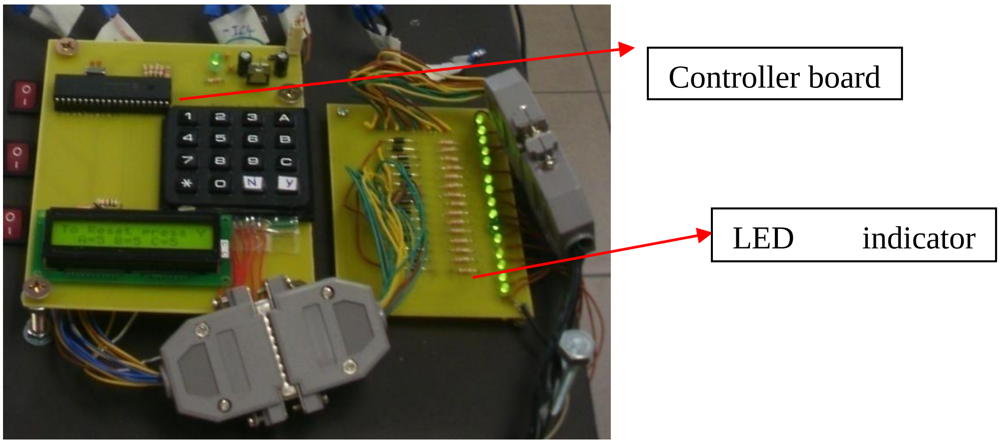

Figure 3.4: Controller and LED Board

Table 3.2: Microcontroller input pin

Input		Connected by
Number of pin	**Pin Description**	
2	RA0	Keypad 4x4 K1(C1)
3	RA1	Keypad 4x4 K2(C2)
4	RA2	Keypad 4x4 K3(C3)
5	RA3	Keypad 4x4 K4(C4)

Table 3.3: Microcontroller Output Pin

Output		Connected by
Number of pin	**Pin description**	
6	RA4	Keypad 4x4 K8(R1)
7	RA5	Keypad 4x4 K7(R2)
8	RE0	Keypad 4x4 K6(R3)
9	RE1	Keypad 4x4 K5(R4)
35	RB2	LCD DB2
36	RB3	LCD DB3
37	RB4	LCD DB4
38	RB5	LCD DB5
39	RB6	LCD DB6
40	RB7	LCD DB7
10	RE2	Phase A1
17	RC2	Phase A2
18	RC3	Phase A3
19	RD0	Phase A4
20	RD1	Phase A5
30	RD7	Phase B1
29	RD6	Phase B2
28	RD5	Phase B3

27	RD4	Phase B4
26	RC7	Phase B5
25	RC6	Phase C1
24	RC5	Phase C2
23	RC4	Phase C3
22	RD3	Phase C4
21	RD2	Phase C5

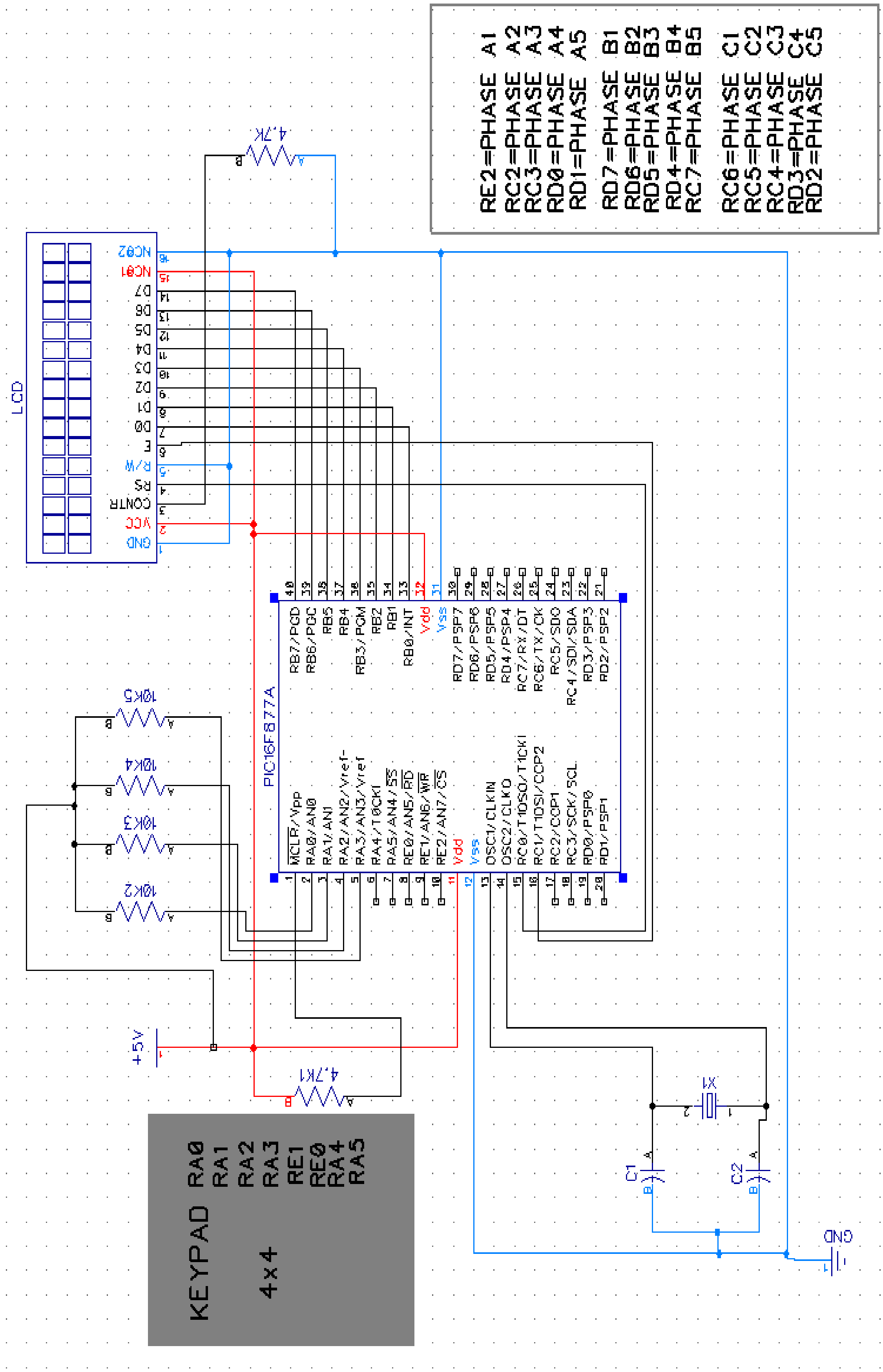

Figure 3.5: Schematic Diagram For The PIC Controller

3.6 Interface PIC16F877A with LCD (2 X 16 characters)

Table 3.4 shows LCD pin connected with microcontroller. Figure 3.6 shows that name of each LCD pin.

