PSZ 19:16 (Pind. 1/07)

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT PAPER AND COPYRIGHT

Author's full name : GAN CHYUAN WUU

Date of birth : 28/4/1988

Title : SECURED ROOM ACCESS SYSTEM

Academic Session: 2010/2011

I declare that this thesis is classified as :

☐ **CONFIDENTIAL** (Contains confidential information under the Official Secret Act 1972)*

☐ **RESTRICTED** (Contains restricted information as specified by the organisation where research was done)*

☑ **OPEN ACCESS** I agree that my thesis to be published as online open access (full text)

I acknowledged that Universiti Teknologi Malaysia reserves the right as follows :

1. The thesis is the property of Universiti Teknologi Malaysia.
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by :

SIGNATURE

880428-04-5401

(NEW IC NO. /PASSPORT NO.)

Date : 13/5/11

SIGNATURE OF SUPERVISOR

DR.IZZELDIN IBRAHIM MOHAMED ABDELAZIZ

NAME OF SUPERVISOR

Date : 13/5/11

NOTES : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organisation with period and reasons for confidentiality or restriction.

“I declare that I have read this report and in my opinion this report is sufficient in terms of scope and quality for award a Degree of Bachelor of Electrical Engineering (Electrical- Electronics).”

Signature: ....

Name of Supervisor: DR.IZZELDIN IBRAHIM
MOHAMMED ABDELAZIZ

Date: 27 MAY 2011

SECURED ROOM ACCESS SYSTEM

GAN CHYUAN WUU

A report submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering (Electrical- Electronics)

Faculty of Electrical Engineering

Universiti Teknologi Malaysia

MAY 2011

I declare that this report entitled "*Secured Room Access System*" is the results of my own research except as cited in the references. The report has not been accepted for any degree and submitted in candidature of any other degree.

Signature : …………………………………….

Name : GAN CHYUAN WUU

Date : 27 MAY 2010

DEDICATION

To my beloved father and mother

Gan Ching Hoo
&
Lum Kim Wah

Sisters
Gan Miao Yong, Gan Mien Chen

To all families, my close friends, my coursemate and all my lectures who have encouraged, guided and inspired me throughout the journey of my education.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of supervisor and colleague. I would like to extend my sincere thanks to all of them.

I am highly indebted to Dr Izzeldin for his guidance and constant supervision as well as for providing necessary information regarding the project & also for support in completing the project.

I would like to express my gratitude towards my parents for providing everything, such as money, to buy anything that are related to this project work and their advise, which is the most needed for this project. They also supported me and encouraged me to complete this project so that I will not procrastinate in doing it.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

ABSTRACT

Due to the increasing need for securing data and places, the numerical code authentication is seeing large market growth. A scalable, small, and efficient device that can be used to secure doorways throughout a complex has been introduced in this project. Microcontroller has been used in this project which provides an efficient interaction between input and output component. The special characteristic of PIC 16F877A give some value added to the system. The main advantage of this project is the user can change the stored password with new password by entering the master password written in the user manual. Besides that, storing password in the EEPROM of the microcontroller allows the stored password being saved in the memory even though the power supply has been removed. Nevertheless, the cost and power requirement is low compared to the other microcontroller family. The system development divided into software and hardware development which are MPLAB IDE software have been using to write the programming and PICkit 2 for tools of burning. The coding used in this hardware has been written using C language for easy to edit and control the bit and register in microcontroller.

ABSTRAK

Memandangkan keperluan keselamatan data dan tempat menjadi semakin meninggi, maka industri kod berangka pengesahan didapati mempunyai pertumbuhan pasaran yang besar. Sesuatu peranti yang kecil dan cekap telah diperkenalkan dalam projek ini untuk menjaga keselamatan di seluruh kompleks. Mikrokontroler digunakan dalam projek ini telah menyumbang kepada interaksi yang cekap antara bahagian input dan output. Ciri- ciri khusus bagi PIC16F877A telah memberi beberapa nilai tambahan kepada sistem. Kelebihan utama projek ini ialah pengguna boleh menukar kunci laluan tersimpan kepada kunci laluan yang baru dengan memasukkan ‘master password’ yang tercatat dalam panduan pengguna. Selain itu, menyimpan kunci laluan dalam ‘EEPROM’ mikrokontroler membolehkan kunci laluan tersimpan disimpan dalam ingatan walaupun bekalan elektrik telah diputuskan. Tambahan pula, kos dan keperluan kuasa bagi keluarga mikrokontroler yang digunakan dalam projek ini adalah lebih rendah berbanding dengan keluarga mikrokontroler lain. Pembanguan sistem boleh dibahagi kepada software dan hardware pembangunan dimana MPLAB IDE software digunakan untuk menulis pengaturcaraan dan PICkit 2 digunakan sebagai pembakar mikrokontroler. Kod yang digunakan dalam projek ini ditulis dalam bahasa ‘C’ supaya bit dan register dalam mikrokontroler senang diedit and dikawal

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION		
PIC	-	Programmable Interface Controller
LED	-	Light Emitting diode
IDE	-	Integrated Development Environment
EEPROM	-	Electrically Erasable Read Only Memory
RFID	-	Radio Frequency Identification
DC	-	Direct Current
AC	-	Alternative Current
LCD	-	Liquid Crystal Display
MCLR	-	Master Clear
WDT	-	Watch Dog Timer
INT	-	Interrupt
PCB	-	Printed Circuit Board
RISC	-	Reduced Instruction Set Computer
Vss	-	Supply Voltage
+Vref	-	Reference Voltage
SSP	-	Synchronous Serial Port
ICSP	-	In-Circuit Serial Programming

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Nowadays, the security systems are an important aspect to prevent unauthorized users to access the system security door. With the increase in incidences of crime, most home owners are looking for security measures that can keep them and their family members safe. Traditionally security door system has been done through the use a conventional locked and keys. Door is the first gateway for the house and divides the inside and outside world. It is the first level of protection against unwanted intruders. Hence, investing time and money in securing a door properly is a very wise decision. In this era, many manufactured have come up with electronic access control system. The most well-known access control system is magnetic card and keypad system. Perhaps the most common form of electronic lock is that using a numerical code for authentication; the correct code must be entered in order for the lock to deactivate.

In this project, a microcontroller-based digital lock which is effective, flexible, and affordable was designed. The digital lock functions as access control system that allows only authorized persons to access a restricted area. Whether the user wants protection for their home, industrial premises or even a small security box, this security system will fulfill their security needs. Most importantly, this security system will make the user feels safe and secure being away from home.

1.2 Problem Statement

Security is becoming increasingly important in many facets of society, ranging from military, corporate world, and personal lives. One area of technology that is seeing growth is that of access control system where a user is granted access to a piece of data, room, etc. via some form of numerical code authentication.

Due to the increasing need for securing data and places, the numerical code authentication industry is seeing large market growth. This need for securing data and places effectively, coupled with the project market growth of the access control system industry, led us into a project where we created a system that uses numerical code authentication for a user to enter through a doorway.

1.3 Project Objectives

The objectives of this project are:

1. Create a system that uses PIC microcontroller based electronic lock to allows authorized persons to access a restricted area.

2. Design a system that when an authorized person enters a predetermined number (password) via the keypad, the relay energizes for a limited time to unlock the solenoid operated lock, so door can be pushed/pulled open. At the end of the preset delay, the relay de-energizes and the door gets locked again. A prompt message is displayed on the LCD module.

1.4 Scope of Project

There are a few guidelines and scopes listed to ensure that the project conducted within its intended boundary. This is to make sure the project is heading the right path.

This project involved the software and hardware development. For the hardware development of the project is to design a secured room door lock using PIC 16F877A microcontroller and design a PCB board for connecting other components such as 4x4 matrix keypad, LCD displays, relay and voltage regulator. While for the software development, C++ used as the development tool to program the system coding and build the device driver for interfacing between the software and hardware.

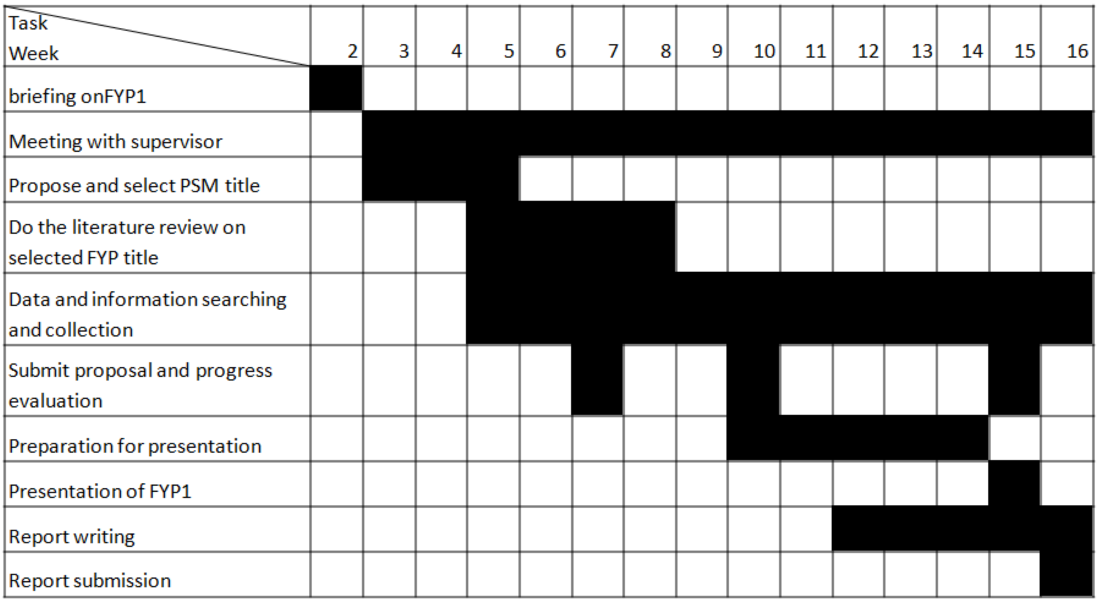
1.5 Thesis Outline

This thesis consists of five chapters and each chapter is briefly discussed here. The first chapter introduced the problem statement, project objectives and scope of project. While chapter 2 reviews the background and theory that is related to this project. Chapter 3 presents the project methodology and application tool that have been used in this project. The results for each stage in the system and discussion of the overall result will be shown in chapter 4. In the last chapter, the research work is summarized and some recommendations are proposed for future development.

1.6 Gantt Chart

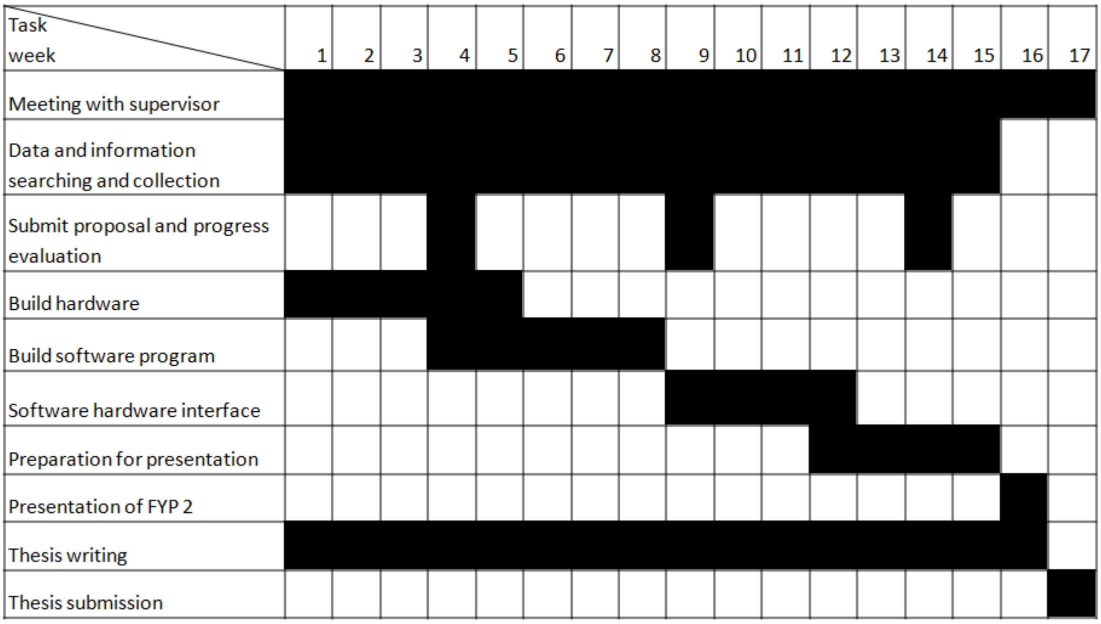
Gantt chart shows the overall project planning to ensure the project was accomplished at the end of the semester. First semester's Gantt chart shows the planning on development of project concept and proposal writing. Second semester's Gantt chart shows the planning on both software and hardware implementation.

Table 1.1: FYP1's Gantt chart



Task Week	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
briefing onFYP1	■														
Meeting with supervisor		■	■	■	■	■	■	■	■	■	■	■	■	■	■
Propose and select PSM title		■	■	■											
Do the literature review on selected FYP title				■	■	■	■								
Data and information searching and collection				■	■	■	■	■	■	■	■	■	■	■	■
Submit proposal and progress evaluation						■			■					■	
Preparation for presentation									■	■	■	■	■		
Presentation of FYP1														■	
Report writing											■	■	■	■	■
Report submission															■

Table 1.2: FYP2's Gantt chart



Task week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Meeting with supervisor	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Data and information searching and collection	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Submit proposal and progress evaluation				■					■					■			
Build hardware	■	■	■	■	■												
Build software program				■	■	■	■	■									
Software hardware interface									■	■	■	■	■				
Preparation for presentation												■	■	■	■		
Presentation of FYP 2																■	
Thesis writing	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
Thesis submission																	■

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter includes the study of hardware use in the project such as PIC16F877A Microcontroller, LCD Display, 4x4 Matrix Keypad and Magnetic Contact. In addition, it's also review about the history of lock and variety type of conventional lock system.

2.2 PIC16F877A Microcontroller

PIC is a family of Harvard architecture microcontrollers made by Microchip Technology, derived from the PIC1650 originally developed by General Instrument's Microelectronics Division. The name PIC was originally an acronym for "Programmable Intelligent Computer". PICs are popular with developers and hobbyists alike due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and reprogramming with flash memory) capability [7].

The PIC16F877A is an 8-bit microcontroller based on reduced instruction set computer (RISC) architecture. The PIC16F877A provides the following features [4]:

i. 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

ii. Peripheral Features
   - Timer0: 8-bit timer/counter with 8-bit prescaler
   - Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
   - Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler

- Two Capture, Compare, PWM modules:
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

iii. Analog Features

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

iv. Special Microcontroller Features

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming

- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

v. CMOS Technology

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

The 16F877A is one of the most popular PIC microcontrollers and it comes in a 40 pin PDIP pin out which consist of many internal peripherals. The 40 pins make it easier to use the peripherals as the functions are spread out over the pins and make it easier to decide what external devices to attach without worrying too much is there enough pins to do the job. One of the disadvantages of the device is that it has no internal oscillator so the user needs an external crystal of other clock source. Figure 2.1 and 2.2 shows PIC 16F877A pin out and its block diagram.

