

**SEMESTER 2 2018/2019**

**EEE 226 MICROPROCESSOR 1**

**MINI PROJECT** **REPORT**

**PROJECT TITLE:**

**PASSWORD BASED DOOR LOCK SECURITY SYSTEM**

**(HOME SECURITY SYSTEM)**

**GROUP 14**

GROUP MEMBERS: PHIROM SASOMSAP A/L EHDI (139491)

AIMAN FARHAN BIN MOHD FOAZI (139461)

LECTURER: DR. SYED SAHAL NAZLI ALHADY BIN SYED HASSAN

Date of Submission: 28 May 2019

**ACKNOWLEDGEMENT**

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

We would like to thank to our lecturers for the course EEE 226 Microprocessor . We would like to express our appreciation for their help and guidance. Without them, we would not be able to complete our project successfully. Throughout the lecturers’ teaching and guidance, we were able to understand the course better and apply the knowledge during mini project process. The lecturers would assist us on the spot when we encounter any problems.

Besides that, we would also like to thank our fellow friends for their help and cooperation in solving problems. When we faced any problems in our project, we discussed with them and with their suggestions, the problems were solved. We also like to express our gratitude for their opinions and criticisms. We were able to exchange our ideas and views about microprocessor and it had improved our knowledge in this course.

Other than that, we would also like to thank our lab partner for his or her commitment in finishing the project. We will have difficulties in completing this project if we are not fully committed. We were able to divide our work equally and both of us are satisfied with the work.

Lastly, we will also like to thank all the laboratory assistants and stockist who were willing to help us and provide us with the components. They had helped us a lot in making this project a success.

**ABSTRACT**

Nowadays, we know that even in our house is no longer a really safe place from crime. We often read about the incidents of house breaking and robbery on the newspaper. There is no surprise when we heard about this news because even our housing estate had not been spared.

Home is a place that can been inhabited by a family, workers or others. It need to be safe because there are many valuable items including a life in that home. As our initiative to help them in protecting their things and loved ones, we create and design “Home Security System”. We create home locking system is used password access to lock or open the door of a house automatically.

In order to understand this system’s operations, we divided the working principle of the password lock into some parts. The first part is the security code, where this code needed to been keyed in first to activate the system. After three times wrong password is been keyed in into the system repeatedly, the system will automatically self-locked. In this system, the security code is \*4567C#.

The next part of the system is the ability to enable the house owner to set or reset the password. House owner needed to set their password to able them open or lock the door. The password is consisted of five characters long, where numbers are only allowed. The password that has been set will being saved into the system forever as long the house owner not resetting it. We also built a resetting password system where house owner needed to key in a code in order to activate it. After the code is been keyed, the house owner then can set a new password they want

# Table of Content

|  |  |
| --- | --- |
| Content | Page number |
| Acknowledgement | 2 |
| Abstract | 3 |
| Content | 4 |
| 1.0 Introduction | 5-12 |
| 2.0 Methodology | 13-16 |
| 3.0 Result | 17 |
| 4.0 Discussions | 18 |
| 5.0 Conclusion | 20 |
| 6.0 References | 20 |
| Appendices | 21-54 |

1. **Introduction**
   1. **Brief Intro**

The theme of our mini project base on security system. Our project title is Home Security System. We are required to develop a system that is related to problem solving on daily life experience. As there are increasing burglary cases happening in daily life, we decided to create a **Home Security System** to prevent these from happening.

**1.1.1 Problem Statement**

Have you always heard about the burglary that occurs in the home stay or other places? It’s can be avoid by creating the *Home Security System.*

**1.1.2 Objectives**

Before we decided to choose security system as our mini project, we noted down several objectives for our references. The objective of this project is

1. To enhance skills in designing and analysing embedded system using 8051 microcontroller that we have learned during this semester.
2. To be able to interface 8051 microcontroller with components such as Keypad, 2x16 LCD screen and servo motor.
3. To build a friendly-user of home security system.
   1. **Proposed Solution**

The prototype of the home security system is built by interfacing the 8051 microcontroller development board to some external hardware such as LCD, servo motor and etc.