Table 3.4: LCD Connection With PIC

Pin	Name	Pin Function	Connection
1	VSS	Ground	GND
2	VCC	Positive supply for LCD	5V
3	VEE	Brightness adjust	Connect to 4.7k ohm resistor
4	RS	Select register, select instruction or data register	RC0
5	R/W	Select read or write	GND
6	E	Start data read or write	RC1
7	DB0	Data bus pin	RB0
8	DB1	Data bus pin	RB1
9	DB2	Data bus pin	RB2
10	DB3	Data bus pin	RB3
11	DB4	Data bus pin	RB4
12	DB5	Data bus pin	RB5
13	DB6	Data bus pin	RB6

14	DB7	Data bus pin	RB7
15	LED+	Backlight positive input	5V
16	LED-	Backlight negative input	GND

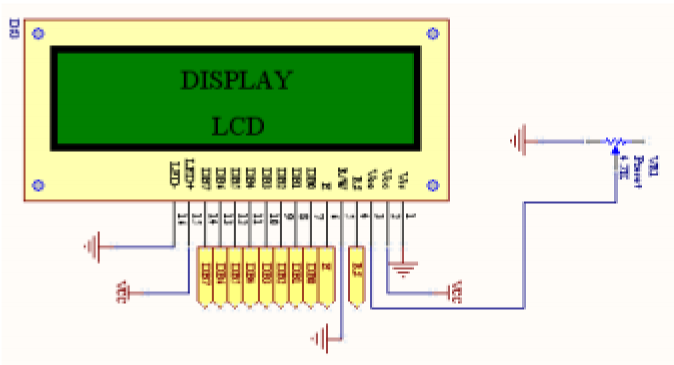

Figure 3.6: 2X 16 characters LCD

3.7 PCB Layout For PIC Controller

For this project Diptrace software is used to draw out PCB board. Drawing in Diptrace schematic is able to convert to PCB layout which can save a lot of time. From the Diptrace software has a function call auto arrange trace that will able to draw the trace without intercept traces. Diptrace is a strong software tool that easy to use and this is reason Diptrace is the chosen among all PCB layout tools. Figure 3.7 shows the top view of controller PCB Layout. Figure 3.8 shows the bottom view of controller PCB layout.