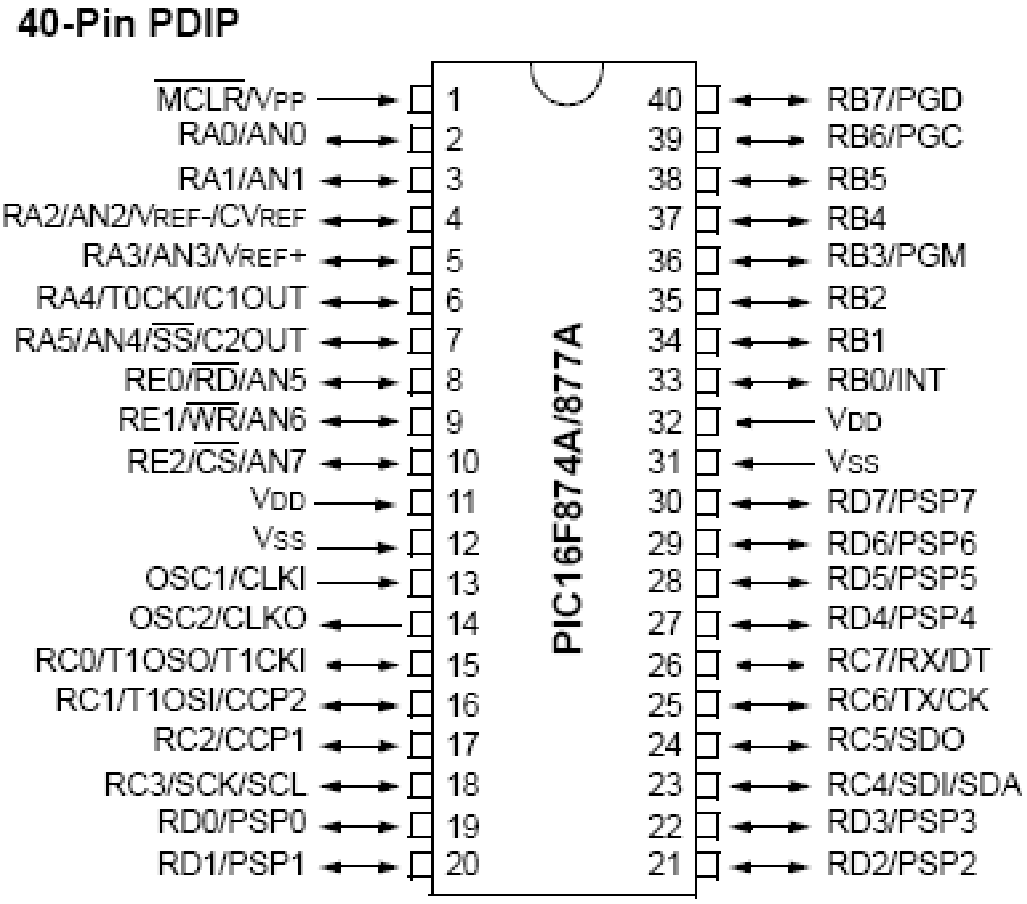


Figure 2.1 Pin out of PIC 16F877A

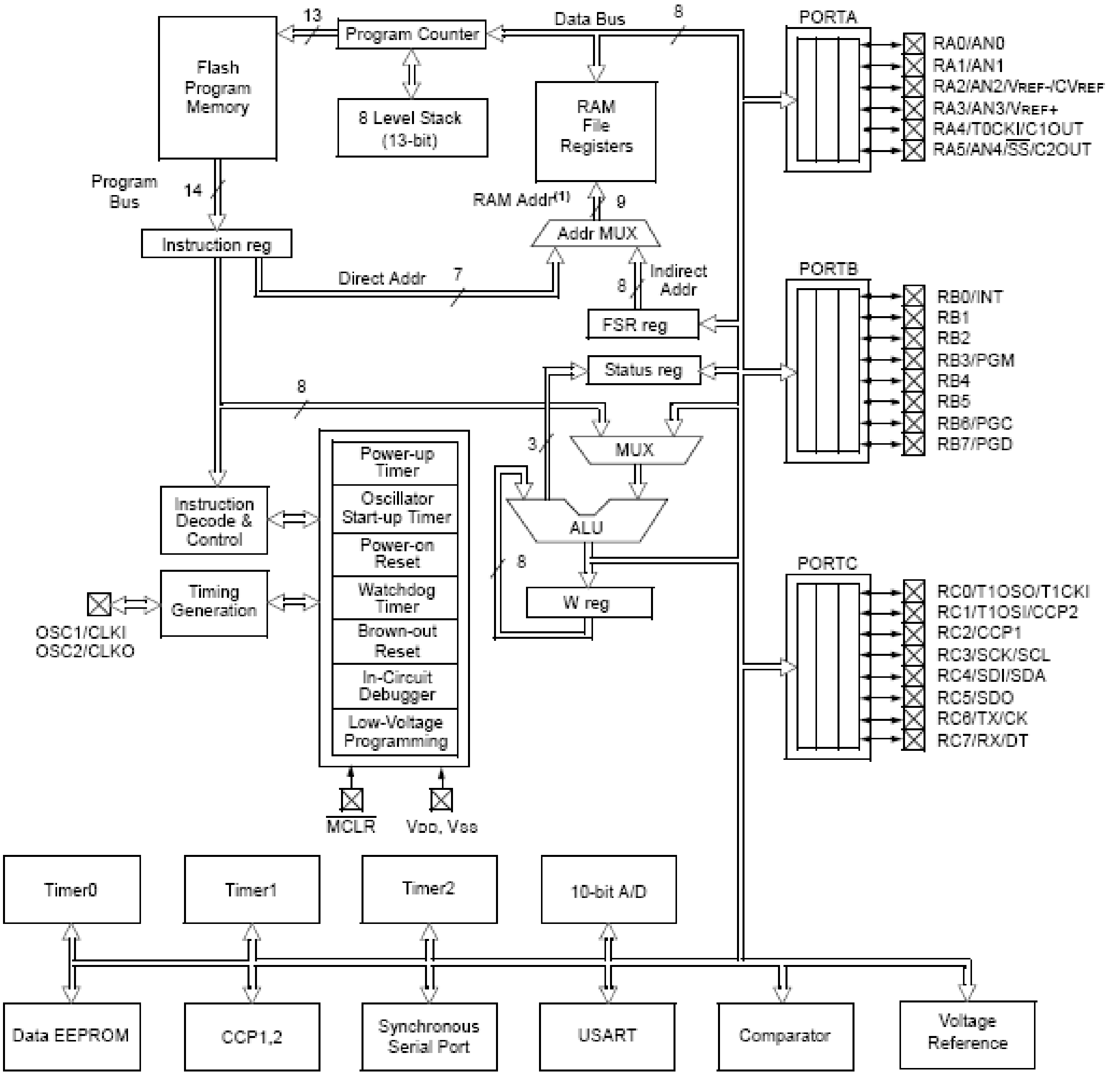


Figure 2.2 Block Diagram of PIC16F877A

2.3 LCD Display

Short for liquid crystal display, a type of display used in digital watches and many portable computers. LCD displays utilize two sheets of polarizing material with a liquid crystal solution between them. An electric current passed through the liquid causes the crystals to align so that light cannot pass through them. Each crystal, therefore, is like a shutter, either allowing light to pass through or blocking the light.

Monochrome LCD images usually appear as blue or dark gray images on top of a grayish-white background. Color LCD displays use two basic techniques for producing color: Passive matrix is the less expensive of the two technologies. The other technology, called thin film transistor (TFT) or active-matrix, produces color images that are as sharp as traditional cathode ray tube (CRT) displays, but the technology is expensive. There are many types of LCDs that are designed for both special and general uses. They can be optimized for static text, detailed still images, or dynamic, fast-changing, video content. They are used in a wide range of applications including: computer monitors, television, instrument panels, aircraft cockpit displays, signage, etc. They are common in consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones. LCDs are more energy efficient and offer safer disposal. Its low electrical power consumption enables it to be used in battery-powered electronic equipment [11].

LCD displays come in many sizes most often named by the number of rows and then the length of the display line. For example a 2x16 LCD display will have two rows of sixteen characters and a 4x20 LCD display will have four rows with twenty characters in each. LCDs can be have backlighting or be reflective (think calculator). In either case the programming that goes into working these displays is the same. Most LCDs that are being made now come with one row of sixteen pins. The first fourteen pins are used to control the display and the last two are for the backlighting (if the display has backlighting) [11]. Figure 2.3 show sample of the LCD display.

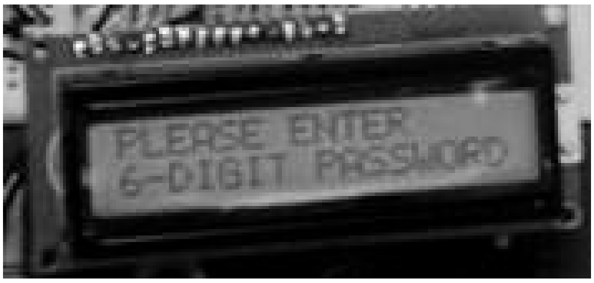


Figure 2.3 LCD Display

2.4 4x4 Matrix Keypad

A keypad is a set of buttons arranged in a block or "pad" which usually bear digits, symbols and usually a complete set of alphabetical letters. If it mostly contains numbers then it can also be called a numeric keypad. Keypads are found on many alphanumeric keyboards and on other devices such as calculators, push-button telephones, combination locks, and digital door locks, which require mainly numeric input [10].

A simple 4x4 matrix keypad allows data entry into bus based systems. The keys on a keypad form a matrix of push-to-make switches. There are sixteen push switches, arranged on a matrix of four rows and four columns. These are connected to 8 pads at the bottom of the keypad. If no key is pressed there is no electrical contact between the rows and the columns. When a key is pressed it makes an electrical connection between the row and the column that it is on. So, for example, if the key labeled “2” is pressed, this makes electrical contact between row 1 (connected to pad 1) and column 2 (connected to pad 6). So, if key “2” is pressed this creates an electrical connection between pad 1 and pad 6. Figure 2.5 and 2.6 shows the sample of 4x4 matrix keypad and the keypad circuit connection. [10]

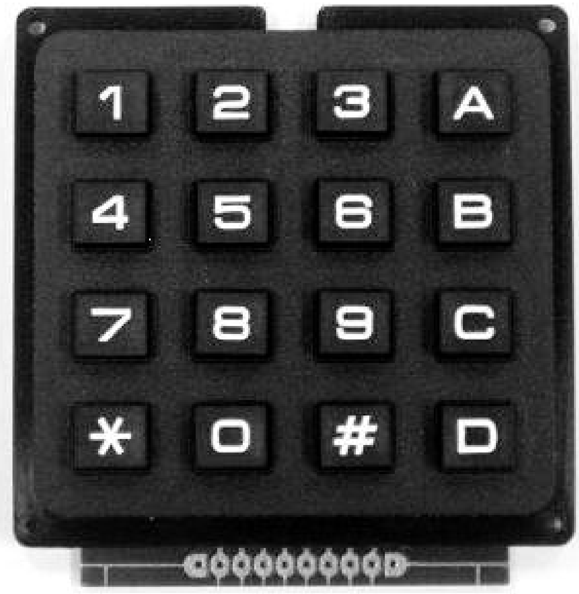


Figure 2.4 4x4 Matrix Keypad

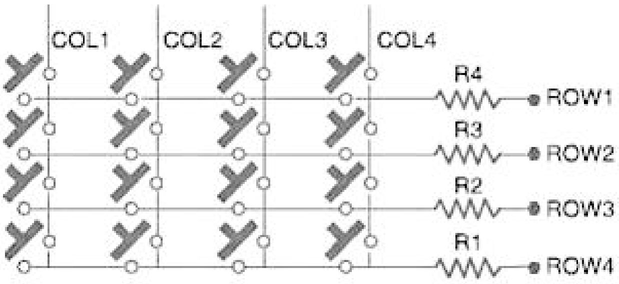


Figure 2.5 Keypad Circuit

2.5 Magnetic Contact

Magnetic contacts are usually NC (Normally Closed) which is used on the doors or windows. They consist of two parts, namely a magnet and a reed switch. When the reed switch is in close proximity to the magnet, the switch will close and vice versa. Usually the magnet is fitted to the door and the reed switch is fitted to the door frame in close proximity to one another such that when the door is closed, the two parts are in close contact and hence the switch is closed. When the door is opened, the magnet will be a distance away from the reed switch and hence the switch will open. Figure 2.6 shows the sample of the magnetic contact. Figure 2.7 shows an installation of magnetic contact on the door or window. [6]

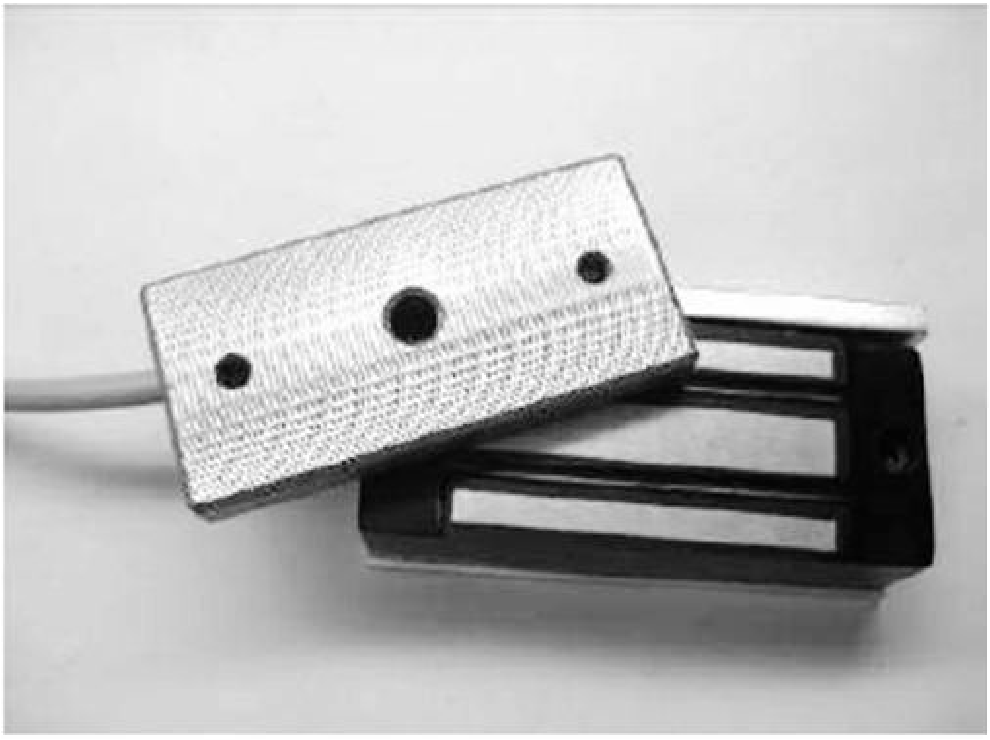


Figure 2.6 Magnetic Contact

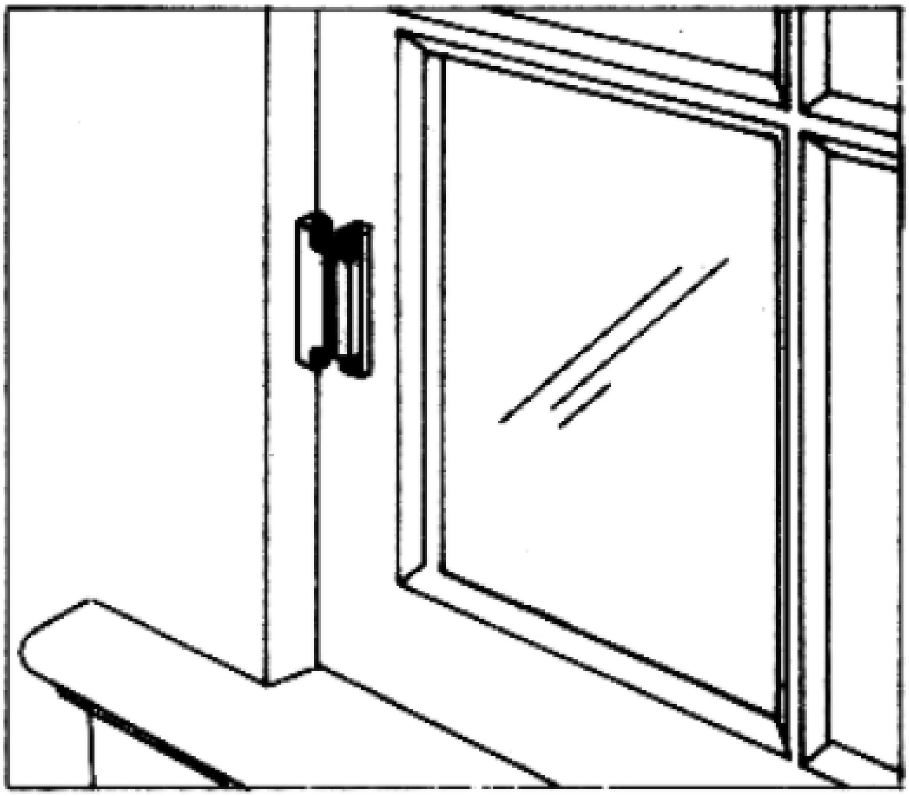


Figure 2.7 Installation of magnetic contact on the door or window

2.6 History of Security Door Lock

Securing one's property has long been a concern of people throughout the world. Beyond hiding the objects or constantly guarding them the most frequently used option is to secure them with a device. Historians are unsure where the first lock was invented, but evidence suggests that locks initially developed independently in the Egyptian, Greek, and Roman civilizations. Wooden locks and keys were in use as early as 4,000 years ago in Assyria. The first known lock with a key is a pin lock. The lock is strung on a rope hanging out of a hole in a door. A cylinder of wood with a hole drilled through its axis is the key, the length of the cylinder being the critical factor. The key is inserted into the hole and the bolt is pushed the correct distance. To lock the door the rope was pulled to extract the key cylinder, simultaneously pulling the bolt closed. This type of lock is still in use in certain parts of the world. Puerto Rico still uses this system. A disadvantage of this lock is that a vandal can push the rope into the hole — an ancient equivalent of putting glue into a lock [12].

Next, the warded lock was invented in China. In the Middle Ages, it came to Europe. This lock is still used in modern times when the security required is not high and cost is a significant factor. This lock has become the most recognizable lock/key design in the Western world. Lock puzzles were used to obscure the locking mechanism or even provide a non-functioning lock for the thief to waste time on. In the early 1900s a wooden pin lock with a wood key was discovered in Egypt and is believed to have been used in 250 b.c [12].

Early improvements in pin locks included increasing the number of pins to increase security, and changing the orientation of the pins to allow the key to provide the unlocking force instead of a rope, thus establishing the principles of the modern pin tumbler lock. [12]

2.7 Combinational Lock

A combination lock is a type of lock in which a sequence of numbers or symbols is used to open the lock. The sequence may be entered using a single rotating dial which interacts with several discs or cams, by using a set of several rotating discs with inscribed numerals which directly interact with the locking mechanism, or through an electronic or mechanical keypad [5].

From a mathematical standpoint, "combination" lock is a misnomer. In mathematics, the term combination represents a set of symbols that can be in any order: 1-2-3 is the same combination as 3-2-1, 2-3-1, and 1-3-2. However, the symbols or numbers for a lock must be entered in the correct sequence: 1-2-3 will not open a lock set to 3-2-1. The correct mathematical term for a set of symbols that must be entered in a specific sequence is a permutation. There are several type of combinational lock can be found in the market such as multiple dial lock and single dial lock. Figure 2.8 and 2.9 shows the sample of the multiple dial lock and single dial lock [5].