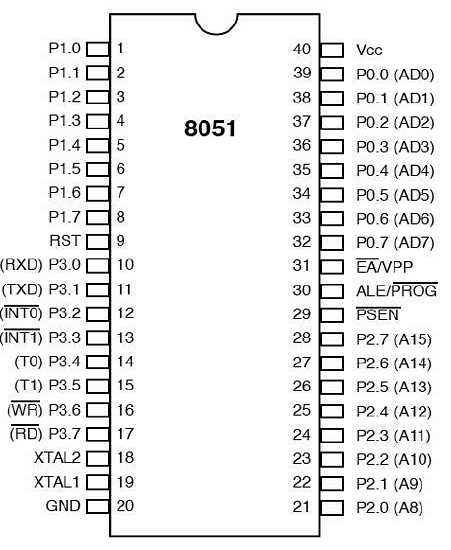
* 1. **Hardware Description**

**Microcontroller 8051**



Microcontroller 8051

The Microcontroller 8051 was designed in 1980’s by Intel. It is an 8 bit microcontroller. It has 128 bytes of RAM, 4K bytes ROM, two timers, one serial port and four general purpose input/output ports. Each port has 8 bit register. For internal functioning and processing, Microcontroller 8051 comes with an integrated built-in RAM. This is prime memory and is employed for storing temporary data. It is unpredictable memory ie. Its data will be lost when the power supply is switched OFF.

   
Pin Diagram of Microcontroller 8051

Pins 1 – 8: Known as Port 1. Unlike other ports, this port does not serve any other functions. Port 1 is an internally pulled up, quasi bi directional I/O port.

Pin 9: RESET pin, to set the 8051 microcontroller to its initial values, while the microcontroller is working or at the initial start of application. A positive pulse is given on this pin to RESET the microcontroller.

Pins 10 – 17:- Known as Port 3. Can be used as universal input or output. These are dual function pins.

Pin 10: Serial asynchronous communication input or serial asynchronous communication output.

Pin 11: Serial asynchronous communication input or serial synchronous communication output.

Pin 12: Interrupt 0 input.

Pin 13: Interrupt 1 input.

Pin 14: Counter 0 clock input.

Pin 15: Counter 1 clock input.

Pin 16: Writing signal for writing content on external RAM.

Pin 17: Reading signal to read contents of external RAM.

Pins 18 and 19: Used for interfacing an external crystal to provide system clock.

Pin 20: Vss – ground (0 V) connection.

Pins 21-28:- Known as Port 2. In addition to serving as I/O port, higher order address bus signals are multiplexed with this quasi bi directional port.

Pin 29: PSEN or Program Store Enable. Used to read signal from external program memory.

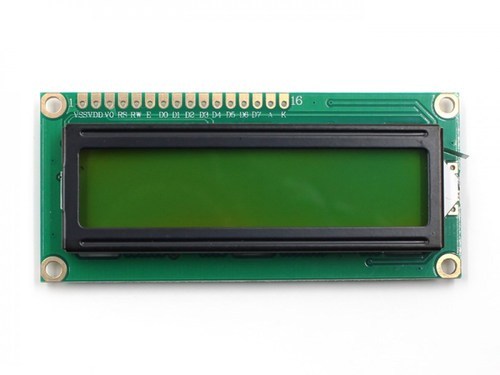
Pin 30: EA/External Access input is used to enable or disallow external memory interfacing. If there is no external memory requirement, this pin is pulled high by connecting it to Vcc.

Pin 31: Address Latch Enable (ALE). Used to demultiplex the address-data signal of port 0 (for external memory interfacing). 2 ALE pulses are available for each machine cycle.

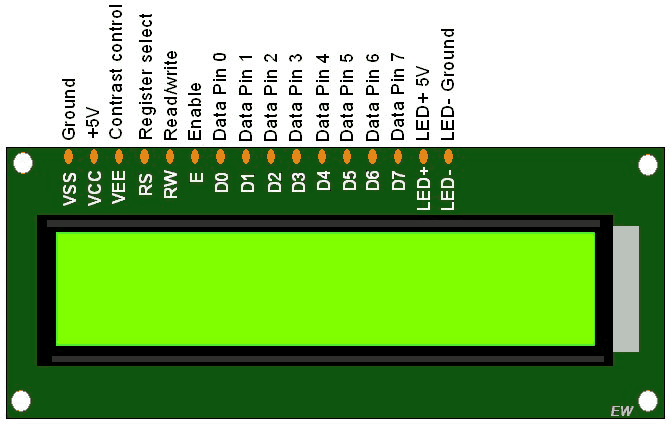
Pins 32-39: Known as Port 0 (P0.0 to P0.7) – In addition to serving as I/O port, lower order address and data bus signals are multiplexed with this port (to serve the purpose of external memory interfacing). This is a bi directional I/O port (the only one in 8051) and external pull up resistors are required to function this port as I/O.

Pin 40: Vcc. Main power source. Usually, +5V DC.