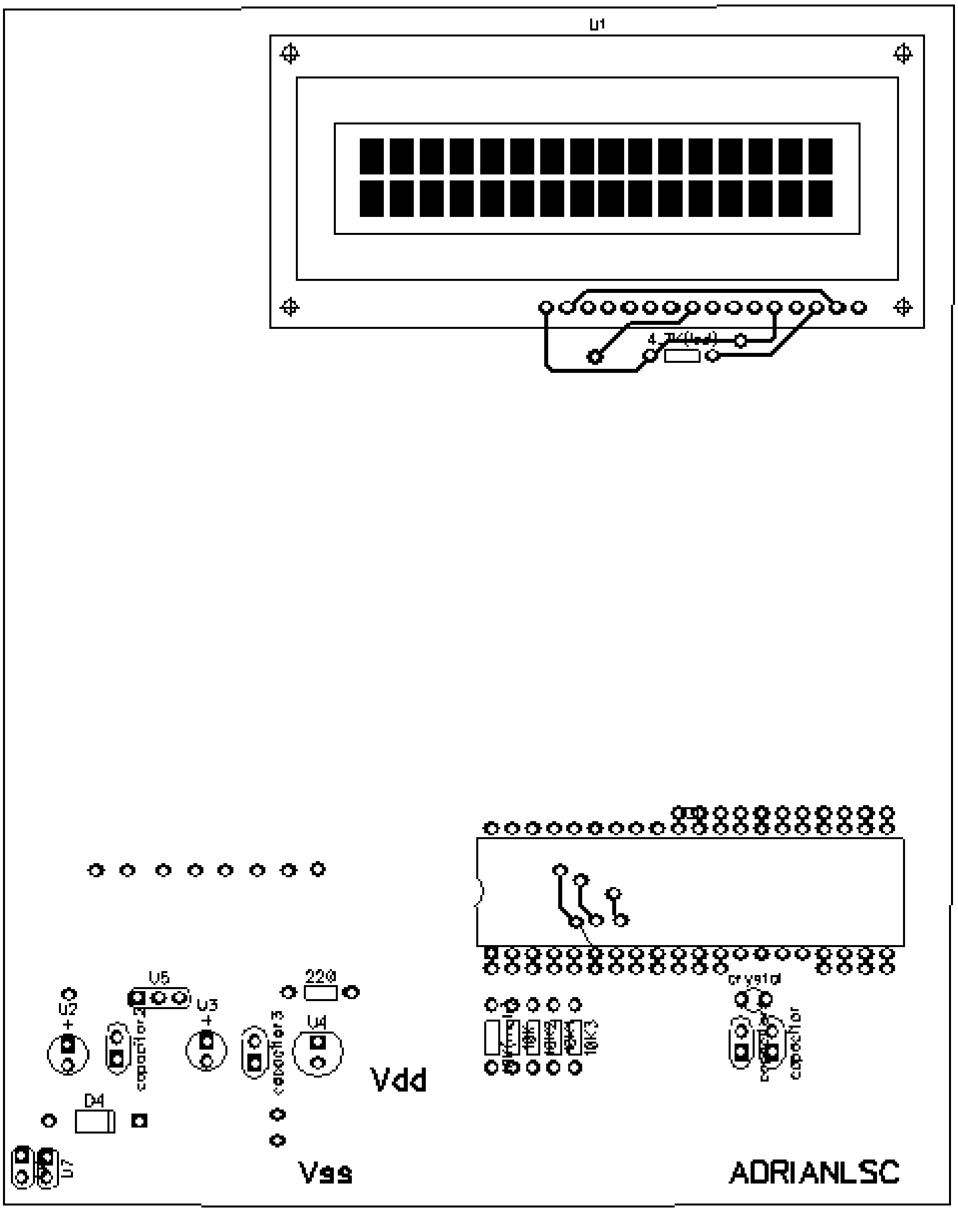

Figure 3.7: Controller Top View PCB Layout

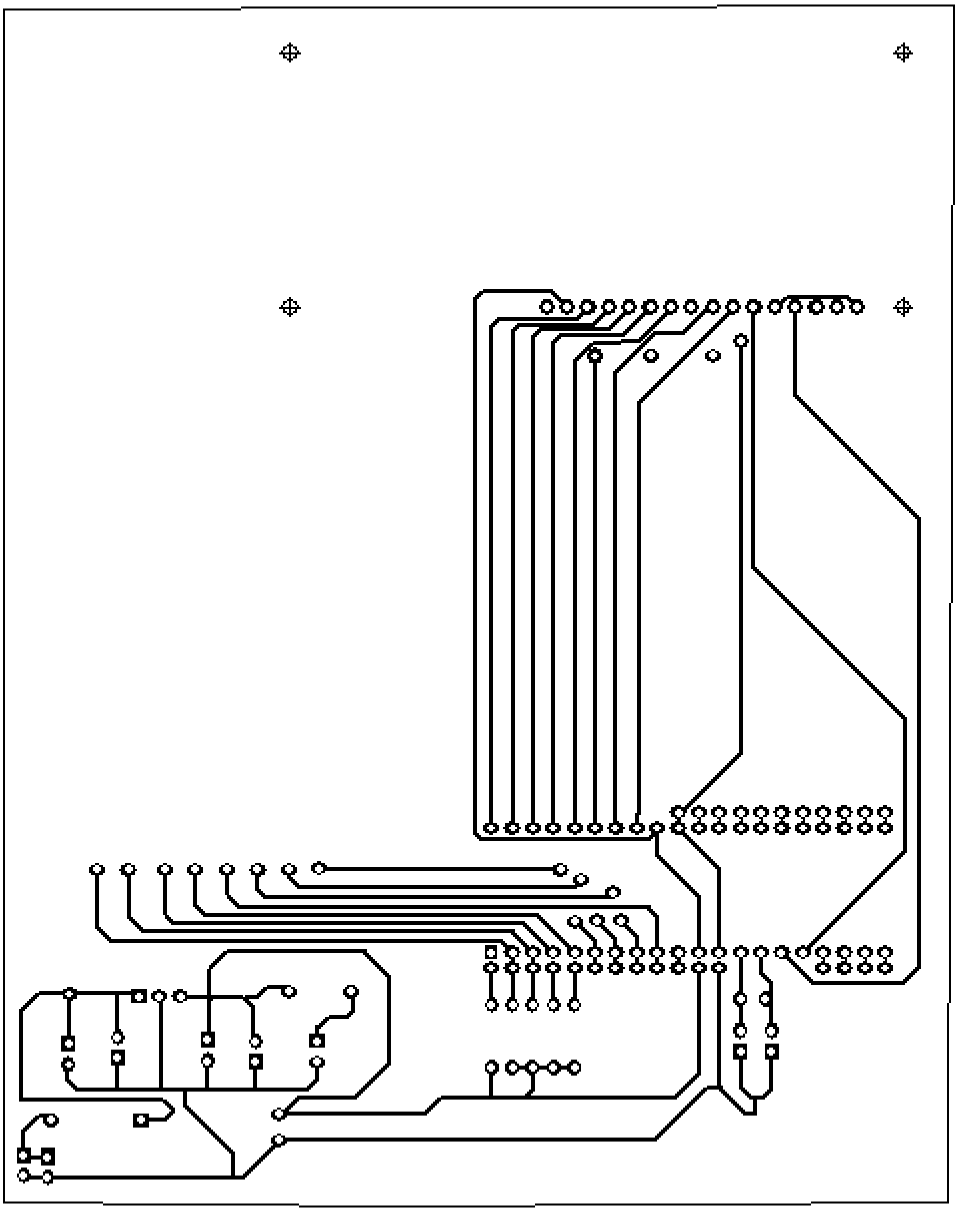

Figure 3. 8: Controller Bottom View PCB layout

3.8 LED Indicator Board

Figure 3.9 shows signal coming from manual switch or controller will turn on the solid state relay. D3 diode shows in Figure 3.9 are to protect wrong polarity supply which helps to prevent damages to PIC controller. Green LED to indicate the power status of the solid state relay. Figure 3.9 shows input signal to solid state relay is able to turn ON/OFF a power load resistor. R21 is use to protect LED from over current that will burn the LED. Figure 4.0 shows top and bottom view of LED indicator PCB board. Figure 3.10 shows the top and bottom view of LED indicator circuit board.

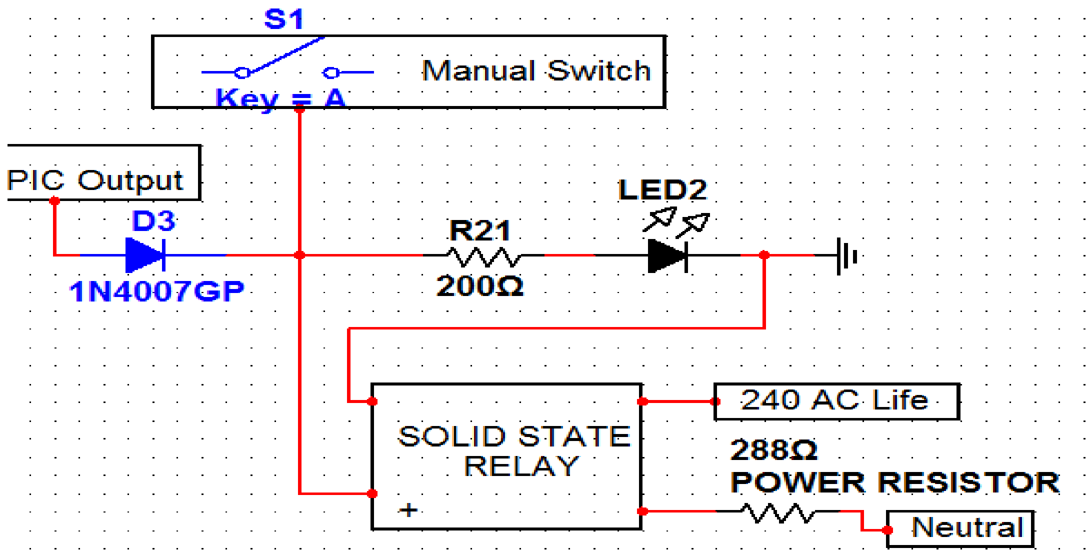

Figure 3.9: LED Indicator Board Connect With Solid State Relay And Power Load Resistor