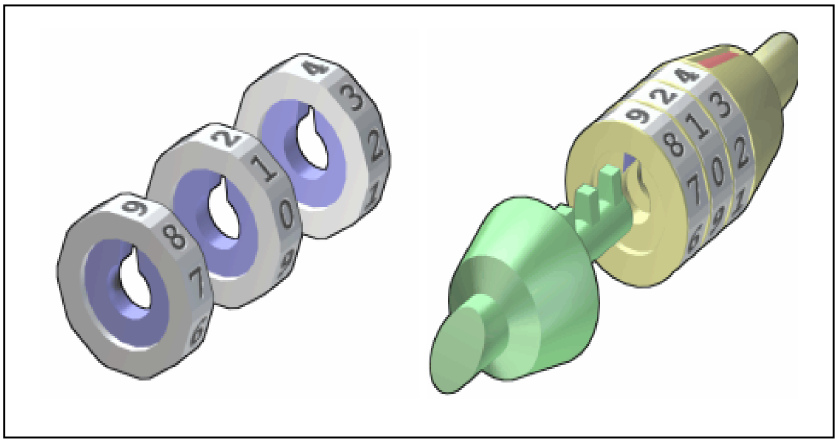


Figure 2.8 Multiple Dial Lock

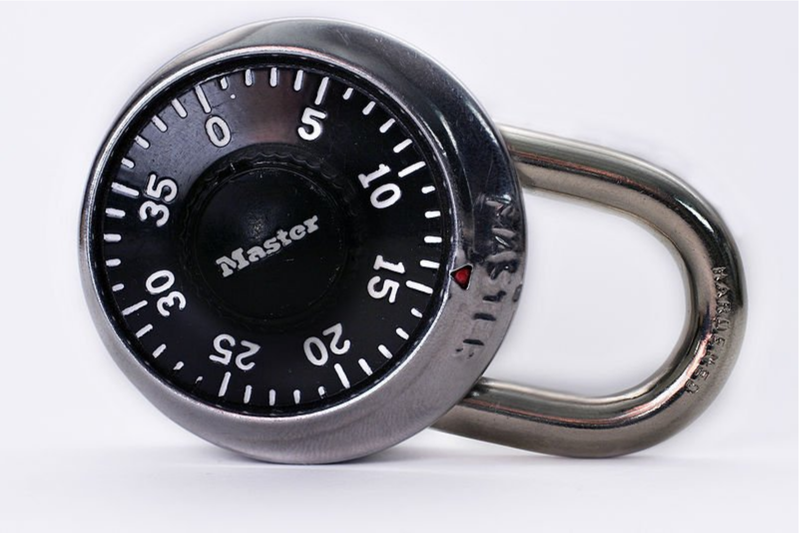


Figure 2.9 Single Dial Lock

2.8 Electronic Lock

Due to the growth of the technology, electronic lock has been developed to replace the combinational lock. An electronic lock (more precisely an electric lock) is a locking device which operates by means of electric current. Electric locks are sometimes stand-alone with an electronic control assembly mounted directly to the lock. More often electric locks are connected to an access control system. The advantages of an electric lock connected to an access control system include: key control, where keys can be added and removed without re-keying the lock cylinder; fine access control, where time and place are factors; and transaction logging, where activity is recorded.

Electronic locks use magnets, solenoids, or motors to actuate the lock by either supplying or removing power. Operating the lock can be as simple as using a switch, for example an apartment intercom door release, or as complex as a biometric based access control system. There are a lot of the high technology electronic locks in the market. Figure 2.10 shows one of the samples of electronic lock (Keypad Security Door Lock).

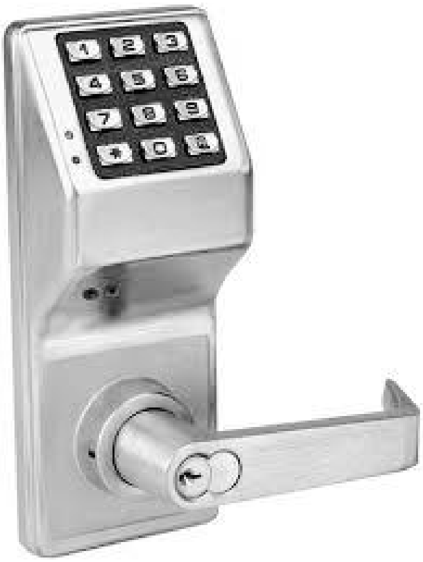


Figure 2.10 Keypad Security Door Lock

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter discusses the design overview, hardware and software implementation including the hardware design and software programming. For the hardware implementation, it consists of power supply, 2x16 LCD display and relay circuit. Software implementation will discuss about the main system programming, keypad scanning algorithm and read/ write EEPROM function.

3.2 Design Overview

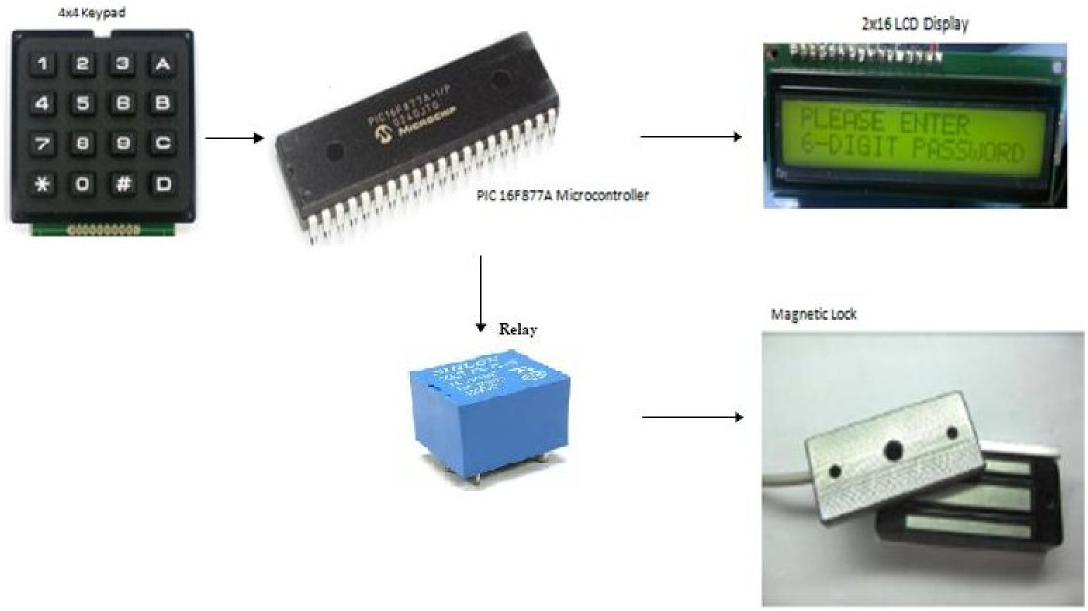


Figure 3.1 Design Overview

Figure 3.1 shows the overall design concept of the project. This project shows the basic of developing a secured room access system using microcontroller. From the figure 3.1, keypad is the input of the system while magnetic lock and the LCD display are the outputs of the system.

First of all, the microcontroller will set to a wait state in order to wait for the first button being pressed where to read the input from keypad. For the new user, the password will read and store in EEPROM initially. After setting the password, the data stored in the EEPROM will read and use to compare with the key enter by the user in order to access the restricted area.

After that, the matching result will show on the LCD display and LEDs indicator. If the key in password same with master password, system will ask for the setting new password. If key in password same with the store password, magnetic lock will be open and green led on. For entering wrong password, the door will be remaining close and red led on. User are only given 3 trial to enter the correct password, if the wrong password detected 3 times in sequence the system will stuck and ask the user to contact the administrator.

3.3 Hardware Implementation

This section will discuss about the components that had been used in this project which is power supply 5V, magnetic contact, LED as output of PIC microcontroller, push button as input of PIC microcontroller and 2x16 LCD display.

3.3.1 Power Supply

In this project there are two values power supply (12V and 5V) is required to support and operate the secured room access system. In order to operate the magnetic lock, an AC to DC adaptor with the output voltage12V has to apply to the device. The operating power for the PIC is 5V. So, the voltage regulator with the series L7805 is required to produce the 5V voltage from the 12V voltage. The voltage regulator L7805 consists of 3 terminal which is output, common and input terminal as shown in the figure 3.2. For producing the 5V power supply, the 12V power supply is connected to the input terminal of voltage regulator, common terminal is connected to the ground and the output terminal will produce a 5V voltage.

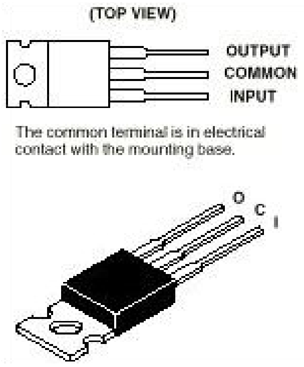

(TOP VIEW)
OUTPUT
COMMON
INPUT
The common terminal is in electrical contact with the mounting base.
O
C
I

Figure 3.2 IC LM7805

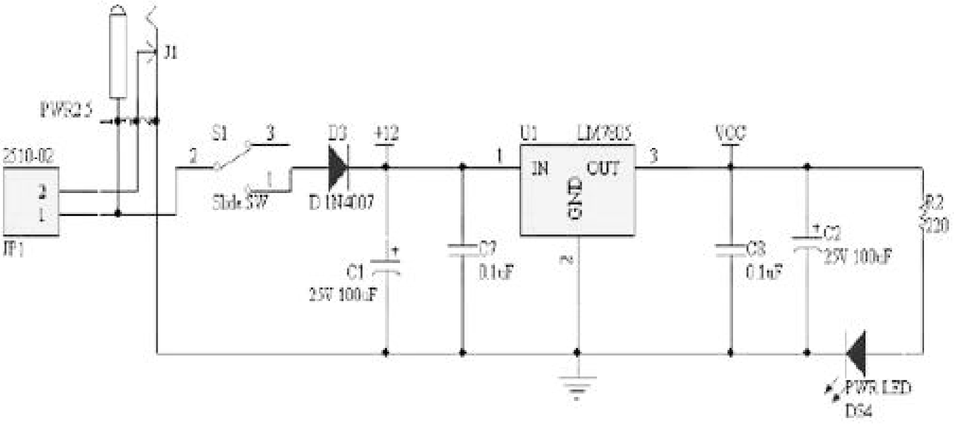


Figure 3.3 Power Supply Circuit

Based on the figure 3.3, there are two ways of supplying the 12V power supply to the circuit. One is using the AC to DC adaptor and another one is using the 12V battery to supply the power. The voltage regulator will generate some heat when the power is on. So, the user must be aware of the voltage regulator otherwise will get scalded. There is a diode can be found in the circuit shown in the figure 3.3 which is function as a protector to the circuit from the wrong polarity connection. Besides that, there are 4 capacitor can be found in the circuit which are function to stabilize the voltage at the input side and output side of the voltage regulator. In addition, there is a green LED connected at the output side of the voltage regulator which function to indicate the power status of the circuit and the resistor connect with the LED is use to protect the LED from getting over current flow and cause the LED getting burn.

3.3.2 2x16 LCD Display

LCD Modules can present textual information to user. It’s like a cheap “monitor” that you can hook in all of your gadgets. There are several types of LCD display can be found in the market. The most popular one can display 2 lines of 16 characters. The LCD modules use in the project consists of 16 pins for interfacing. The function of LCD modules used in this system is to display the welcome note to the user and asked the user to enter the password number. Figure 3.4 shows the sample of 2x16 LCD Display and table 3.1 shows the LCD pins function and configuration.

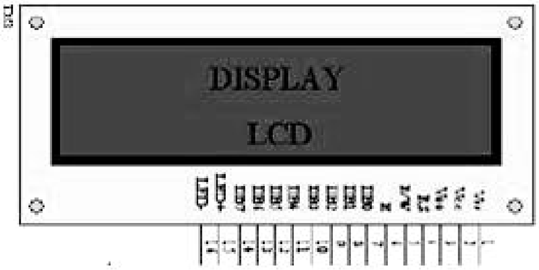
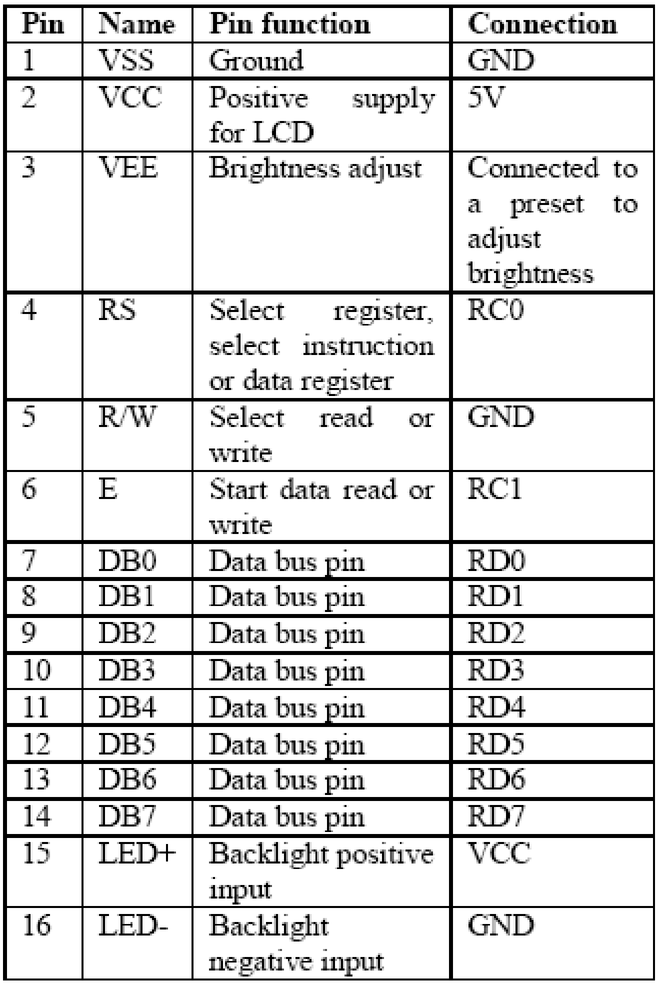


Figure 3.4 2x16 LCD Display

Table 3.1: LCD pins function and configuration



Pin	Name	Pin function	Connection
1	VSS	Ground	GND
2	VCC	Positive supply for LCD	5V
3	VEE	Brightness adjust	Connected to a preset to adjust brightness
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	RD0
8	DB1	Data bus pin	RD1
9	DB2	Data bus pin	RD2
10	DB3	Data bus pin	RD3
11	DB4	Data bus pin	RD4
12	DB5	Data bus pin	RD5
13	DB6	Data bus pin	RD6
14	DB7	Data bus pin	RD7
15	LED+	Backlight positive input	VCC
16	LED-	Backlight negative input	GND

3.3.3 Relay

Figure 3.5 shows a sample of relay which as an electromechanical switch. The relay is made up of an electromagnet and a set of contacts. When the current flow through the coil of the relay, it will create a magnetic field which will pull the switch to another contact. Normally, there are two different circuits connected to the relay and the relays are allowing one circuit become complete circuit for each position of the switch contact. So, using relay in the control appliance is allowing one circuit to switch to a second circuit which can be completely separate from the first. For instance, a low voltage battery circuit can use a relay to switch a 230v AC mains circuit.

Basically, there is a relatively large current required passes through the coil of the relay. It is about 30mA for a 12V relay. On the other hand, 100mA current can be use to operate the lower voltages circuit. Thus, transistor is usually used to amplify the small IC current to the larger value required for the relay coil. Relays are usually Single Pole Double Throw (SPDT) or Double Pole Double Throw (DPDT) but they can have many more sets of switch contacts, for example relays with 4 sets of changeover contacts are readily available.

Refer to the Figure3.5, the relay used for the project consists of 5 pins, 2 pins is the 2 end of the coil, 1 is COM, 1 is NO (normally open) and 1 is NC (normally close). One end of the coil is connect with the NPN transistor to amplify the small IC current to larger value required for the relay coil and another end is connected to 12V. NO pin is connected to a connector which is provided to locate the magnetic door lock and COM is connected to the 12V as well in this project.