**Liquid Crystal Display (LCD) 2x16**

   
LCD 2x16

The serial LCD is functional and is easily interfaced by a microcontroller using an input or output pin. The 2x16 LCD display is preferred over seven segments and other multi segment LEDs. This is because the LCD is economical, easily programmable, have no limitation of displaying special and even custom characters, animations and so on. The LCD has two registers, namely, command and data. The command register stores the command instructions given to the LCD. The data is the ASCII value of the character to be displayed on the LCD.

**

Pin diagram of LCD 2x16

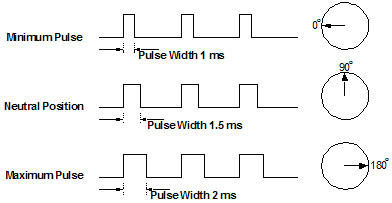
The table below shows the pin description of the LCD and its command code.

|  |  |  |
| --- | --- | --- |
| PIN No. | Name | Features |
| 1 | VSS | Ground voltage |
| 2 | VCC | Source +5V |
| 3 | VEE | Contrast voltage |
| 4 | RS | Register selection |
| 5 | R/W | Read or write modes setting |
| 6 | E | Enable pin |
| 7 | DB0 | Data bit 0 (LSB) |
| 8 | DB1 | Data bit 1 |
| 9 | DB2 | Data bit 2 |
| 10 | DB3 | Data bit 3 |
| 11 | DB4 | Data bit 4 |
| 12 | DB5 | Data bit 5 |
| 13 | DB6 | Data bit 6 |
| 14 | DB7 | Data bit 7 (MSB) |
| 15 | BPL | Back plane light |
| 16 | GND | Ground voltage |

LCD pin description.

**Servo Motor**

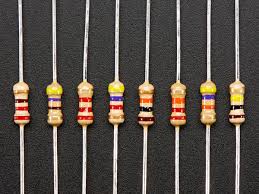
****

**

Pulse Width Modulator of Servo Motor

A servo motor is a rotary actuator that allows a precise control of angular position, velocity and acceleration.

**Resistor**

****

Resistors

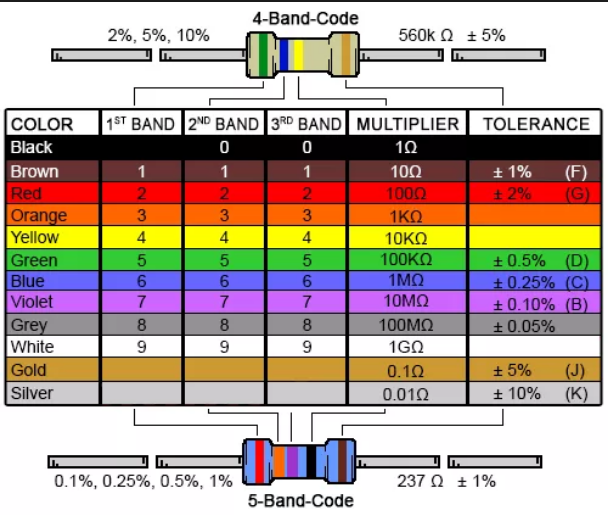


Table for resistor colour code.

A resistor is a passive two-terminal electrical component that is used to reduce current flow. The colours on the resistors represents that amount of resistance in it. Each colour has its own value.

In electronic circuits, resistors are used to limit current flow, to adjust signal levels, bias active elements, and terminate transmission line among other use.

**Keypad 4x4**

****

Keypad is used as the input data which the house owner needed to type the serial code and password for the security system. At the lowest level, keypad is organized in a matrix of rows and columns. The microprocessor access both rows and columns through ports.

**Component list**

|  |  |
| --- | --- |
| Component/Equipment | Quantity |
| 8051 Development System | 1 |
| Servo Motor | 1 |
| 2x16 LCD Screen | 1 |
| Resistor 4.7k Ohm | 4 |
| Resistor 1k Ohm | 4 |
| Resistor 150 Ohm | 3 |
| 4x4 matrix keypad | 1 |