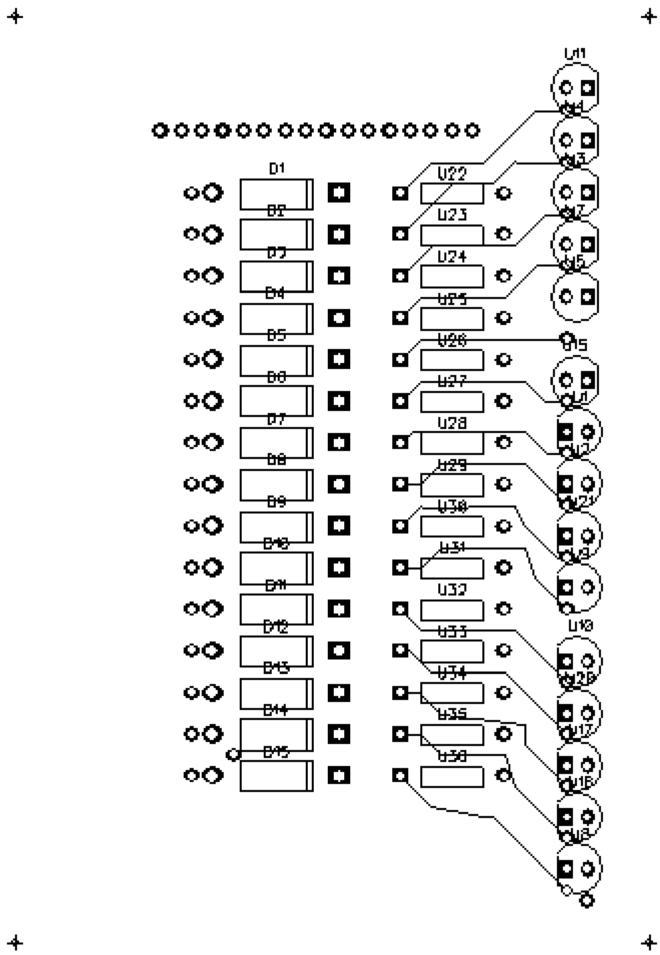

Figure 3.10: Top View Of LED Indicator PCB Board

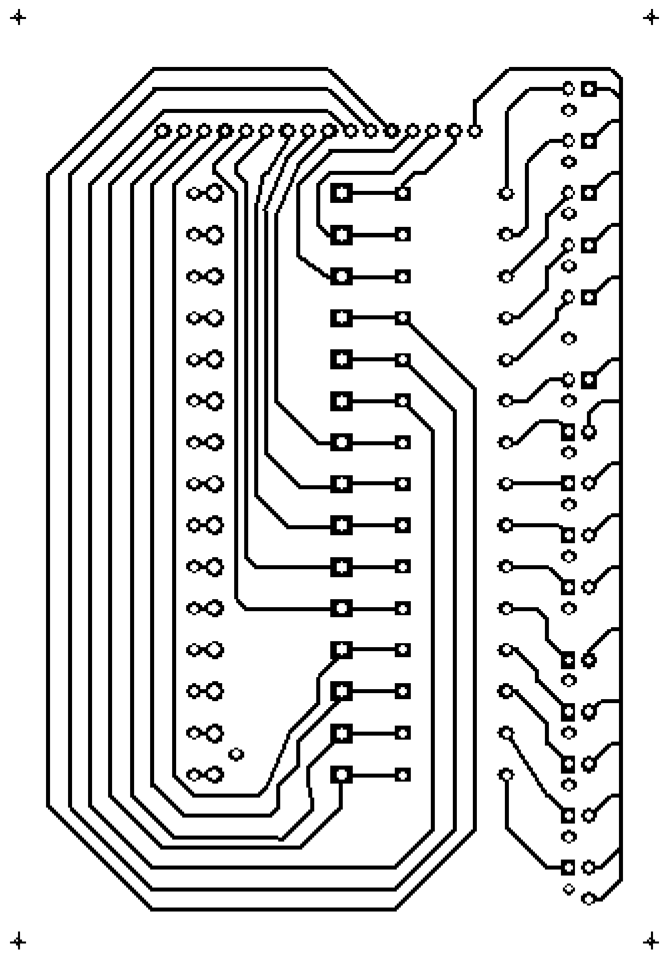

Figure 3.11: Bottom View Of LED Indicator PCB Board

3.9 Microcontroller Program Flow Diagram

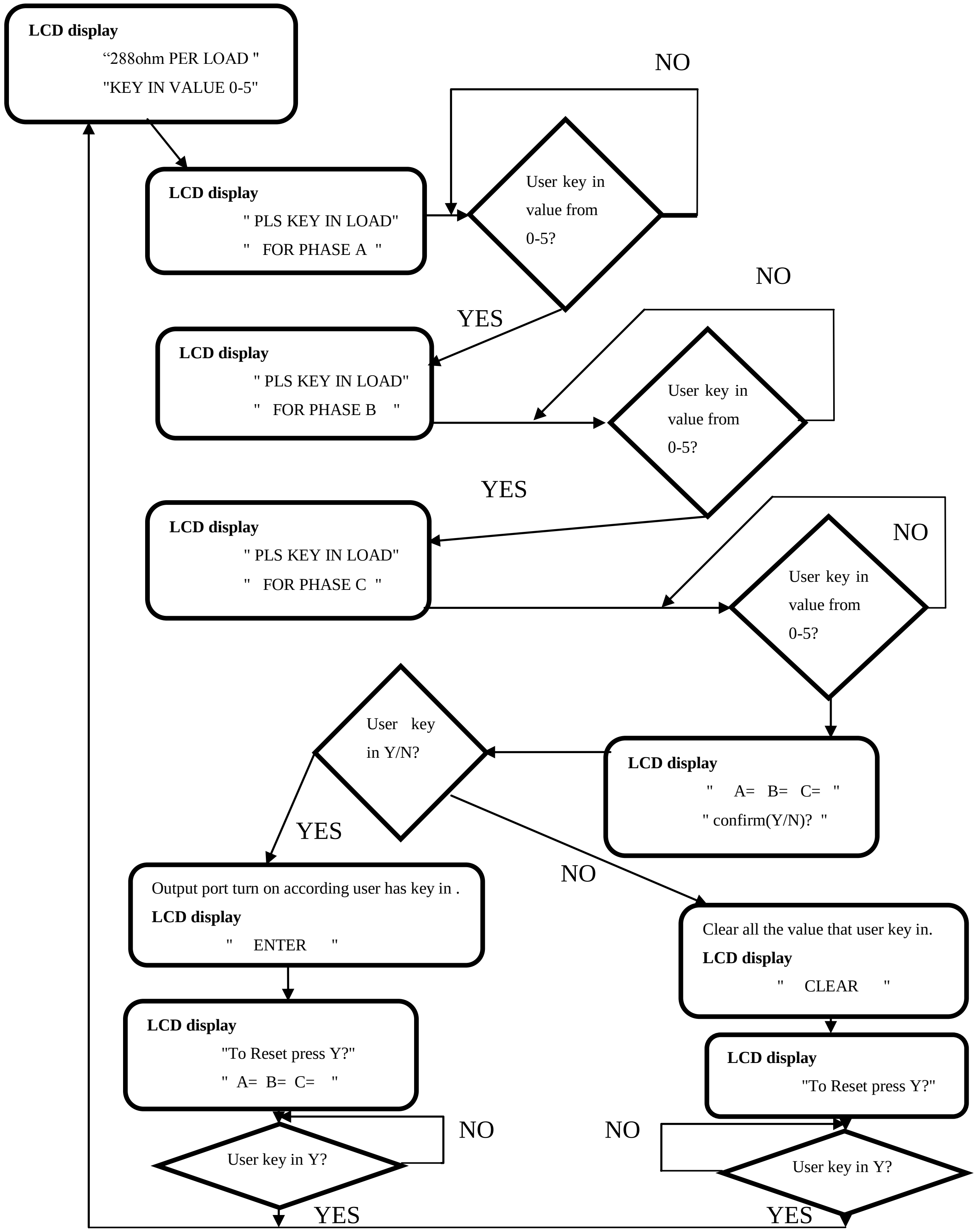

Figure 3. 12: Microcontroller program Flow Chart

3.10 Microcontroller Program Source Code Description

Define port for RS, E and 15 output ports are shown on figure 3.12.

```
//==================================================================
//  include
//==================================================================
#include <pic.h>


//==================================================================
//  configuration
//==================================================================
__CONFIG ( 0x3F32 );

//==================================================================
//  define port
//==================================================================
#define rs          RC0
#define e           RC1
#define lcd_data    PORTB

#define phasea1 RE2
#define phasea2 RC2
#define phasea3 RC3
#define phasea4 RD0
#define phasea5 RD1

#define phaseb1 RD7
#define phaseb2 RD6
#define phaseb3 RD5
#define phaseb4 RD4
#define phaseb5 RC7



#define phasec1 RC6
#define phasec2 RC5
#define phasec3 RC4
#define phasec4 RD3
#define phasec5 RD2
```

Figure 3.13: Source code 1

Function prototype shown on figure 3.13.The Function prototype serves the following purposes:

1) It tells the return type of the data that the function will return.
2) It tells the number of arguments passed to the function.
3) It tells the data types of the each of the passed arguments.
4) Also it tells the order in which the arguments are passed to the function.

```
//==========================================================================================
//  function prototype
//==========================================================================================
void delay(unsigned long data);
void continueloop(void);
void send_config(unsigned char data);
void send_char(unsigned char data);
void e_pulse(void);
void lcd_goto(unsigned char data);
void lcd_clr(void);
void send_string(const char *s);
void clearrow1(void);
void clearrow2(void);
void clearrow3(void);
void clearrow4(void);
void scancolumn1(void);
void scancolumn2(void);
void scancolumn3(void);
void scancolumn4(void);
```

Figure 3.14: Source code 2

Set input and output pins are shown on Figure 3.14.

```
void main(void)
{
    ADCON1=0b00000110;  //set all portA pins as digital I/O
    TRISA=0b11001111;   //clear bit 4&5 portA as output and set the rest as input
    TRISB=0b00000000;   //set portB as output
    TRISD=0b00000000;   //set portD as output
    TRISC=0b00000000;   //set bit4-7 portC as input(connected to 4 row of keypad)
    TRISE=0b00000000;   //set portE as output
```

Figure 3.15: Source code 3

Figure 3.15 shows the configure LCD

```
  send_config(0b00000001);    //clear display at lcd
  send_config(0b00000010);    //Lcd Return to home
  send_config(0b00000110);    //entry mode-cursor increase 1
  send_config(0b00001100);    //diplay on, cursor off and cursor blink off
  send_config(0b00111000);    //function
```

Figure 3.16: Source code 4

Send string to LCD is shown in figure 3.16. LCD first to shows “288ohm PER LOAD” on first line and second line shows “KEY IN VALUE 0-5”.

```
lcd_clr();                          //clear LCD
__delay_ms(300);
lcd_goto(0);
send_string(" 288ohm PER LOAD");    //display " 288ohm PER LOAD"
lcd_goto(20);
send_string("KEY IN VALUE 0-5");    //        "KEY IN VALUE 0-5"
__delay_ms(1200);
```