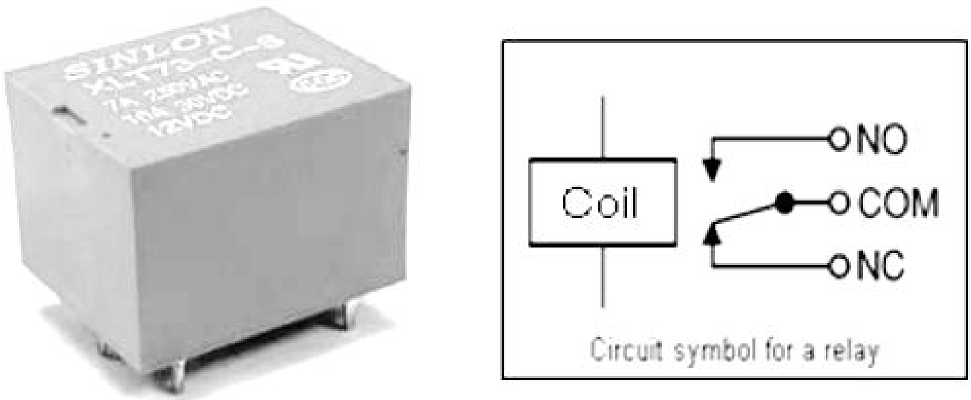


Figure 3.5 Relay

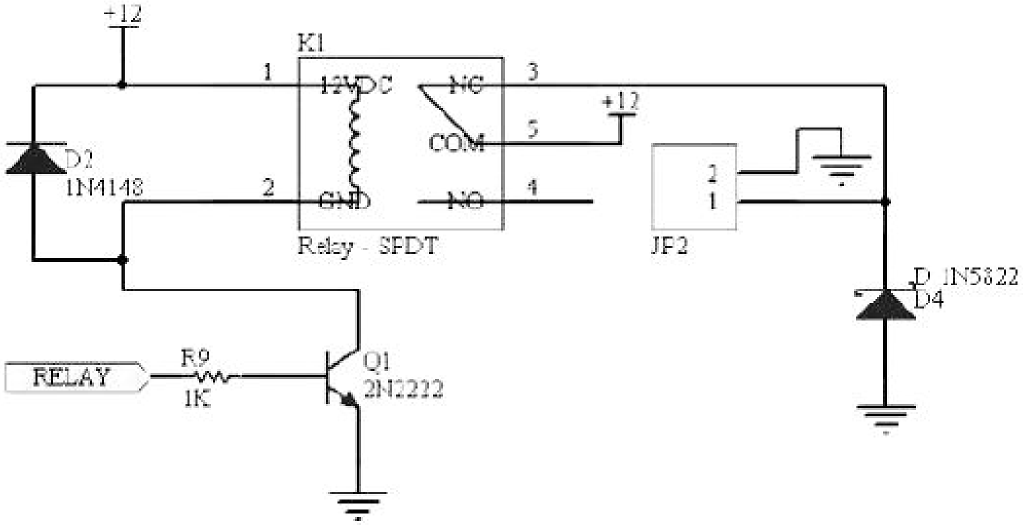


Figure 3.6 Magnetic Contact Circuit

3.4 Software Implementation

For software implementation, C++ language is used to program microcontroller. Besides, MPLAB is used to burn the programming hex file into the microcontroller.

3.4.1 Main System Programming

The flow chart for the system main program is shown in Figure 3.7. When the system is powered on, the microcontroller will initialize. The first task is to check whether the user is the new user or not. When the user is new user, system will ask for entering new password and stored in the EEPROM initially. If the user is not the new user, system will proceed to the next task which is checking the number of wrong password entered by the user. If the number of fail is equal to 3, the system will start infinite looping until the reset button being press. Otherwise, the system will always ask for entering password.

Next, system will checking for the password entered by the user is that equal to the master password. When entered password same with the master password, the user is allow to change the stored password with the new password and storing in the EEPROM. For the entered password same with the stored password, the magnetic lock will open and green LED on. Otherwise, the number of fail will be increase by 1and the magnetic lock will remain close. After that, the system will return back to the second task.

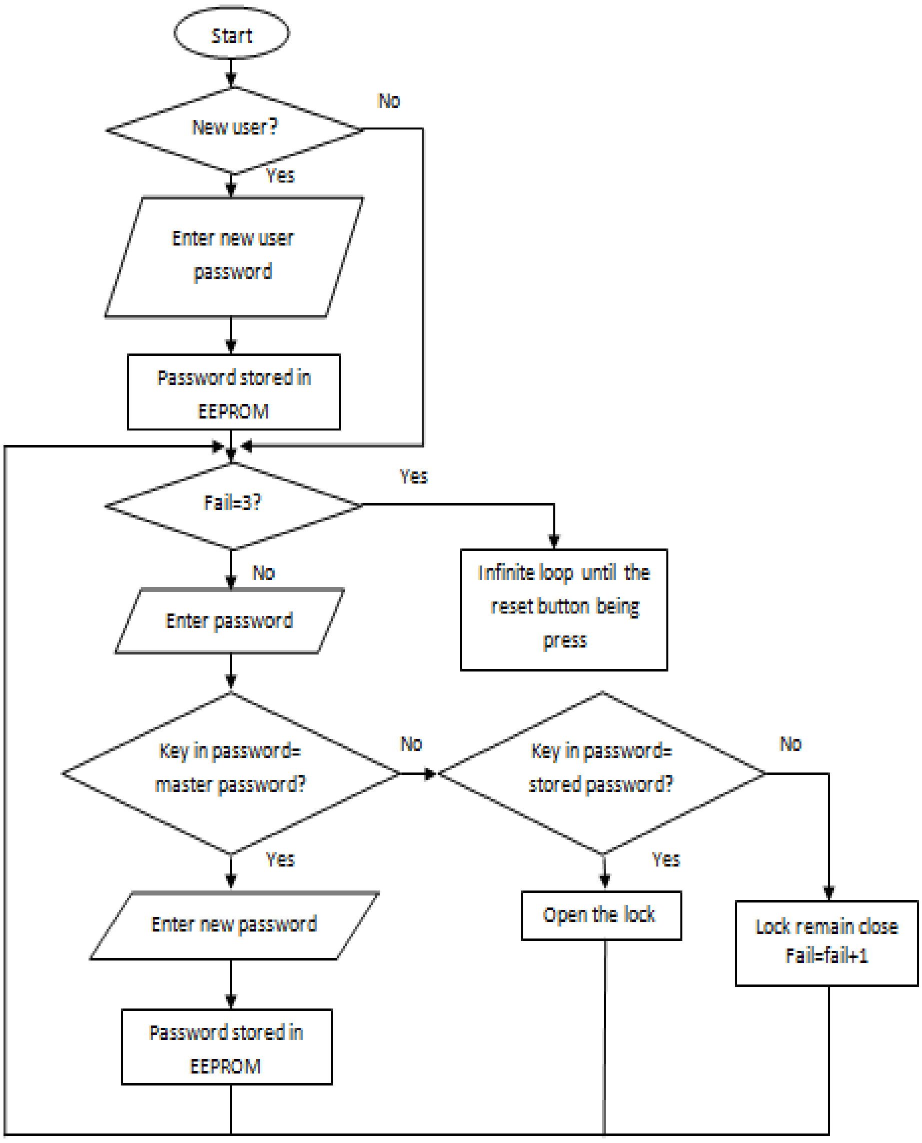

Start
New user?
No
Yes
Enter new user password
Password stored in EEPROM
Fail=3?
Yes
No
Infinite loop until the reset button being press
Enter password
Key in password= master password?
No
Key in password= stored password?
No
Yes
Yes
Enter new password
Open the lock
Lock remain close Fail=fail+1
Password stored in EEPROM

Figure 3.7 Programming Flow Chart for the System

3.4.2 Keypad Scanning Algorithm

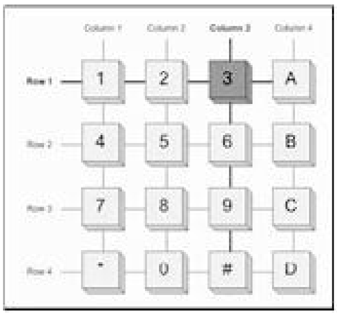
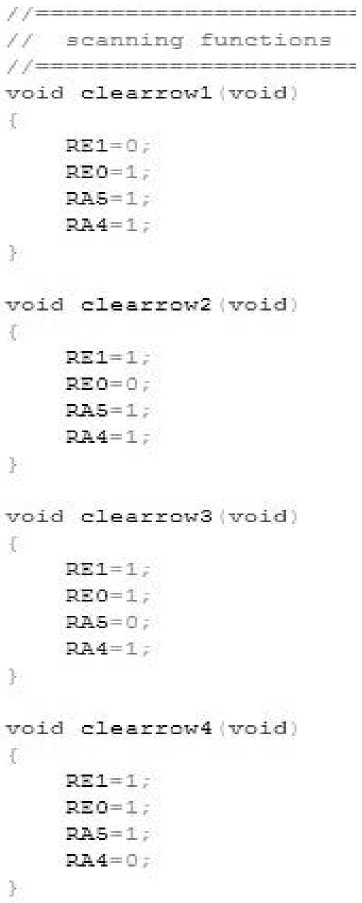
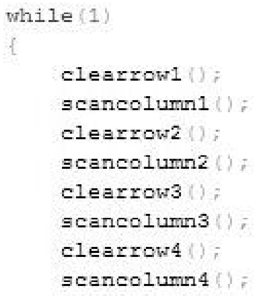


Figure 3.8 Connections of the 4x4 Keypad Pins with PIC16F877A



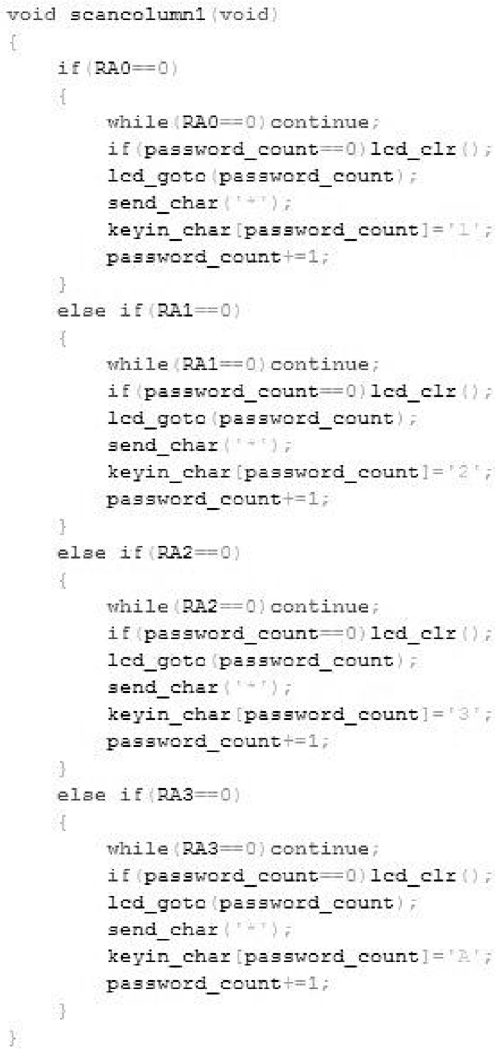
```
//======================
//  scanning functions
//======================
void clearrow1(void)
{
    RE1=0;
    RE0=1;
    RA5=1;
    RA4=1;
}

void clearrow2(void)
{
    RE1=1;
    RE0=0;
    RA5=1;
    RA4=1;
}

void clearrow3(void)
{
    RE1=1;
    RE0=1;
    RA5=0;
    RA4=1;
}

void clearrow4(void)
{
    RE1=1;
    RE0=1;
    RA5=1;
    RA4=0;
}
```

Figure 3.9 Clear Row Function



```
while(1)
{
    clearrow1();
    scancolumn1();
    clearrow2();
    scancolumn2();
    clearrow3();
    scancolumn3();
    clearrow4();
    scancolumn4();
```

Figure 3.10 Keypad Scanning Algorithm



```
void scancolumn1(void)
{
    if(RA0==0)
    {
        while(RA0==0)continue;
        if(password_count==0)lcd_clr();
        lcd_goto(password_count);
        send_char('*');
        keyin_char[password_count]='1';
        password_count+=1;
    }
    else if(RA1==0)
    {
        while(RA1==0)continue;
        if(password_count==0)lcd_clr();
        lcd_goto(password_count);
        send_char('*');
        keyin_char[password_count]='2';
        password_count+=1;
    }
    else if(RA2==0)
    {
        while(RA2==0)continue;
        if(password_count==0)lcd_clr();
        lcd_goto(password_count);
        send_char('*');
        keyin_char[password_count]='3';
        password_count+=1;
    }
    else if(RA3==0)
    {
        while(RA3==0)continue;
        if(password_count==0)lcd_clr();
        lcd_goto(password_count);
        send_char('*');
        keyin_char[password_count]='A';
        password_count+=1;
    }
}
```

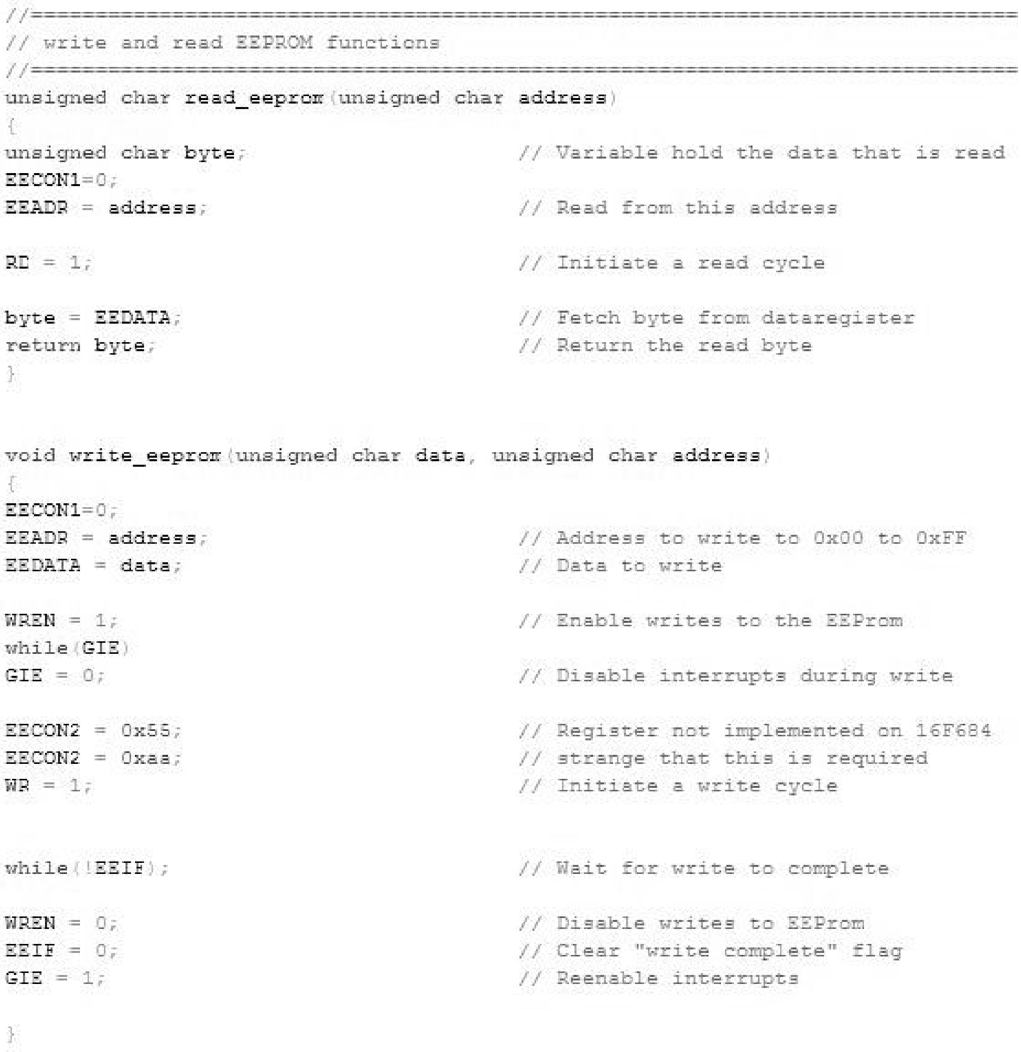
Figure 3.11 Scan Column Function

Figure 3.8 shows the connection of the 4x4 keypad pins with PIC16F877A. For the case when number ‘3’of the keypad being pressed, the 2 pin RA2 and RE1 will be shorted. So, the keypad pin has been divided into 2 group where 4 pins are the input and 4 pins are the output of PIC for decoding the number from keypad without using the decoder. Based on the Figure 3.8, the input pin are set to RA0-RA3 while RA4, RA5, RE0 and RE1 are set to the output pin.

Figure 3.10 shows a simple keypad scanning algorithm. When the user entering the 6 digit password for getting access to the restricted area, the program will scan row by row and column as well which can be showed in the Figure3.9 and 3.11. The scanning algorithm consists of several steps. First of all, the output pin RE1 will be cleared and set the rest of the output pin. Then, the system will go through a ‘scancolumn1’ function where to detect which number of the first row that has been pressed. For example, if number ‘2’ being pressed means the pin of RA1 is detect a 0. After scan for the first row, the system will proceed with the ‘scancolumn2’, ‘scancolumn3’ and ‘scancolumn4’ function in order detect the rest number such as ‘4’, ‘5’, ‘6’, until ‘D’.

3.4.3 Read/ Write EEPROM

Storing data in EEPROM is one of the methods to remain the password data when the power supply is removed. Thus, the user will use the same password to access the restricted area even though the system has been reset. Figure 3.12 shows the Read/ Write EEPROM function.



```
//==============================================================================
// write and read EEPROM functions
//==============================================================================
unsigned char read_eeprom(unsigned char address)
{
unsigned char byte;                     // Variable hold the data that is read
EECON1=0;
EEADR = address;                        // Read from this address

RD = 1;                                 // Initiate a read cycle

byte = EEDATA;                          // Fetch byte from dataregister
return byte;                            // Return the read byte
}


void write_eeprom(unsigned char data, unsigned char address)
{
EECON1=0;
EEADR = address;                        // Address to write to 0x00 to 0xFF
EEDATA = data;                          // Data to write

WREN = 1;                               // Enable writes to the EEProm
while(GIE)
GIE = 0;                                // Disable interrupts during write

EECON2 = 0x55;                          // Register not implemented on 16F684
EECON2 = 0xaa;                          // strange that this is required
WR = 1;                                 // Initiate a write cycle


while(!EEIF);                           // Wait for write to complete

WREN = 0;                               // Disable writes to EEProm
EEIF = 0;                               // Clear "write complete" flag
GIE = 1;                                // Reenable interrupts

}
```

Figure 3.12 Read/ Write EEPROM Function

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter discusses the hardware and software development, demonstration result and discussion of the project. For the hardware development, it consists of project schematic, pin configuration and circuit design. Software development will discuss about the programming process, compiling process and burning process. For the demonstration result, it will show all the result of the system condition. Lastly, problem faced in the project will be discussed in the discussion.