Figure 3.17: Source code 5

Text string change after 1.2 second and on first line of LCD shows "PLS KEY IN LOAD" and " FOR PHASE A " on second line is shown in figure 3.17.

```
if(count==0)

        {   lcd_clr();
            lcd_goto(0);
            send_string(" PLS KEY IN LOAD");       //display " PLS KEY IN LOAD"
            lcd_goto(20);
            send_string("   FOR PHASE A  ");       //        "   FOR PHASE A  "
            __delay_ms(600);
```

Figure 3.18: Source code 6

Figure 3.18 shows the connection of the 4x4 keypad pins with PIC16F877A. When a key for example '3' is pressed, the 2 pin RA2 and RE1 will be shorted. Thus, to use a keypad without keypad decoder, the eight pins of the keypad will have to be separated into 2 groups (4 pin to input and 4 pin to output of PIC). Refer to Figure 4.8, RA0-RA3 will set as input while RA4, RA5, RE0 and RE1 will set as output. Source code in figure 3.19 shows a simple method to read the keypad. Program will scan row and column of keypad to read 1 digit entered by user (figure 3.20 and 3.19). First, clear the output pin RE1 and set the others. Go to a 'scancolumn1' function shown in figure 3.21. If the RA0 (input) detect a 0, it means the '1' key is pressed. Now, clear the second column which is pin RE0 and set the others. Go to 'scancolumn2' function to scan whether the key '4' or '5' is being pressed. If RA1 (input pin) detects a 0, meaning that the key '5' is pressed. Clear RA5. At last, clear RA4 and go for 'scancolumn4' function. Refer to the scancolumn1 function in figure 3.21 , if RA0 pin equal to '0' ('1' key is pressed), the program under the "if" command will be activated. First, the "while" command will wait the '1' to be released to make sure the program under the "if" command will only run one time for a press. The purpose of the 'count' variable is to let the LCD screen clear when the first digit is entered. The LCD will display the value that user key in to tell user already been entered. This program will not have any operation if user enters wrongly.

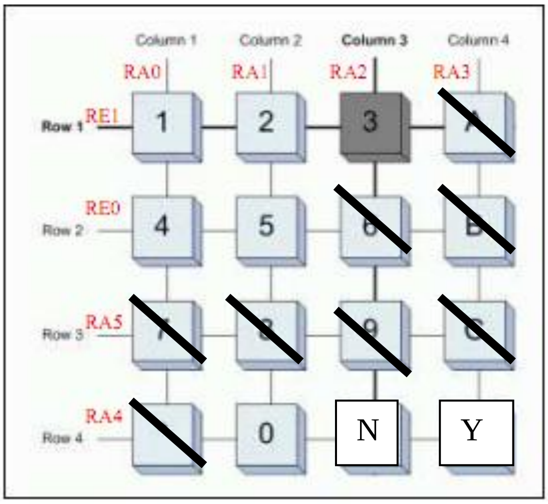

Figure 3.19: Source code 7

```
clearrow1();                        //Clear 1st output pin and set the others
scancolumn1();                      //scan column 1,2,4
clearrow2();                        //Clear 2nd output pin and set the others
scancolumn2();                      //scan column
clearrow4();                        //Clear 4th output pin and set the others
scancolumn4();                      //scan column
```

Figure 3.20: Source code 8

```
//  scanning functions for keypad
//============================================================================

void clearrow1(void)
{
    RE1=0;
    RE0=1;
    RA5=1;
    RA4=1;
}

void clearrow2(void)              //clear the 2nd row and set the others
{
    RE1=1;
    RE0=0;
    RA5=1;
    RA4=1;
}


void clearrow4(void)              //clear the 4th roe and set the others
{
    RE1=1;
    RE0=1;
    RA5=1;
    RA4=0;
}
```

Figure 3.21: Source code 9

```
void scancolumn1(void)
{
    if(RA0==0)                          //if key '1' is being pressed
    {
        while(RA0==0)continue;          //waiting the key to be released
        lcd_clr();                      //Clear the LCD
        lcd_goto(0);                    //The cursor of LCD back to home
        send_char('1');                 //Display the symbol '1' at LCD
        keyin_char[count]='1';          //Stall the '1' value at the keyin_char array
        count+=1;                       //increase the count variable's value by 1 and the result stall back to the variable
        __delay_ms(300);
    }
    else if(RA1==0)                         //if key '2' is being pressed
    {
        while(RA1==0)continue;          //waiting the key to be released
        lcd_clr();                      //Clear the LCD
        lcd_goto(0);                    //The cursor of LCD back to home
        send_char('2');                 //Display the symbol '2' at LCD
        keyin_char[count]='2';          //Stall the '2' value at the keyin_char array
        count+=1;                       //increase the count variable's value by 1 and the result stall back to the variable
        __delay_ms(300);
}
    else if(RA2==0)                     //if key '3' is being pressed
    {
        while(RA2==0)continue;          //waiting the key to be released
        lcd_clr();                      //Clear the LCD
        lcd_goto(0);                    //The cursor of LCD back to home
        send_char('3');                 //Display the symbol '3' at LCD
        keyin_char[count]='3';          //Stall the '3' value at the keyin_char array
        count+=1;                       //increase the count variable's value by 1 and the result stall back to the variable
        __delay_ms(300);
    }
}
```

Figure 3.22: Source code 10

After PIC has detected first value entered then only the program proceed display on first line of LCD shows “PLS KEY IN LOAD” and “ FOR PHASE B ” on second line and same goes for phase C. After ‘count’ equals to three the program proceed to get confirmation from user whether want to fire the load to phase line (figure 3.22). If user press ‘Y’(yes) the program start turn on the particular output which key in by user(figure 3.23). For user entered ‘N’ (no) the program clears all the data that has entered by user (figure 3.24). Lastly microcontroller program will reset and execute all over again if only user press ‘Y’ as reset.

```
else if(count==3)
    {


                lcd_clr();
                lcd_goto(0);
                send_char(' ');                 //display "  A=  B=  C=    "
                send_char(' ');                 //display " confirm(Y/N)?  "
                send_char('A');
                send_char('=');
                send_char(keyin_char[0]);
                send_char(' ');
                send_char('B');
                send_char('=');
                send_char(keyin_char[1]);
                send_char(' ');
                send_char('C');
                send_char('=');
                send_char(keyin_char[2]);
                send_char(' ');
                send_char(' ');
                send_char(' ');

                lcd_goto(20);
                send_string(" confirm(Y/N)?  ");
```

Figure 3.23: Source code 11

```
if((keyin_char[0]=='1'))phasea1=1;

else if((keyin_char[0]=='2'))
{           phasea1=1;
            phasea2=1;
}
else if((keyin_char[0]=='3'))
{   phasea1=1;
    phasea2=1;
    phasea3=1;
}
else if((keyin_char[0]=='4'))
{   phasea1=1;
    phasea2=1;
    phasea3=1;
    phasea4=1;
}
else if((keyin_char[0]=='5'))
{   phasea1=1;
    phasea2=1;
    phasea3=1;
    phasea4=1;
    phasea5=1;
}
```

Figure 3.24: Source code 12

```
while(RA2==1||RA3==1)              //scan colume
{
    RE1=1;
    RE0=1;
    RA4=0;
    RA5=1;


    if(RA2==0)                             //if key 'N' is being pressed
        {
            while(RA2==0)continue;         //waiting the key to be released
            lcd_clr();                     //Clear the LCD if the key is the 1st password
            lcd_goto(0);                   //The cursor of LCD points to the column equivalent to the value of password_count variable
            send_string("     CLEAR      ");//Display "     CLEAR      "
                __delay_ms(600);
            PORTC=0;
            PORTD=0;
            RE2=0;
            lcd_clr();                         //clear LCD
            lcd_goto(0);                       //cursor goto home
            send_string("To Reset press Y?");  //display "To Reset press Y?"
            continueloop();                    //goto continueloop fuction
            RA2==0;
            count=0;
            break;
        }
```

Figure 3.25: Source code 13

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Input Voltage To Solid State Relay From PIC16 Controller Under ‘ON’ State

The control voltage range of Crydom D4825-10 solid state relay is from 3-32V DC. Table 4.1 shows the output voltage from PIC can able to turn on ssr and turn off when there is no supply into it.