4.2 Hardware Development

This section will discuss about the schematic diagram of the secured room access system and the pin configuration of the design circuit.

4.2.1 Schematic

The hardware design consisted of the voltage regulator, LCD display, LED indicator, crystal, 4x4 matrix keypad, relay and buzzer. The voltage regulator functioned as a converter to convert 12V power supply to 5V which is the operating voltage of the microcontroller and LCD display.

4x4 matrix keypad worked as an input of the system where the passwords use to open the door will be enter via the keypad. LCD display, LED, buzzer and the relay will be output of the system. The LCD display functioned to show the system message to the user. For the LED indicator in the system, green LED will on when the user successful entered the correct password. On the other hand, red LED will on when the user key in the wrong password.

In addition, relay worked as a switch to energize and de-energize the magnetic lock. The buzzer use to indicate the correct password or the wrong password entered by the user. Figure 4.1 shows the schematic diagram of the secured room access system.

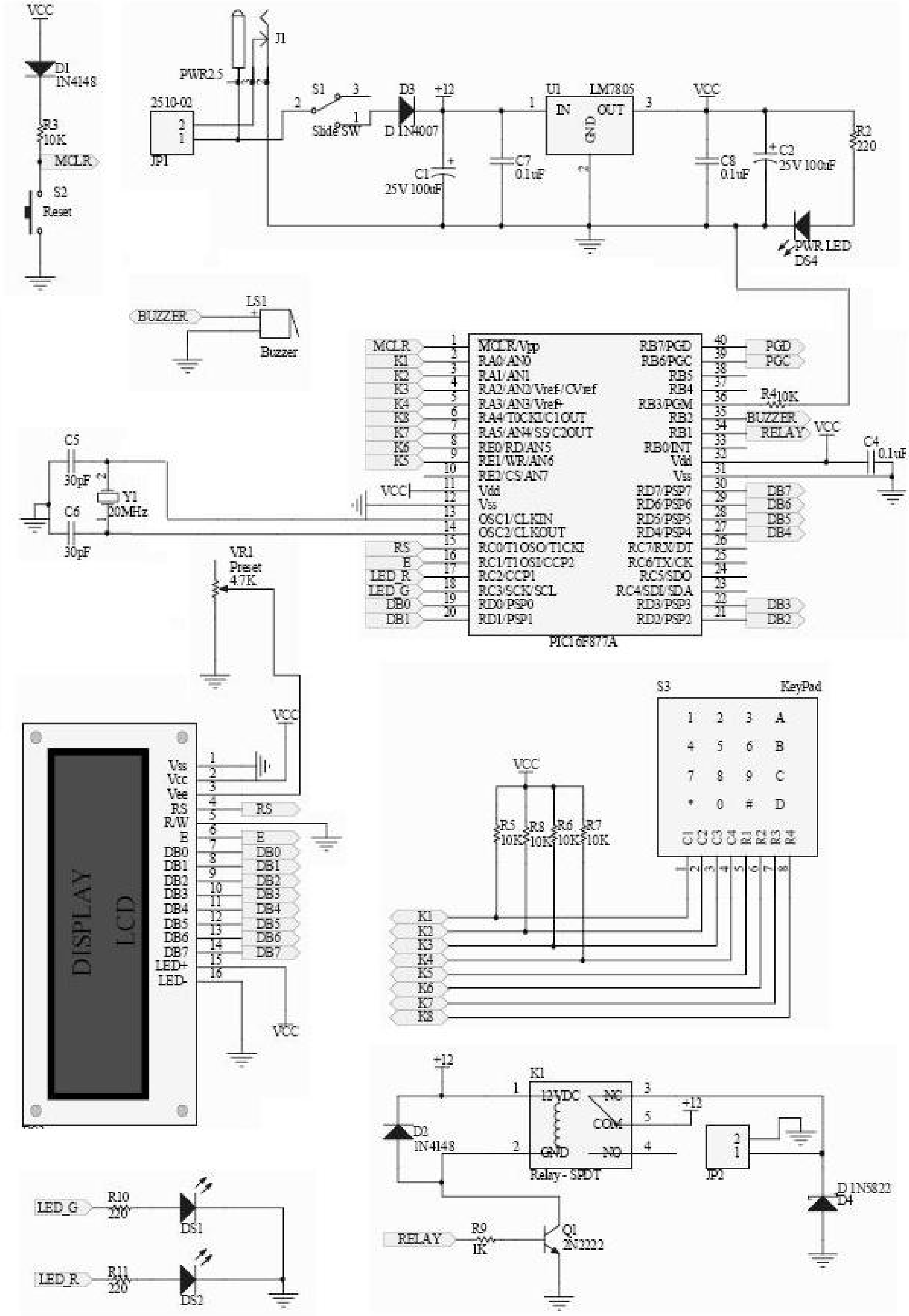


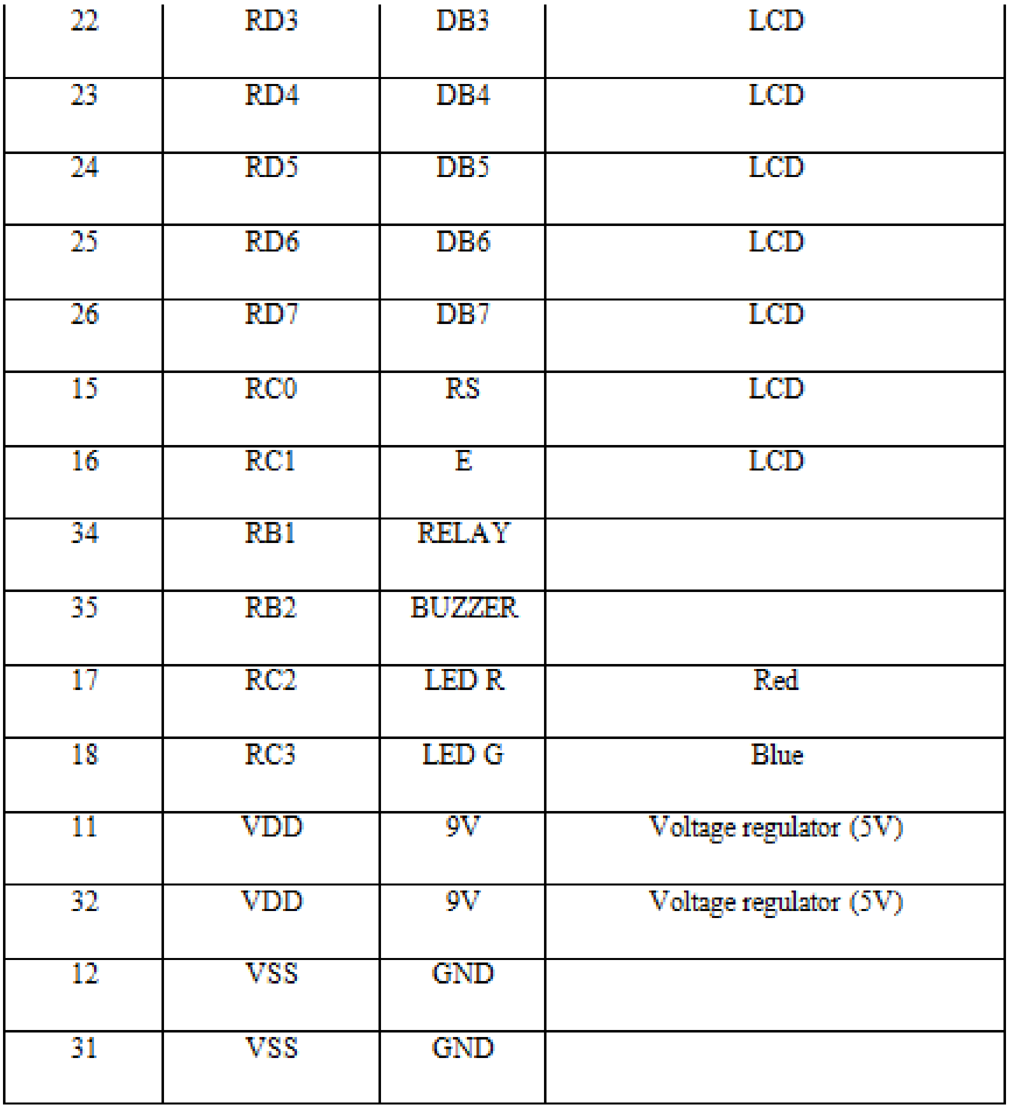
Figure 4.1 Schematic Diagram

4.2.2 Pin Configuration

The components and devices is connected according to pin configurations (Table4.2). Port A of PIC16F887 functioned as input pin from 4x4 matrix keypad. Port B, C and Port D functioned as output pin for LCD display, relay, LED indicator and buzzer. Voltage regulator was applied to regulate the supply voltage (Vss) and reference voltage (+Vref) at 5V.

Table 4.1: Pin Configuration

Pin No.	Input Pins		Description
2	RA0	K1	KEYPAD
3	RA1	K2	KEYPAD
4	RA2	K3	KEYPAD
5	RA3	K4	KEYPAD
6	RA4	K8	KEYPAD
7	RA5	K7	KEYPAD
8	RE0	K6	KEYPAD
9	RE1	K5	KEYPAD
19	RD0	DB0	LCD
20	RD1	DB1	LCD
21	RD2	DB2	LCD



22	RD3	DB3	LCD
23	RD4	DB4	LCD
24	RD5	DB5	LCD
25	RD6	DB6	LCD
26	RD7	DB7	LCD
15	RC0	RS	LCD
16	RC1	E	LCD
34	RB1	RELAY	
35	RB2	BUZZER	
17	RC2	LED R	Red
18	RC3	LED G	Blue
11	VDD	9V	Voltage regulator (5V)
32	VDD	9V	Voltage regulator (5V)
12	VSS	GND	
31	VSS	GND	

4.2.3 Circuit Design

The Circuit design was tested on the proto board to check the functionality of circuit design. Figure 4.2 shows the Circuit Design on Proto Board.

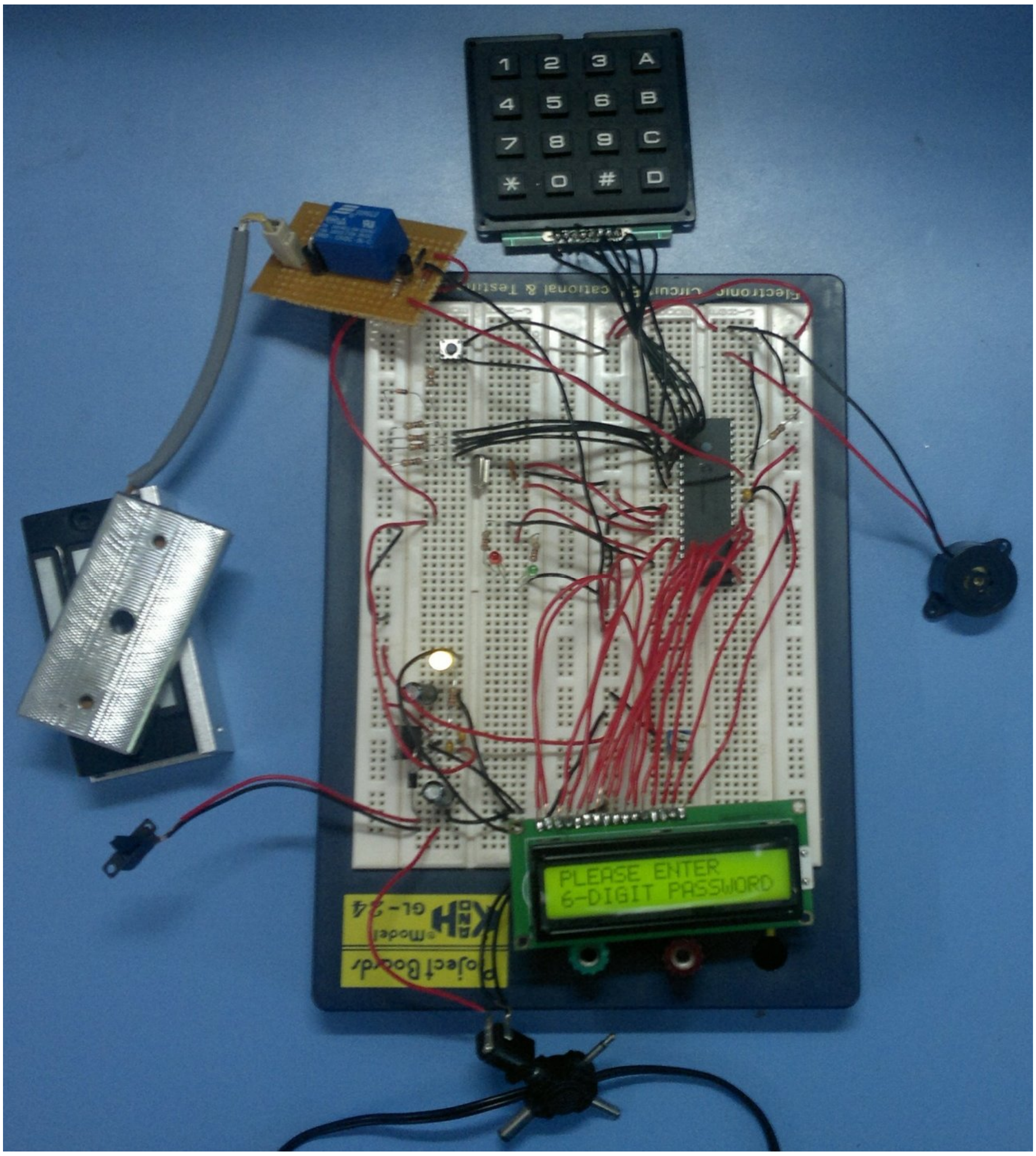


Figure 4.2 Circuit Design on Proto Board

4.3 Software Development

Software or programming is the one of the important part for device to control interface of tools module in order to produce an intelligence devices. The programming was the instructions allowed the microcontroller executed the system automatically and generated the required output. The programming is written using C language because it is easy to edit and control the bit and register in microcontroller. The IDE used was MPLAB version 8.30 and the programmer used was PICkit2 version 2.40. The programming scripts consists of PIC16F887A devices configurations, I/O pins configurations, and LCD display output configuration.

4.3.1 Programming Process

The programming has been written using C language. There are two step involve in the software programming process. Firstly, the programming code was compiled using the compiler into the hex file. Then, the hex file was burned into PIC16F887A using the burner. Figure 4.3 and 4.4 shows the process of writing programming and burner programming and the flow chart of the programming process.