Table 4.1: Input Voltage To Solid State Relay

PHASE A	**VOLTAGE(V)**
Phase A ‘1’	3.50
Phase A ‘2’	3.47
Phase A ‘3’	3.47
Phase A ‘4’	3.47
Phase A ‘5’	3.48
PHASE B	**VOLTAGE(V)**
Phase B ‘1’	3.47
Phase B ‘2’	3.48

Phase B ‘3’	3.46
Phase B ‘4’	3.48
Phase B ‘5’	3.47
PHASE C	**VOLTAGE(V)**
Phase C ‘1’	3.47
Phase C ‘2’	3.47
Phase C ‘3’	3.46
Phase C ‘4’	3.47
Phase C ‘5’	3.47

4.2 Calculation and Result

Single-phase voltage into resistors is 240AC. A power resistor is 288ohm. Power resistors are connected in parallel. Table 4.2 shows the calculated results for maximum 5 of power resistors. Figure 4.1 shows the increases resistors connect in parallel, increase real power watt. Figure 4.2 shows the relationship between number of resistors and total resistors.

Formula:

Resistor (ohm) = $\frac{R_1 R_2}{R_1 + R_2}$

$$Power(w) = \frac{V^2}{R}$$

Table 4.2:Calculated Results

Number Of Resistor	**1**	**2**	**3**	**4**	**5**
Real power watt	200	400	600	800	1000
Total	288	144	96	72	57.6

Resistor(Ω)					

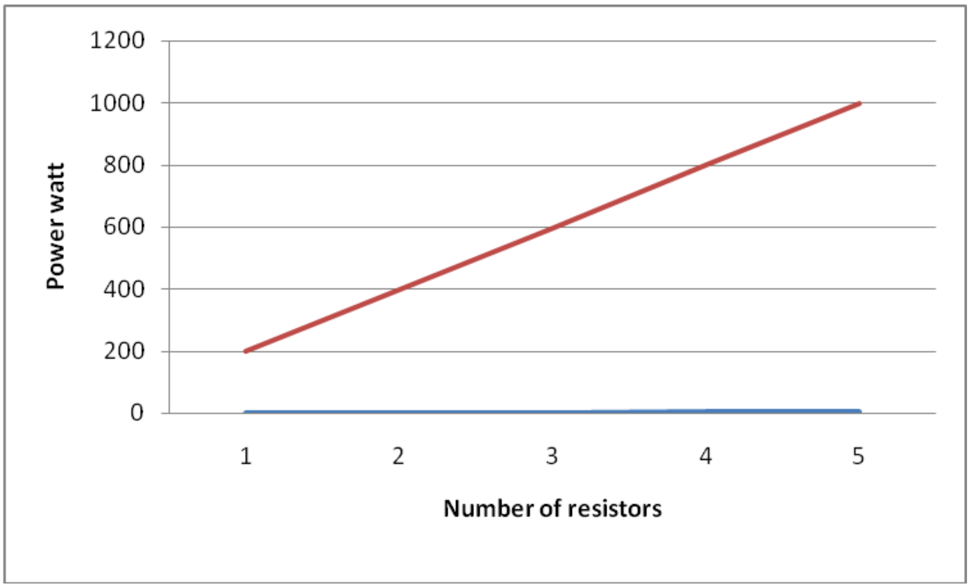

Figure 4.1: Graph For Real Power VS Number of Power Resistor

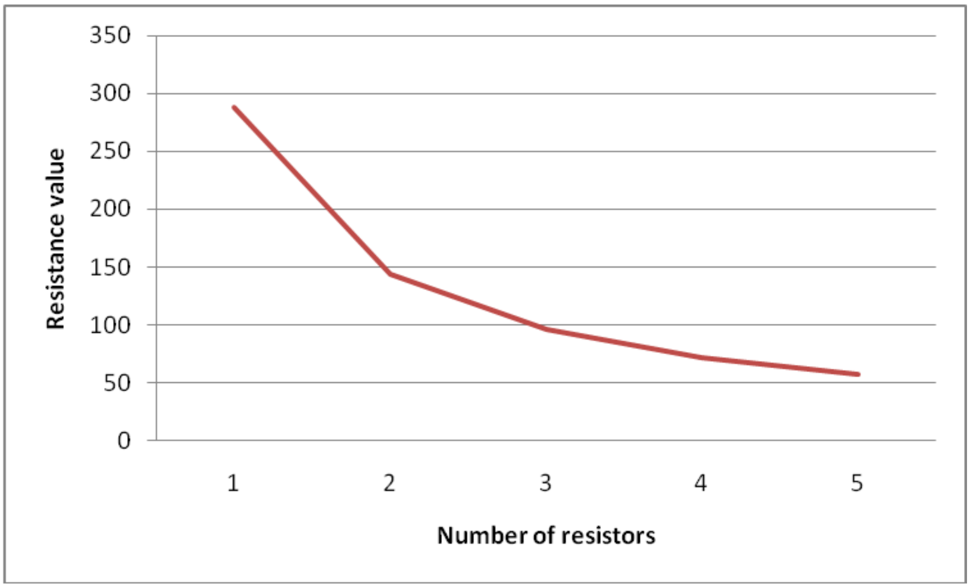

Figure 4.2: Graph For Resistance VS Number Of Resistor

4.3 Comparison of PSCAD Result and Calculated Result

The PSCAD result at the starting is almost similar to calculated result. However, number of resistors connect in parallel are getting increase the power(watt) as a result getting slide different compare PSCAD and calculated result. The power (watt) result gain from PSCAD is lower than calculated result. Table 4.3 shows the PSCAD result. Figure 4.3 shows the relationship between power and number of resistors.

Table 4.3: PSCAD Result

Number Of Resistor	1	2	3	4	5
Real power watt	198	398	592	785	968

Figure 4.3: PSCAD Graph For Real Power VS Number of Power Resistor

4.4 Microcontroller Result

Figure 4.4 shows the micontroller and LED indicator board. LCD first to shows "288ohm PER LOAD" on first line and second line shows "KEY IN VALUE 0-5". After 1.2s LCD display change to shows "PLS KEY IN LOAD", " FOR PHASE A " on second line is shown in figure 4.6.

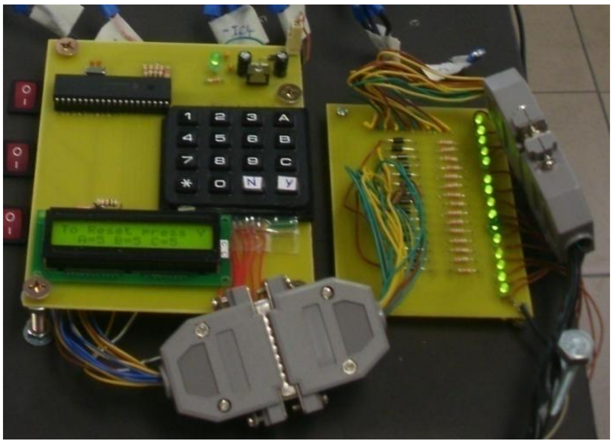

Figure 4. 4: Microntroller and LED Indicator

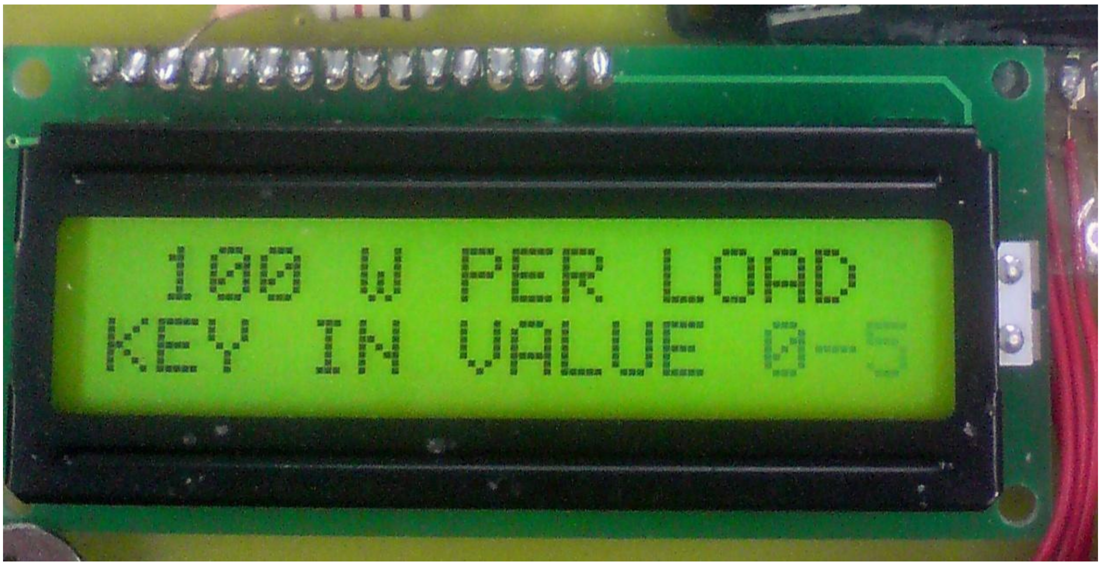

Figure 4.5: Display 1

Figure 4.6: Display 2

After that, if user presses 0-5 on the keypad then only the LCD display “PLS KEY IN LOAD”, “ FOR PHASE B ” is shown in figure 4.7.

Figure 4.7: Display 3

After that, if user presses 0-5 on the keypad then only the LCD display “PLS KEY IN LOAD”, “ FOR PHASE C ” is shown in figure 4.8.

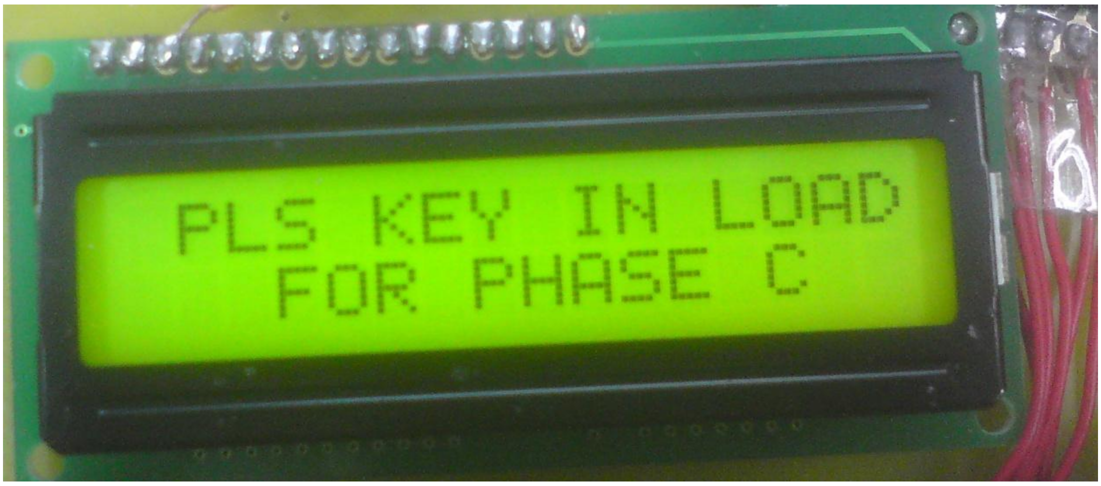

Figure 4.8: Display 4

After every load profile has been enter for three phases then LCD display number of value has entered on the phases and request confirmation from user as shown in figure 4.9.

Figure 4.9: Display 5

If user presses ‘Y’ the system will turn on particullar output pin as user enters (figure 4.10). The LED indicator light turn on according to user enters as well as SSR(figure 4.11).If user enters ‘N’ the LCD shows “ CLEAR “and wait for user to enter ‘Y’ to run the program again.

Figure 4.10: Display 6

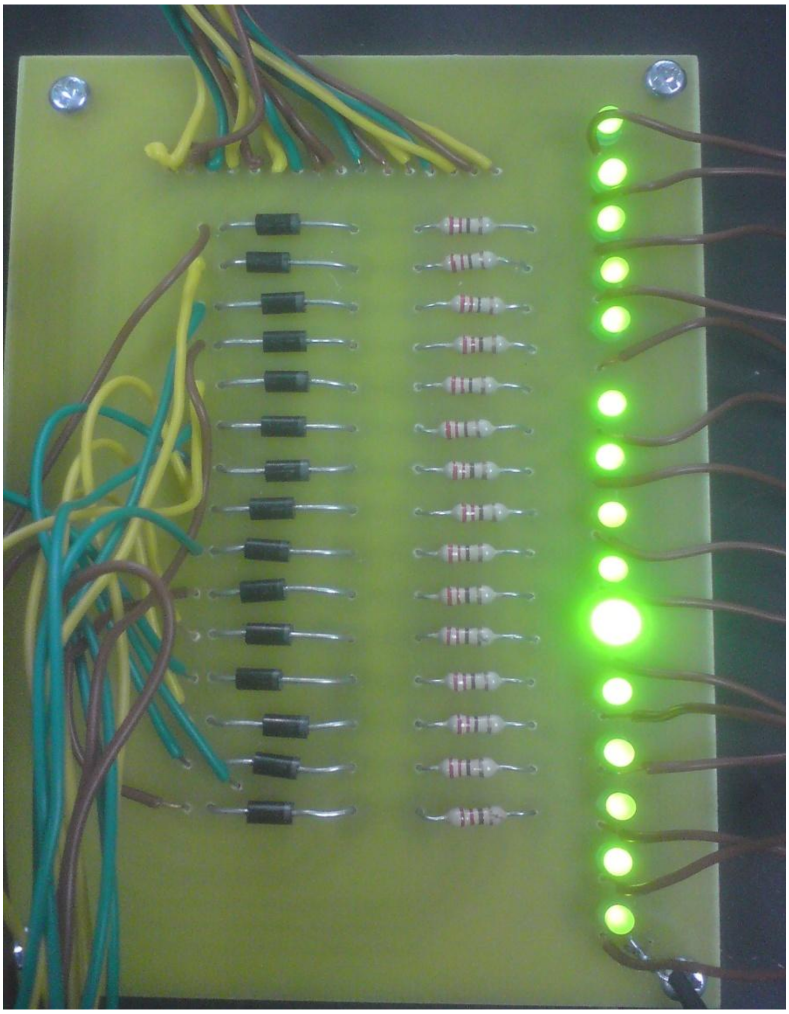

Figure 4.11: Display 7

4.5 Discussion

Generally, this project has overall objectives. All the operating was successfully goes on with the load emulator flow chart in chapter 1. The main component of load emulation is the algorithm that generates achieved the desired currents to be drawn so as to mimic the actual load. Users are able to control the load applied into the line phase. By using this system user can studies the characteristic of load.