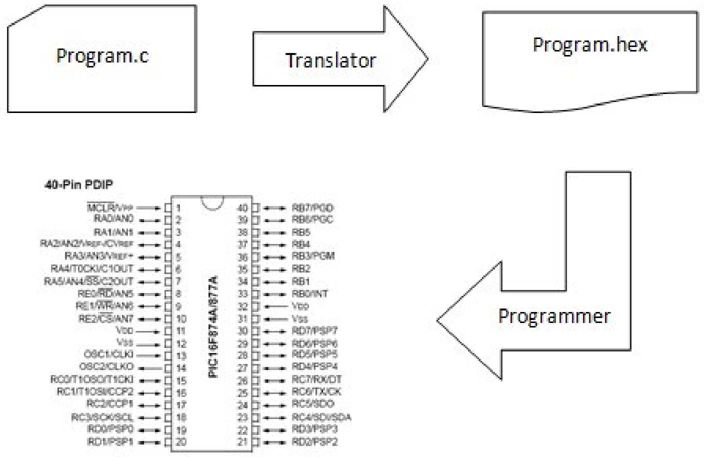


Figure 4.3 Programming Process

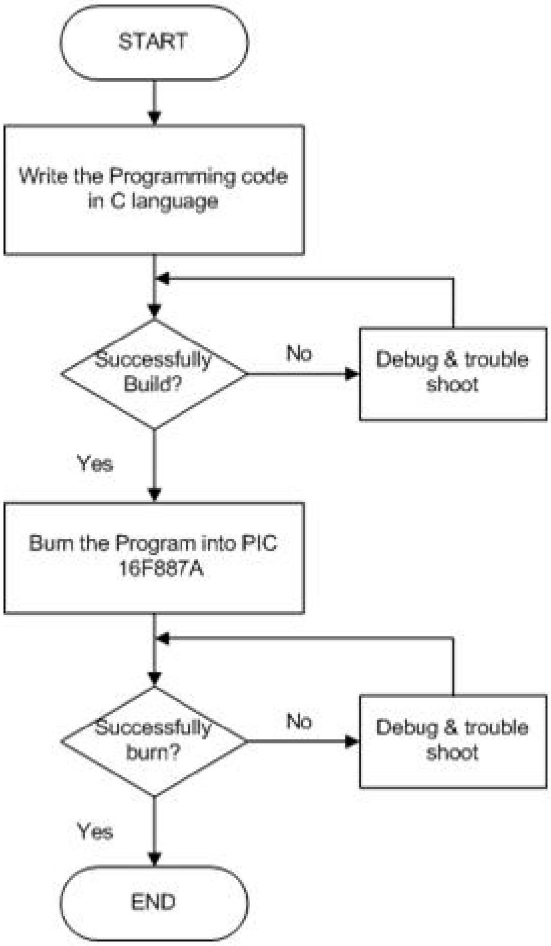
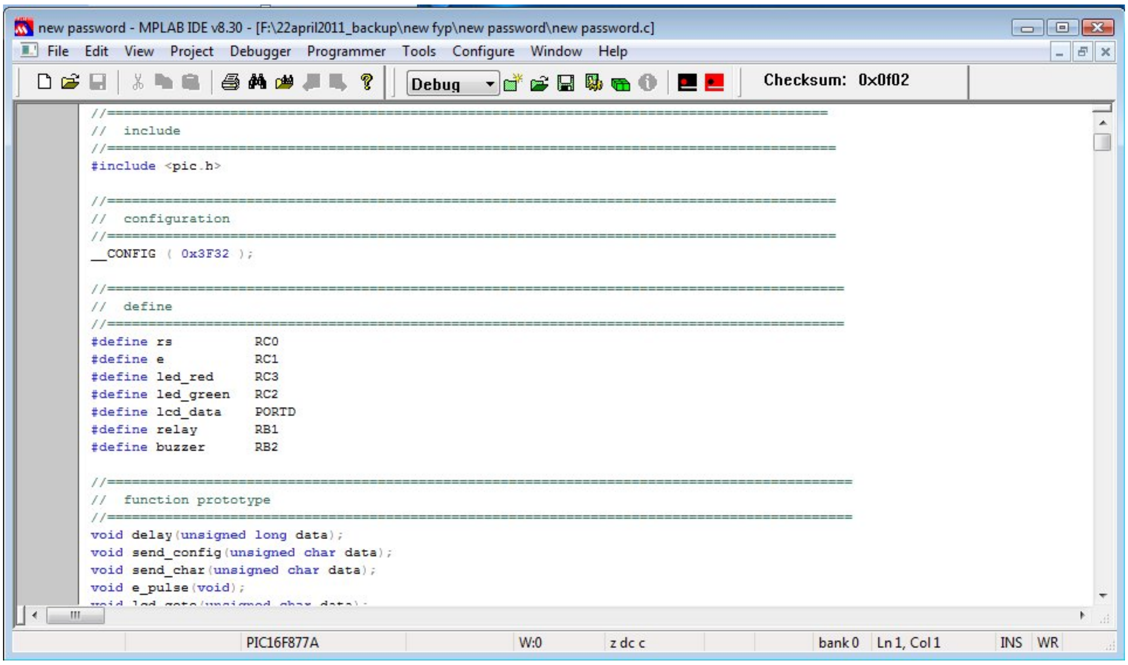


Figure 4.4 Flow Chart of PIC Programming

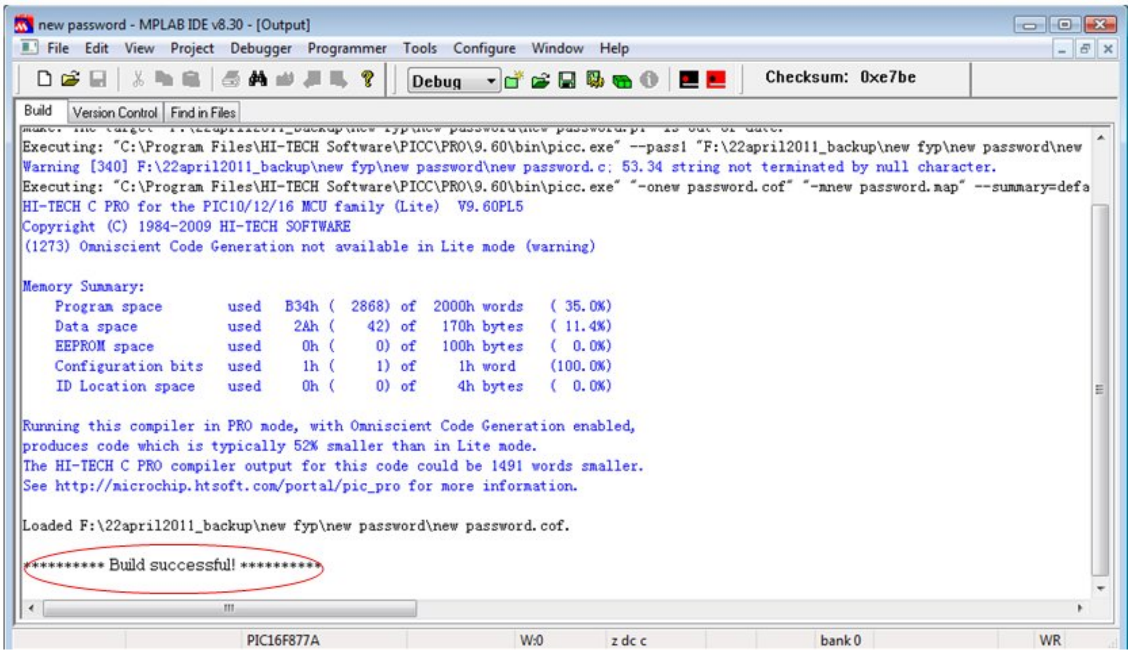
4.3.2 Compiling Process

MPLAB IDE v8.30 was used to compile the C language programming code to the hex file. The programming script (refer to appendix) successfully compiled shows in the figure 4.6.



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

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

//=====================================================================================
//  define
//=====================================================================================
#define rs          RC0
#define e           RC1
#define led_red     RC3
#define led_green   RC2
#define lcd_data    PORTD
#define relay       RB1
#define buzzer      RB2

//=====================================================================================
//  function prototype
//=====================================================================================
void delay(unsigned long data);
void send_config(unsigned char data);
void send_char(unsigned char data);
void e_pulse(void);
void lcd_goto(unsigned char data);
```

Figure 4.5 MPLAB IDE Software Interface

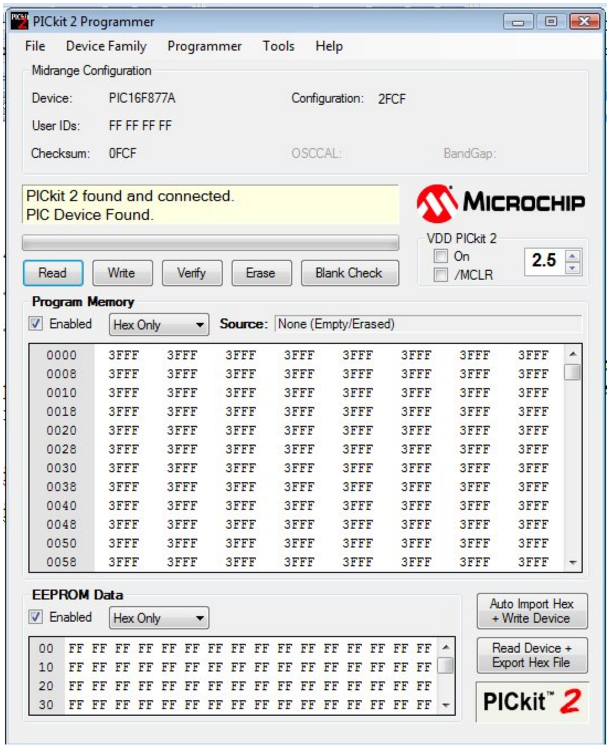

new password - MPLAB IDE v8.30 - [Output]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug
Checksum: 0xe7be
Build
Version Control
Find in Files
PIC16F877A
W:0
z dc c
bank 0
WR

```
make: The target "F:\22april2011_backup\new fyp\new password\new password.p1" is out of date.
Executing: "C:\Program Files\HI-TECH Software\PICC\PRO\9.60\bin\picc.exe" --pass1 "F:\22april2011_backup\new fyp\new password\new
Warning [340] F:\22april2011_backup\new fyp\new password\new password.c: 53.34 string not terminated by null character.
Executing: "C:\Program Files\HI-TECH Software\PICC\PRO\9.60\bin\picc.exe" "-onew password.cof" "-mnew password.map" --summary=defa
HI-TECH C PRO for the PIC10/12/16 MCU family (Lite)  V9.60PL5
Copyright (C) 1984-2009 HI-TECH SOFTWARE
(1273) Omniscient Code Generation not available in Lite mode (warning)

Memory Summary:
    Program space        used   B34h (  2868) of  2000h words   ( 35.0%)
    Data space           used    2Ah (    42) of   170h bytes   ( 11.4%)
    EEPROM space         used     0h (     0) of   100h bytes   (  0.0%)
    Configuration bits   used     1h (     1) of     1h word    (100.0%)
    ID Location space    used     0h (     0) of     4h bytes   (  0.0%)

Running this compiler in PRO mode, with Omniscient Code Generation enabled,
produces code which is typically 52% smaller than in Lite mode.
The HI-TECH C PRO compiler output for this code could be 1491 words smaller.
See http://microchip.htsoft.com/portal/pic_pro for more information.

Loaded F:\22april2011_backup\new fyp\new password\new password.cof.

********** Build successful! **********
```

Figure 4.6 Compiler's Output

4.3.3 Burning Process

Burner software uses for write the programming into microcontroller. In this project the programmer used was PICkit2 version 2.40. There are several step have to follow in order to burn the program into the microcontroller. Firstly, load the desired hex file into the programmer. Then, write the hex file into the microcontroller. Lastly, the program has to verify before remove the microcontroller from the programmer.



Midrange Configuration

Device: PIC16F877A

Configuration: 2FCF

User IDs: FF FF FF FF

Checksum: 0FCF

OSCCAL:

BandGap:

PICkit 2 found and connected.
PIC Device Found.

VDD PICkit 2 □ On □ /MCLR 2.5

Program Memory

☑ Enabled Hex Only **Source:** None (Empty/Erased)

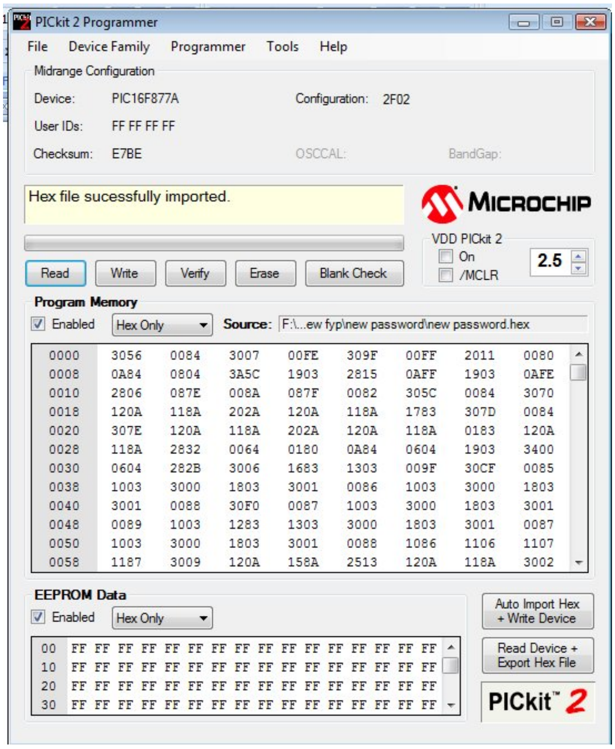
```
0000   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0008   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0010   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0018   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0020   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0028   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0030   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0038   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0040   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0048   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0050   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0058   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
```

EEPROM Data

☑ Enabled Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.7 PICkit 2 Programmer Interface



Midrange Configuration

Device: PIC16F877A

Configuration: 2F02

User IDs: FF FF FF FF

Checksum: E7BE

OSCCAL:

BandGap:

Hex file sucessfully imported.

VDD PICkit 2 □ On □ /MCLR 2.5

Program Memory

☑ Enabled Hex Only **Source:** F:\...ew fyp\new password\new password.hex

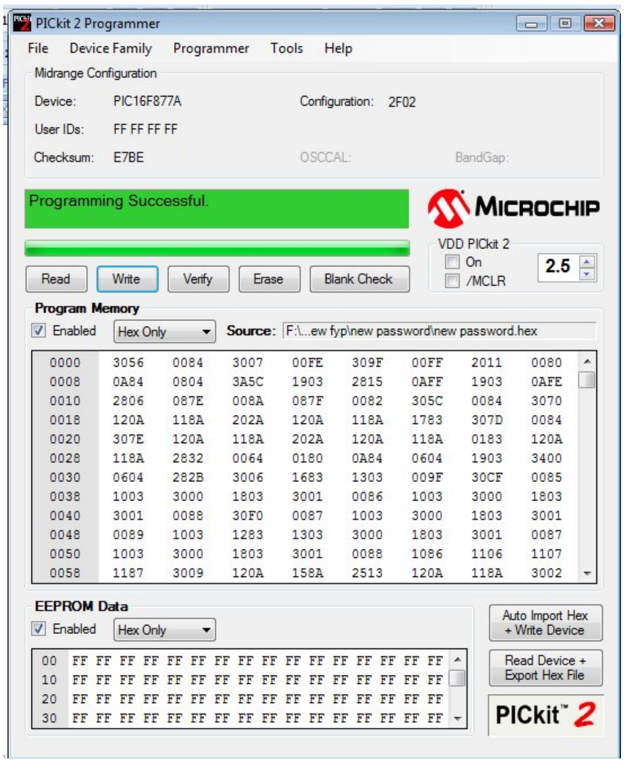
```
0000   3056  0084  3007  00FE  309F  00FF  2011  0080
0008   0A84  0804  3A5C  1903  2815  0AFF  1903  0AFE
0010   2806  087E  008A  087F  0082  305C  0084  3070
0018   120A  118A  202A  120A  118A  1783  307D  0084
0020   307E  120A  118A  202A  120A  118A  0183  120A
0028   118A  2832  0064  0180  0A84  0604  1903  3400
0030   0604  282B  3006  1683  1303  009F  30CF  0085
0038   1003  3000  1803  3001  0086  1003  3000  1803
0040   3001  0088  30F0  0087  1003  3000  1803  3001
0048   0089  1003  1283  1303  3000  1803  3001  0087
0050   1003  3000  1803  3001  0088  1086  1106  1107
0058   1187  3009  120A  158A  2513  120A  118A  3002
```

EEPROM Data

☑ Enabled Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.8 Hex file Successfully Loaded into the Programmer



Midrange Configuration

Device: PIC16F877A

Configuration: 2F02

User IDs: FF FF FF FF

Checksum: E7BE

OSCCAL:

BandGap:

Programming Successful.

VDD PICkit 2

□ On

□ /MCLR

2.5

Program Memory

☑ Enabled Hex Only

Source: F:\...ew fyp\new password\new password.hex

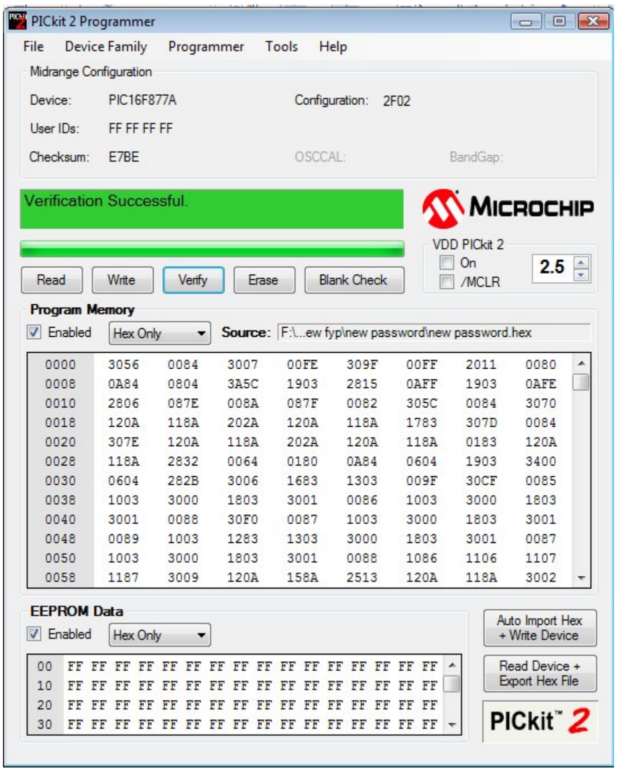
```
0000   3056  0084  3007  00FE  309F  00FF  2011  0080
0008   0A84  0804  3A5C  1903  2815  0AFF  1903  0AFE
0010   2806  087E  008A  087F  0082  305C  0084  3070
0018   120A  118A  202A  120A  118A  1783  307D  0084
0020   307E  120A  118A  202A  120A  118A  0183  120A
0028   118A  2832  0064  0180  0A84  0604  1903  3400
0030   0604  282B  3006  1683  1303  009F  30CF  0085
0038   1003  3000  1803  3001  0086  1003  3000  1803
0040   3001  0088  30F0  0087  1003  3000  1803  3001
0048   0089  1003  1283  1303  3000  1803  3001  0087
0050   1003  3000  1803  3001  0088  1086  1106  1107
0058   1187  3009  120A  158A  2513  120A  118A  3002
```

EEPROM Data

☑ Enabled Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.9 Programming Successful



Midrange Configuration

Device: PIC16F877A

Configuration: 2F02

User IDs: FF FF FF FF

Checksum: E7BE

OSCCAL:

BandGap:

Verification Successful.

VDD PICkit 2

□ On

□ /MCLR

2.5

Program Memory

☑ Enabled Hex Only

Source: F:\...ew fyp\new password\new password.hex