Load emulator has designed to be easy use and the instruction given from the controller to user is clearly understood. Besides that, protection of the circuit can ensure the safety for user. This load emulator provides two switching type which are manual switching and auto switching. Both of these types can able to select number of resistive load fire and select which phase to fire the load into it. Moreover load emulator can make power changes of each line phase. The load on each phase can be varied in 200W to a maximum 1000W at nominal voltage.

Overall load emulator is a part of SSEZ (Small Scale Energy zone).

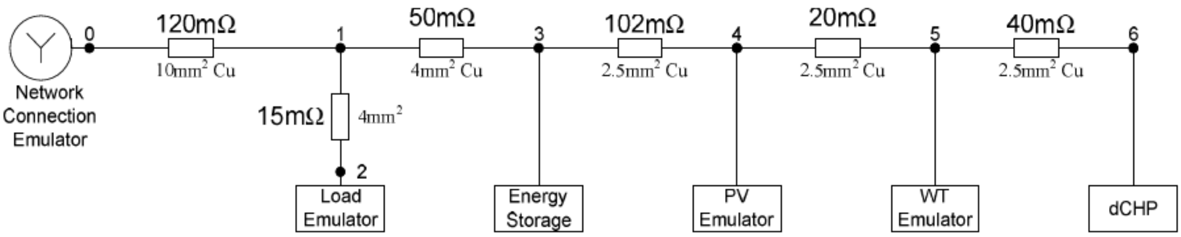

Figure 6.3: Single line diagram of the topology of the Experimental SSEZ

4.6 Problems encountered and troubleshooting

During this "Load Emulator" progress, a lot of problems had been encountered from designing the system until the end of the project. A lot of efforts have been paid to accomplish this system.

Sometime, hardware fabricated is not ideality same with blueprint of design. A variety of problems will occur every step of progressing even if detail considering on blueprint. So, the only way to solving is always keep improve, troubleshooting and consult with supervisor and experience senior.

4.6.1 Control Unit

i) Software

When designing the system by using microcontroller, several days spent on study in microcontroller book and internet research is necessary to learn the basic programming skill in order to help on source code writing.

In programming work of microcontroller, many time of fail is face because lack of experience and advance knowledge on this region. When appear failing, utmost to find the solution and study until whole of programming work is successful in order to able cumulate knowledge and experience to handle same problem occur at the next time.

Often using breadboard to testing the function of microcontroller when new source code is build in the microcontroller to achieve less error occur when microcontroller perform task controlling. Besides, thing more about the probability condition may happen on the system also can avoid error occur when real practice. Always keep correction on programming to let the microcontroller perform well at real practice. If the functions of microcontroller occur error, troubleshooting is taking to improve function of microcontroller.

ii) Hardware

Fabricate hardware control unit is challenging on electronic knowledge. Microcontroller is device of easy to broken, especially factor on thermal, voltage and current. If circuit is short circuit or some circuit path is some connection wrong, microcontroller may burn because voltage of supply not suitable with microcontroller. To fabricate control unit, effort study principle circuit and circuit theorem is necessary to design a circuit harmonic between the microcontroller and other IC device. Buffer and RC circuit are using to avoid the high transient pulse influence result of microcontroller. Voltage supply of control unit using voltage regulator to maintain the voltage is always on suitable value (5V). Every microcontroller burning case occur, find and remember the causes of fail and learned until the circuit is successful.

Besides that, always research the IC or component to improve and solving the problem on the control unit part.

4.6.2 PCB Design

When designing the Printed Circuit Board (PCB), spend some days to study about the software of “Diptrace” is necessary. “Diptrace” Software is adopted in PCB design because this software is free. In PCB design, width of path ways needs to regard in the circuit. If the width is too wide, power loss will high and the space of design will be narrow. If the width is small, the path way may not conduct. The best way to solve this problem is print out the circuit design on the white paper and show with experience lab assistance to consult the design is valid before etch process is taking.

Besides, size of board is important factor on the project. It is need to accurate the dimension according space of box to envelop the PCB and special dimension to considering about the arrangement doing on the packaging process

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

All the information detected is connected to microcontroller system and LED connections to solid state relay are in appropriate situation. The overall of objectives in chapter 1 is achieved. Auto switch used PIC to control while manual switch used hand to switch on and off. It has been shown that although the dc input voltage for solid state relay is different by using auto switch or manual switch, but the result for the measurement of the load are the same. The microcontroller and manual switch is just giving signal to turn on the SSR. The load emulator offers a more flexible platform for testing inverters in a laboratory environment. The load emulator can provide different load characteristics with regard to the typical load variation. The load emulator can provide different load characteristics with regard to the typical load variation. Variety of load profiles can be generated with the different combination of 15 resistors. The load emulator produce different load characteristics can be studied.

The work described in this thesis focuses on the experimental development and evaluation of this load emulator connected with the power grid systems that behave as the real load. Using load emulation, the feasibility of connecting a particular load to a grid under various conditions can be studied in the absence of any electromechanical machinery.

5.2 Recommendations

The developed load emulator can improve by implement Bluetooth system which interact personal computer with microcontroller hence the keypad will not be use in the project. *Bluetooth* low energy signals have a range of 100 meters or more. The adaptive frequency hopping employed by *Bluetooth* technology allows the device to switch among many frequencies to not just reduce interference but also to actively identify crowded spectra and avoid them. Therefore if the Bluetooth device implement into system it can run more effective with less wiring. Besides that, less microcontroller pins use hence more sensors and devices can be implemented into the system.

The developed load emulator just consider resistive load for further improvement capacitive and inductive load can be implement inside load emulator hence more studies on load characteristic. For Both inductive and capacitive loads create what is known as reactance in an AC circuit. Reactance is a circuit element's opposition to an alternating current, caused by the build up of electric or magnetic fields in the element due to the current and is the "imaginary" component of impedance, or the resistance to AC signals at a certain frequency. By applying this formula $Z = R + jX$ into system as improvement on load emulator.

.

REFERENCES

1. Y. Srinivasa Rao and Mukul Chandorkar, “Electrical Load Emulation using Optimal Feedback Control Technique,” *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 748-753, Gippsland, Australia, 9-13 February 2009.

2. Y. Srinivasa Rao and Mukul Chandorkar, “Electrical Load Emulation using Power Electronic Converters,” *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 1-6, Gippsland, Australia, 19-21 November 2008.

3. Y. Srinivasa Rao and Mukul Chandorkar, “Rapid Prototyping Tool for Electrical Load Emulation using Power Electronic Converters,” *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 106-111, Gippsland, Australia, 4-6 October 2009.

4. Irving M. Gottlieb. (1993). *Electronic Power Control*. McGraw-Hill Education.

5. Vinod Kumar Khanna. (2003). *The Insulated Gate Bipolar Transistor: IGBT Theory And Design.* Wiley-Interscience.

6. M. H. Rashid. (2007). *Power Electronics Handbook: Devices, circuits, And Applications.* Academic Press.

7. Nihal Kularatna. (1998). *Power electronics design handbook: low-power components and applications*. Newnes.

8. *The free encyclopedia*. (2011, January 14). FL: Wikimedia Foundation, Inc. Retrieved January 14, 2004, from http://en.wikipedia.org/wiki/Screw terminal

9. *The free encyclopedia*. (2011, April 6). FL: Wikimedia Foundation, Inc. Retrieved April 6, 2011, from http://en.wikipedia.org/wiki/Power_supply.