```
0000   3056  0084  3007  00FE  309F  00FF  2011  0080
0008   0A84  0804  3A5C  1903  2815  0AFF  1903  0AFE
0010   2806  087E  008A  087F  0082  305C  0084  3070
0018   120A  118A  202A  120A  118A  1783  307D  0084
0020   307E  120A  118A  202A  120A  118A  0183  120A
0028   118A  2832  0064  0180  0A84  0604  1903  3400
0030   0604  282B  3006  1683  1303  009F  30CF  0085
0038   1003  3000  1803  3001  0086  1003  3000  1803
0040   3001  0088  30F0  0087  1003  3000  1803  3001
0048   0089  1003  1283  1303  3000  1803  3001  0087
0050   1003  3000  1803  3001  0088  1086  1106  1107
0058   1187  3009  120A  158A  2513  120A  118A  3002
```

EEPROM Data

☑ Enabled Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.10 Verification Successful

4.4 Demonstration Result

Design Demonstration was done by using a proto board. The secured room access system can be divided into 5 condition and show on the result as below. First of all, the user will be asked to set the new password in the EEPROM of the microcontroller. Figure 4.11 shows the result of the device in power off mode. Figure 4.12 shows the result of new user condition.

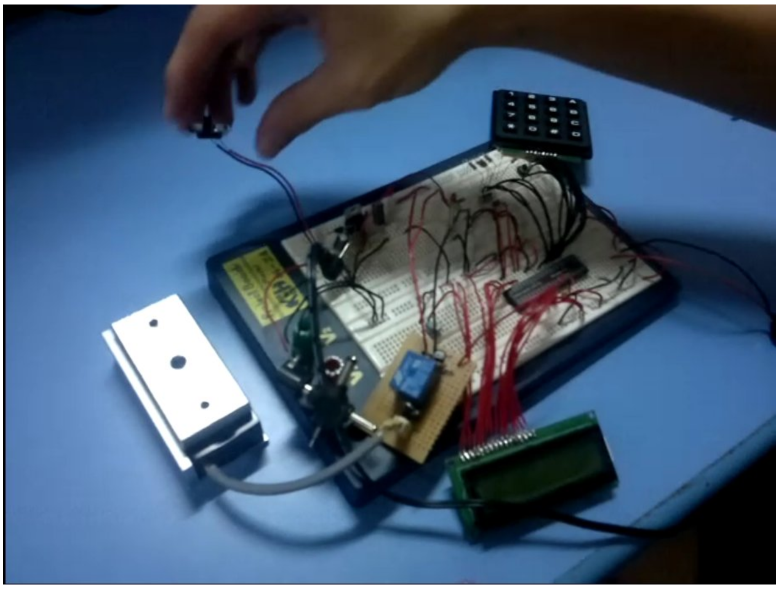


Figure 4.11 Device in Power Off Condition

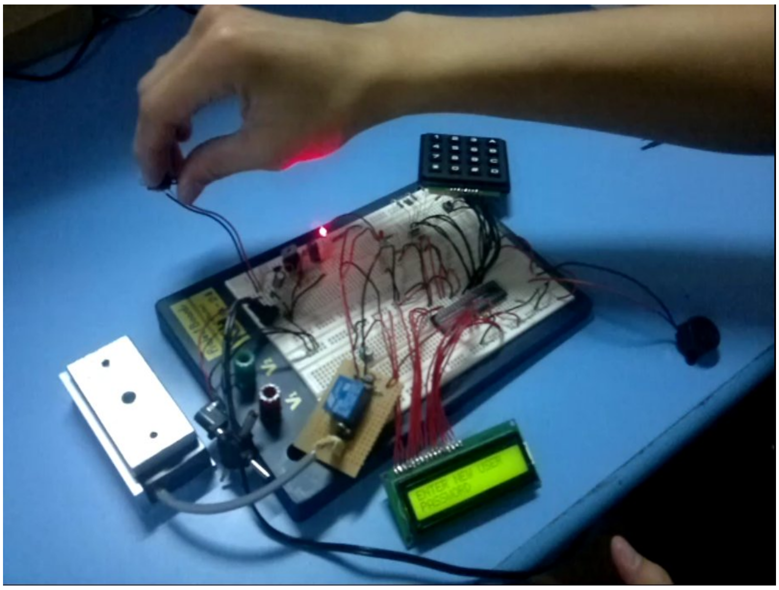


Figure 4.12 New User Condition

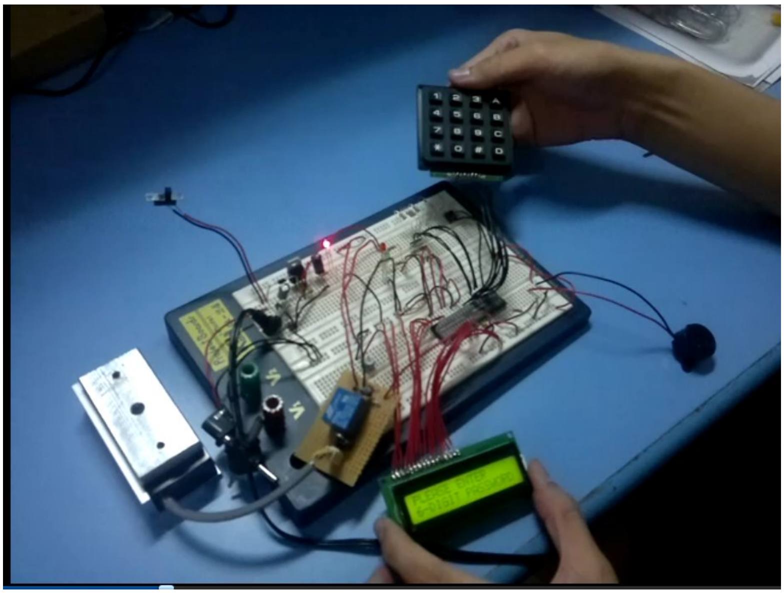


Figure 4.13 Asking for Password

For the correct password entered condition, the 'SUCCESS' message will display on the LCD display, green LED on and the magnetic lock will be open. Result shown as the figure 4.14 and 4.15 below.

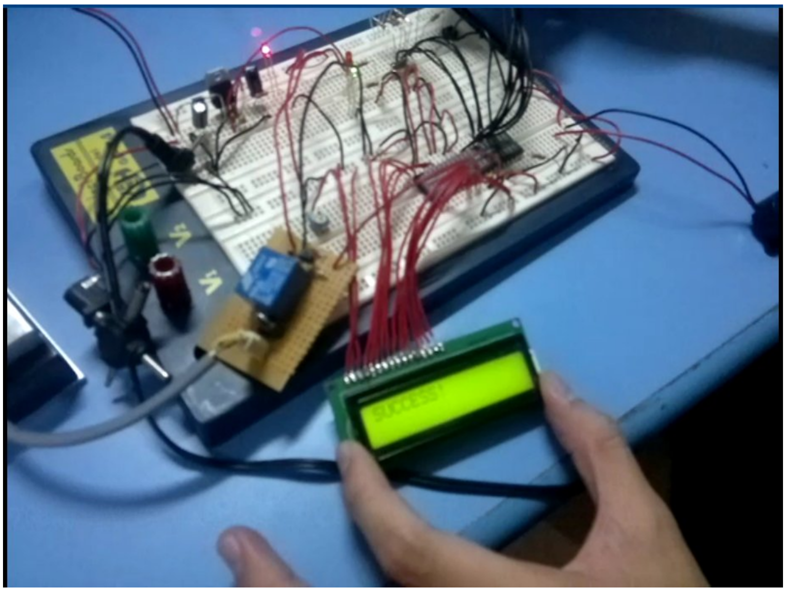


Figure 4.14 Correct Password Condition

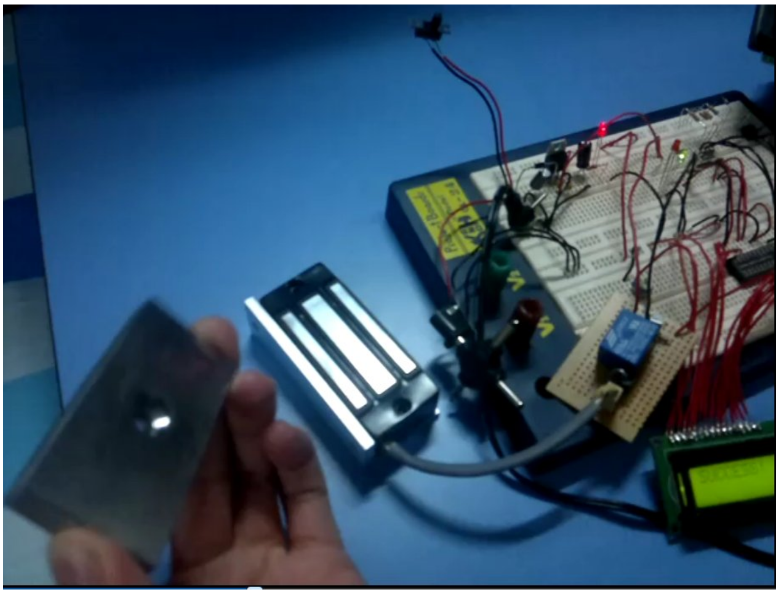


Figure 4.15 Magnetic Lock Open

For the wrong password entered condition, the 'Error' message will display on the LCD display, red LED on and the magnetic lock will remain close. Result shown as the figure 4.16.

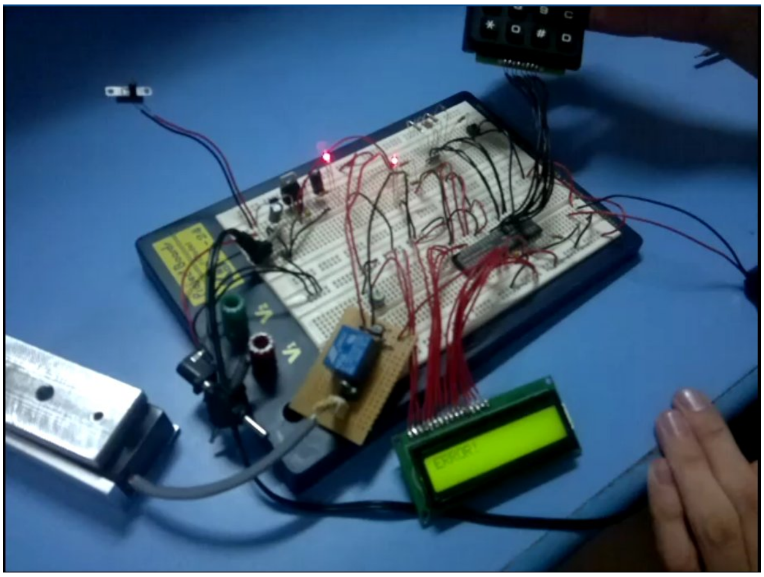


Figure 4.16 Wrong Password Condition

For the 3 times wrong password entered condition, system will go though the infinite looping process, 'CONTACT SYSTEM ADMINISTRATOR' message will display on the LCD display. The system can be reset by pressing the reset button.

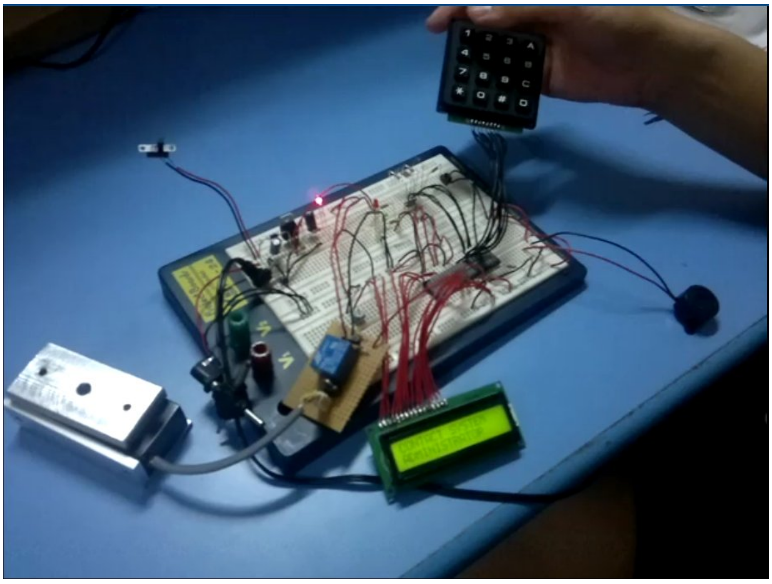


Figure 4.17 Three Times Wrong Password Condition

For the entered password same with the master password condition, system will as for entering the new password, ‘ENTER NEW PASSWORD’ message will display on the LCD display. After the password has been entered, ‘PASSWORD CHANGED’ message will display on the LCD display.

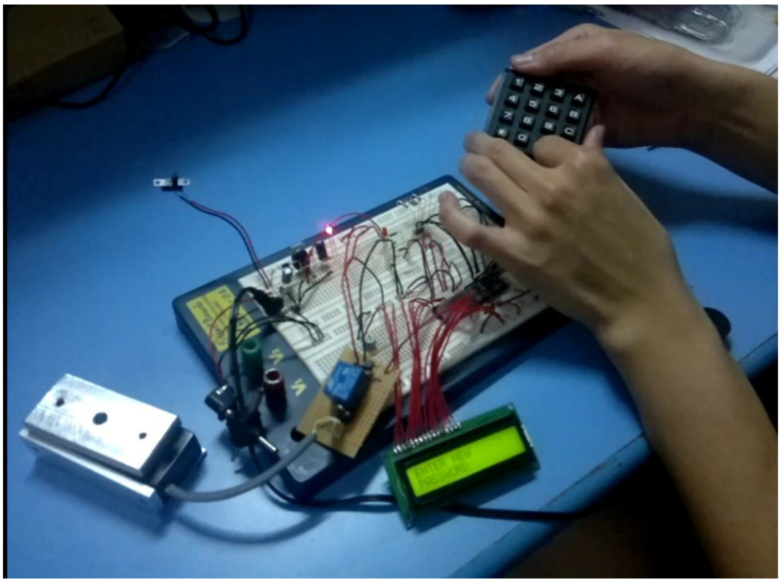


Figure 4.18 Master Password Condition

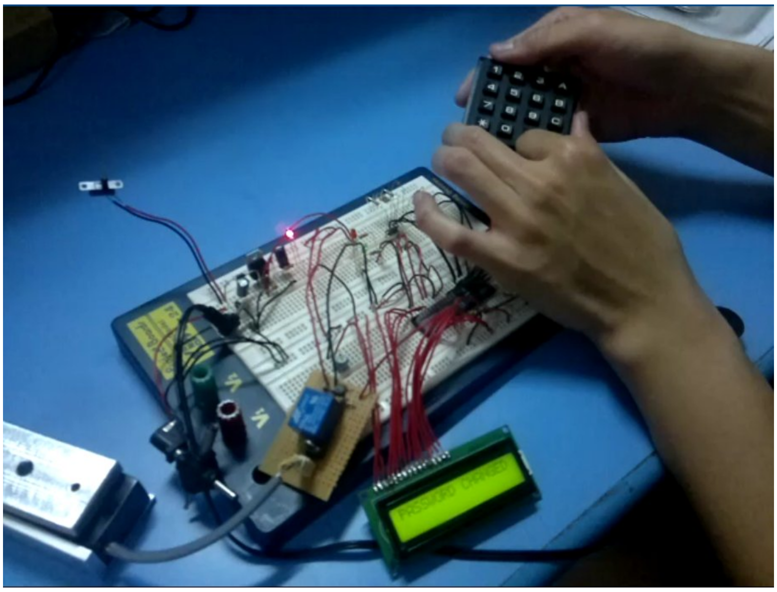


Figure 4.19 Password Changed

4.5 Discussion

There are several problem has been faced when producing the secured room access system. Initially, lack of idea for programming of keypad scanning is the main problem causing the project design period become longer. Not only that, there are some circuit design problem directly causing to the longer project design period such as the voltage regulator and relay circuit.

Last but not the least, programming are aslo one of the main problem contribute to the longer project design period. For example, how to the write or read data from the EEPROM of the microcontroller.

Based on the hardware and software development result, the problem faced in the project were solved and the project was completed succefully with the desired project scope.

CHAPTER 5

CONCLUSION

5.1 Introduction

This chapter discusses about the conclusion, commercial potential and some future work for the project. The future works include the use of RFID, biometric authentication and add some function to the system.

5.2 Conclusion

Electrical and electronic engineering knowledge has been applied to the project. It provides practice to design a system or component within a defined specification. Problem solving skill improved in hardware and software development. Based on the results obtained, the project's objectives are archived within the defined scopes.

This system given some value added to the security system and more users friendly. The main advantage of this project, the user can change the store password by entering the master password. For example, the new user of the room can ask for the master password from the supervisor or refer to the user manual in order to change the password to his or her own password instead of using the previous user password which is not secured for the new user.

The system development divided into software and hardware development which are MPLAB IDE software have been using to write the programming and PICkit 2 v2.40 for tools of burning. The coding used in this hardware has been written using C language.

A system allows only authorized personnel access to secure areas has been developed to maintain the security easily and efficiently. The expected outcomes of the project are successfully archived. High accuracy and reliability of the result is shown based on the result obtained. As a conclusion, the project is successfully accomplished and fulfilled all the expected outcomes.

5.3 Commercial Potential

This product is easy to install where the product only consist of the magnetic lock, keypad module and LCD module. This product also user-friendly where it is easy to use - just enter the password and the user can access to the restricted area. Another feature of this product is the door is automatically relocks after 10 seconds. This product is low cost system hence has low price that can be afforded by anyone. The power requirement is low compared to the other high technology security system. This system can be used for residents, industrial premises or just to be a security box.

5.4 Future Work

The performance of the system still can be increases so that this system will be more up-to-date and more flexible. For future works, some recommendations have been listed in order to improve the system performance.

i. Used RFID for more convenience input recognition [9]

Radio-frequency identification (RFID) is the use of an object (typically referred to as an RFID tag) applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. Some tags can be read from several meters away and beyond the line of sight of the reader. By using RFID technology, this project can be used to implement a quick and convenient access system at a certain area or department.

ii. Used biometric authentication (fingerprint authentication)

Fingerprint has been used as identifications for individuals since late 19th century and it has been discovered that every individual has different fingerprints even for identical twins. Fingerprints have the properties of distinctiveness or individuality, and the fingerprints of a particular person remain almost the same (persistence) over time. These properties make fingerprints suitable for biometric uses.

iii. Add more display message function on the LCD display

The system can be improved by adding some display message function in the design where the user can choose to display some information telling the visitor about their available time or let the visitor leaving some message for the room owner.

REFERENCES

[1] Jufri bin Md Rus, “Coded Security Door Employing 16F84 Microcontroller With Infrared Remote Control Features”, Faculty of Electrical University Technology Malaysia, November 2006

[2] Richard Barnett, Larry O’Cull, Sarah Cox, “Embedded C Programming and the Microchip PIC”,Thomson Delmar Learning, Canada, 2005

[3] John Iovine, “PIC microcontroller project book”, McGraw-Hill, New York, 2000

[4] PIC 16F877A Manual Datasheet, Microchip Technology Inc. *www.microchip.com*

[5] http://en.wikipedia.org/wiki/Combination_lock. Retrieved on 18 September, 2010

[6] Afzahaniff Hussin, “Security and Control System”, Faculty of Electrical University Technology Malaysia, May 2008

[7] http://en.wikipedia.org/wiki/PIC_microcontroller. Retrieved on 25 September, 2010

[8] Anshuman Bezborah, “PIC Microcontroller-Based Electronic Lock”, Electronic For You, June 2010

[9] Azwarina Binti Azizdin, “Access Control System”, Faculty of Electrical University Technology Malaysia, April 2010

[10] http://en.wikipedia.org/wiki/Keypad. Retrieved on 11 October, 2010

[11] http://www.spikenzielabs.com/SpikenzieLabs/LCD_How_To.html. Retrieved on 4 February, 2011

[12] http://en.wikipedia.org/wiki/Lock_(device). Retrieved on 14 February, 2011

APPENDICES

Appendix A: Programming Coding

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

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

//============================================================================================
//  define
//============================================================================================
#define rs          RC0
#define e           RC1
#define led_red     RC3
#define led_green   RC2
#define lcd_data    PORTD
#define relay       RB1
#define buzzer      RB2

//=============================================================================================
//  function prototype
//=============================================================================================
void delay(unsigned long data);
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);
void beep_once(void);
void beep_twice(void);
void beep_sms(void);
unsigned char read_eeprom(unsigned char address);
void write_eeprom(unsigned char data, unsigned char address);

//==============================================================================================
//  global variable
//==============================================================================================
unsigned char password_count=0;
unsigned char id_value[6];
unsigned char keyin_char[6];              // Declare an array to stall the 6-digit key in    ↙
    password
unsigned char pass_store[6];              // Stall the 6-digit desired password
unsigned char master_password[6]="654321";
unsigned char a;
unsigned char fail=0;


//==============================================================================================
//  main function
//==============================================================================================
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=0b11110000;   //set bit4-7 portC as input(connected to 4 row of keypad)
    TRISE=0b00000000;   //set portE as output

    PORTC=0;
    PORTD=0;
    relay=0;
    buzzer=0;
    led_green=0;
    led_red=0;

    send_config(0b00001001);    //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

//=============================================================================================↙
    ===
//  setting new user password
//=============================================================================================↙
    ====
    a=read_eeprom(0x00);                        //read the data from eeprom address 0x00
    if(a==0xff)
    {
        led_green=0;
        led_red=0;
        lcd_clr();                              //clear LCD
        delay(1000);
        lcd_goto(0);                            //initial display
        send_string("ENTER NEW USER");         //Display "ENTER NEW USER" on lcd
        lcd_goto(20);                           //Display on 2nd line
        send_string("PASSWORD");               //Display "PASSWORD" on lcd
        while(1)
        {                                       //keypad scanning algorithm
            clearrow1();                        //Clear 1st output pin and set the others
            scancolumn1();                      //scan column 1-4
            clearrow2();                        //Clear 2nd output pin and set the others
            scancolumn2();                      //scan column 1-4
            clearrow3();                        //Clear 3rd output pin and set the others
            scancolumn3();                      //scan column 1-4
            clearrow4();                        //Clear 4th output pin and set the others
            scancolumn4();                      //scan column 1-4

            if(password_count==6)
            {
                password_count=0;
                pass_store[0]=keyin_char[0];
                pass_store[1]=keyin_char[1];
                pass_store[2]=keyin_char[2];
                pass_store[3]=keyin_char[3];
                pass_store[4]=keyin_char[4];
                pass_store[5]=keyin_char[5];

                write_eeprom(pass_store[0],0x00);
                write_eeprom(pass_store[1],0x01);
                write_eeprom(pass_store[2],0x02);
                write_eeprom(pass_store[3],0x03);
                write_eeprom(pass_store[4],0x04);
                write_eeprom(pass_store[5],0x05);
                lcd_clr();
                lcd_goto(0);
                send_string("PASSWORD SET");
                beep_once();
                delay(500000);

                goto label1;
            }
        }
    }
    else
    {
        goto label1;
    }
//=================================================================================


label1: led_green=0;                            //label the point of repeatation
        led_red=0;
        relay=0;
        lcd_clr();
        lcd_goto(0);                            //initial display
        send_string("PLEASE ENTER");            //Display "PLEASE ENTER" on lcd
        lcd_goto(20);                           //Display on 2nd line
        send_string("6-DIGIT PASSWORD");        //Display "6-DIGIT PASSWORD" on lcd

        if(fail==3)
        {
            lcd_clr();
            lcd_goto(0);                        //initial display
            send_string("CONTACT SYSTEM");      //Display "THE DEVICE WILL" on lcd
            lcd_goto(20);                       //Display on 2nd line
            send_string("ADMINISTRATOR");   //Display "EXPLODE IN 5 MIN" on lcd
            while(1);                           //infinity loop
        }

        while(1)
        {                                       //keypad scanning algorithm
            clearrow1();                        //Clear 1st output pin and set the others
            scancolumn1();                      //scan column 1-4
            clearrow2();                        //Clear 2nd output pin and set the others
            scancolumn2();                      //scan column 1-4
            clearrow3();                        //Clear 3rd output pin and set the others
            scancolumn3();                      //scan column 1-4
            clearrow4();                        //Clear 4th output pin and set the others
            scancolumn4();                      //scan column 1-4


            if(password_count==6)
            {
                password_count=0;
                pass_store[0]=read_eeprom (0x00);   //read the store password from the eeprom
                pass_store[1]=read_eeprom (0x01);
                pass_store[2]=read_eeprom (0x02);
                pass_store[3]=read_eeprom (0x03);
                pass_store[4]=read_eeprom (0x04);
                pass_store[5]=read_eeprom (0x05);

                if( (keyin_char[0]==master_password[0])&&(keyin_char[1]==master_password[1])&&
                    (keyin_char[2]==master_password[2])&&(keyin_char[3]==master_password[3])&& ↙
        //compare the keyin value with master password to test whether password
                    (keyin_char[4]==master_password[4])&&(keyin_char[5]==master_password[5]))  ↙
        //is master password or not
```

```
            {
                fail=0;
                lcd_clr();
                lcd_goto(0);                            //initial display
                send_string("ENTER NEW");               //Display "ENTER NEW" on lcd
                lcd_goto(20);                           //Display on 2nd line
                send_string("PASSWORD");                //Display "PASSWORD" on lcd
                while(1)
                {                                       //keypad scanning algorithm
                    clearrow1();                        //Clear 1st output pin and set the ↙
others
                    scancolumn1();                      //scan column 1-4
                    clearrow2();                        //Clear 2nd output pin and set the ↙
others
                    scancolumn2();                      //scan column
                    clearrow3();                        //Clear 3rd output pin and set the ↙
others
                    scancolumn3();                      //scan column
                    clearrow4();                        //Clear 4th output pin and set the ↙
others
                    scancolumn4();                      //scan column


                    if(password_count==6)
                    {
                        password_count=0;
                        pass_store[0]=keyin_char[0];
                        pass_store[1]=keyin_char[1];
                        pass_store[2]=keyin_char[2];
                        pass_store[3]=keyin_char[3];
                        pass_store[4]=keyin_char[4];
                        pass_store[5]=keyin_char[5];

                        id_value[0]=pass_store[0];          // varification
                        id_value[1]=pass_store[1];
                        id_value[2]=pass_store[2];
                        id_value[3]=pass_store[3];
                        id_value[4]=pass_store[4];
                        id_value[5]=pass_store[5];

                        delay(100);

                        if( (keyin_char[0]==id_value[0])&&(keyin_char[1]==id_value[1])&&
                            (keyin_char[2]==id_value[2])&&(keyin_char[3]==id_value[3])&&   ↙
    //compare the keyin value with id_value to verify whether
                            (keyin_char[4]==id_value[4])&&(keyin_char[5]==id_value[5]))    ↙
    //password is correct
                        {

                            write_eeprom(pass_store[0],0x00);       // write the new enter ↙
password to the eeprom
                            write_eeprom(pass_store[1],0x01);
                            write_eeprom(pass_store[2],0x02);
                            write_eeprom(pass_store[3],0x03);
                            write_eeprom(pass_store[4],0x04);
                            write_eeprom(pass_store[5],0x05);

                            lcd_clr();
                            lcd_goto(0);                            //initial display
                            send_string("PASSWORD CHANGED");        //Display "PASSWORD    ↙
CHANGED" on lcd
                            beep_once();
                            delay(500000);
```

```
                                goto label1;
                            }
                            else
                            {
                                lcd_clr();
                                lcd_goto(0);                            //initial display
                                send_string(″VARIFY FAILED″);           //Display ″VARIFY      ↙
    FAILED″ on lcd
                                delay(500000);
                                goto label1;
                            }

                        }
                    }
                }


                else if((keyin_char[0]==pass_store[0])&&(keyin_char[1]==pass_store[1])&&
                        (keyin_char[2]==pass_store[2])&&(keyin_char[3]==pass_store[3])&&    // ↙
    compare the keyin value with the store value to test whether
                        (keyin_char[4]==pass_store[4])&&(keyin_char[5]==pass_store[5]))     // ↙
    password is correct
                {
                    fail=0;
                    lcd_clr();                      //clear lcd
                    lcd_goto(0);                    //initial display
                    send_string(″SUCCESS!″);        //Display ″SUCCESS!″ on lcd
                    led_green=1;                    //Green light on
                    relay=1;                        //relay on
                    beep_sms();                     //beep sms
                    delay(1000000);
                    goto label1;                    //continue system with jump to label1
                }

                else
                {
                    fail=fail+1;                    //increasing number of fail password enter
                    lcd_clr();                      //clear lcd
                    lcd_goto(0);                    //initial display
                    send_string(″ERROR!″);          //Display ″ERROR!″ on lcd
                    led_red=1;                      //red light on
                    beep_twice();                   //beep once time
                    delay(500000);
                    goto label1;                    //continue system with jump to label1
                }
            }

        }


}

//=============================================================================================↙
    =
//  scanning functions
//=============================================================================================↙
    =
void clearrow1(void)            //clear the 1st row and set the others
{
    RE1=0;                      //RE1,RE0, RA5 and RA4 are the output pins from PIC which      ↙
    connect to 4 pins of keypad
    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 clearrow3(void)            //clear the 3rd row and set the others
{
    RE1=1;
    RE0=1;
    RA5=0;
    RA4=1;
}

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

void scancolumn1(void)
{
    if(RA0==0)                          //if key '1' is being pressed
    {
        while(RA0==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='1'; //Stall the '1' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA1==0)                     //if key '2' is being pressed
    {
        while(RA1==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='2'; //Stall the '2' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA2==0)                     //if key '3' is being pressed
    {
        while(RA2==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='3'; //Stall the '3' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA3==0)                     //if key 'A' is being pressed
    {
        while(RA3==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='A'; //Stall the 'A' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
}

void scancolumn2(void)
{
    if(RA0==0)                          //if key '4' is being pressed
    {
        while(RA0==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='4'; //Stall the '4' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA1==0)                     //if key '5' is being pressed
    {
        while(RA1==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='5'; //Stall the '5' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA2==0)                     //if key '6' is being pressed
    {
        while(RA2==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='6'; //Stall the '6' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA3==0)                     //if key 'B' is being pressed
    {
        while(RA3==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='B'; //Stall the 'B' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
}

void scancolumn3(void)
{
    if(RA0==0)                          //if key '7' is being pressed
    {
        while(RA0==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='7'; //Stall the '7' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA1==0)                     //if key '8' is being pressed
    {
        while(RA1==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='8'; //Stall the '8' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA2==0)                     //if key '9' is being pressed
    {
        while(RA2==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='9'; //Stall the '9' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA3==0)                     //if key 'C' is being pressed
    {
        while(RA3==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='C'; //Stall the 'C' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
}

void scancolumn4(void)
{
    if(RA0==0)                          //if key '*' is being pressed
    {
        while(RA0==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='*'; //Stall the '*' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA1==0)                     //if key '0' is being pressed
    {
        while(RA1==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='0'; //Stall the '0' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA2==0)                     //if key '#' is being pressed
    {
        while(RA2==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='#'; //Stall the '#' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
    else if(RA3==0)                     //if key 'D' is being pressed
    {
        while(RA3==0)continue;          //waiting the key to be released
        if(password_count==0)lcd_clr(); //Clear the LCD if the key is the 1st password
        lcd_goto(password_count);       //The cursor of LCD points to the column equivalent to ↙
    the value of password_count variable
        send_char('*');                 //Display the symbol '*' at LCD
        keyin_char[password_count]='D'; //Stall the 'D' value at the keyin_char array
        password_count+=1;              //increase the Password_count variable's value by 1 and↙
     the result stall back to the variable
    }
}

//=================================================================================================↙
    =
//  General Purpose functions
//=================================================================================================↙
    =
void delay(unsigned long data)
{
    for( ;data>0;data-=1);
}

void beep_once(void)
{
    buzzer=1;                       //buzzer on
    delay(50000);
    buzzer=0;                       //buzzer off
}

void beep_twice(void)
{
    buzzer=1;                       //buzzer on
    delay(8000);
    buzzer=0;                       //buzzer off
    delay(13000);
    buzzer=1;                       //buzzer on
    delay(8000);
    buzzer=0;                       //buzzer off
}

void beep_sms(void)
{
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(30000);
    buzzer=0;
    delay(13000);
    buzzer=1;
    delay(30000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(30000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(30000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(30000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
    delay(20000);
    buzzer=1;
    delay(8000);
    buzzer=0;
}
//==============================================================================================↙
    =
//  LCD functions
//=============================================================================================↙
    =
void send_config(unsigned char data)
{
    rs=0;                               //clear rs into config mode
    lcd_data=data;
    delay(50);
    e_pulse();
}

void send_char(unsigned char data)
{
    rs=1;                               //set rs into write mode
    lcd_data=data;
    delay(50);
    e_pulse();
}

void e_pulse(void)
{
    e=1;
    delay(50);
    e=0;
    delay(50);
}

void lcd_goto(unsigned char data)
{
    if(data<16)
    {
        send_config(0x80+data);
    }
    else
    {
        data=data-20;
        send_config(0xc0+data);
    }
}

void lcd_clr(void)
{
    send_config(0x01);
    delay(50);
}

void send_string(const char *s)
{
    unsigned char i=0;
    while (s && *s)send_char (*s++);

}

//=============================================================================================↙
    =
// write and read EEPROM functions
//=============================================================================================↙
    =
unsigned char read_eeprom(unsigned char address)
{
unsigned char byte;                     // Variable hold the data that is read
EECON1=0;
EEADR = address;                        // Read from this address
RD = 1;                                 // Initiate a read cycle

byte = EEDATA;                          // Fetch byte from dataregister
return byte;                            // Return the read byte
}


void write_eeprom(unsigned char data, unsigned char address)
{
EECON1=0;
EEADR = address;                        // Address to write to 0x00 to 0xFF
EEDATA = data;                          // Data to write

WREN = 1;                               // Enable writes to the EEProm
while(GIE)
GIE = 0;                                // Disable interrupts during write

EECON2 = 0x55;                          // Register not implemented on 16F684
EECON2 = 0xaa;                          // strange that this is required
WR = 1;                                 // Initiate a write cycle


while(!EEIF);                           // Wait for write to complete

WREN = 0;                               // Disable writes to EEProm
EEIF = 0;                               // Clear "write complete" flag
GIE = 1;                                // Reenable interrupts

}
```