**2.0 Methodology**

**Block Diagram**

Microcontroller

8051

LCD

Servo motor

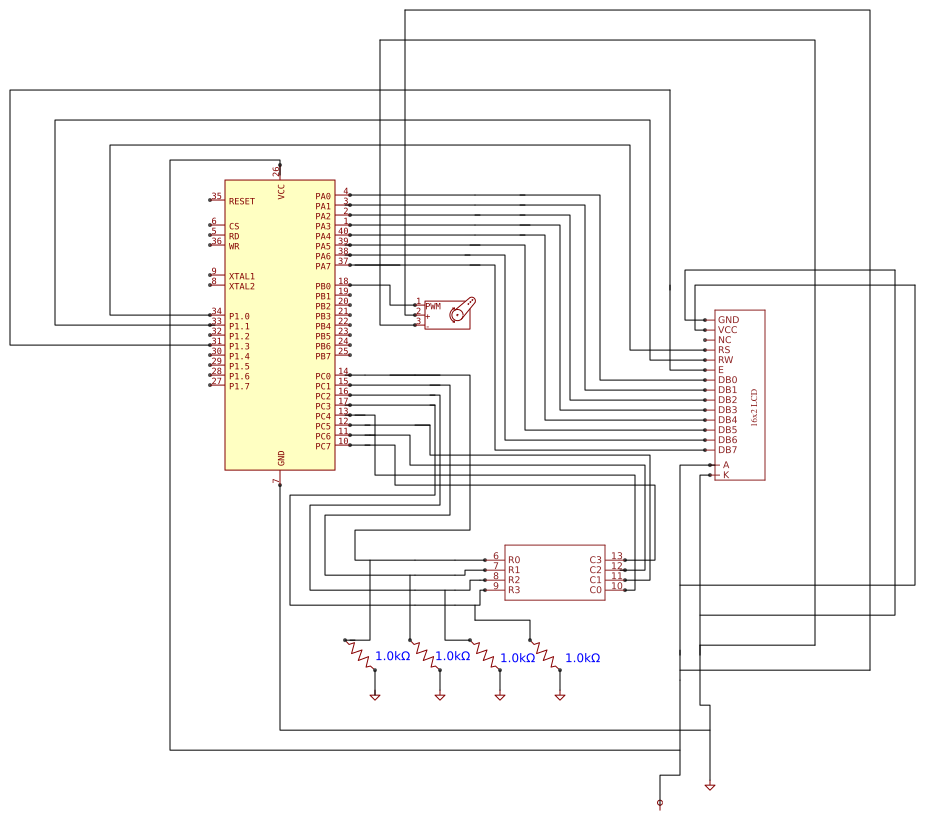
PPI 8255

Switch

Keypad

Block Diagram of the Mini Project

The block diagram above shows the connection between the microcontroller 8051 and the components. Arrows pointing into the Microcontroller 8051 or PPI 8255 indicates the input while arrows pointing away from the PPI 8255 indicates the output. The PPI 8255 is interfaces with the Microcontroller 8051.

**2.2 Schematic Diagram** 

Schematic Diagram of the Mini Project.

**2.3 FLOW CHART**

START

\Get Data from P1.0

Open the door   
(Servo Motor)

YES

P1.0=1?

NO

The door not open

Delay 5 seconds

Close the door   
(Servo Motor)

END

The flow chart of interrupt program to control the door

**2.4 Project Descriptions**

Once the system startup, the LCD screen will always display “GROUP14 PROJECT, SECURITY CODE:” which the user need to insert the password to open the door. If the user insert the wrong password, the LCD will show “TRY AGAIN!”.When the user enter the correct password, the LCD will show “CODE ACCEPTED” and PRESS # TO OPEN, PRESS \* TO RESET. When the user press# to open, by using the interrupt, the motor will rotate to open the door so that user can enter home. If the user forget to close the door, after five seconds the door will close automatically.

This project consists of 3 main part as listed below:

* LCD display
* Door Servo Motor Mechanism
* Keypad

**2.4.1 LCD Display**

LCD will display or give user the information. The user need to insert the key words to get the information from LCD.

**2.4.2 Keypad**

The user will give the instructions to the LCD by pressing the keypad.

**2.4.2 Door Servo Motor Mechanism**

The Servo Motor is used as a mechanism to open and close the door. The door will close automatically after five seconds.

1. **RESULTS**
2. First, the 2x16 LCD screen will display the introduction message “GROUP14 PROJECT”.
3. Second, the LCD screen will display “SECURITY CODE”. Then user need to key in the 5 digits security code which is “4567C”.
4. The LCD will display “TRY AGAIN ” when the user enter the incorrect password ( the chance have only three time to enter correct code).
5. When the user insert the correct password , the LCD will display “ACCEPTED!” and “PRESS# TO OPEN , PRESS\* TO RESET”.
6. The user can set the new password when press \* and the LCD will display “NEW PSSWORD”.
7. When the user press #, the LCD will display “OPENED !, WELCOME HOME ”.
8. After the word of “opened” is displayed on LCD , the motor will rotate to open the door.
9. The door will close automatically after five seconds and “AUTOMATIC LOCK” is display on LCD.

1. **DISCUSSIONS**
   1. **Servo Motor Operation for Locker Door**

The servo motor is used to open the locker door. Basically, electrical pulse signal or normally know as pulse width modulation (PWM) control the servo motor. The required pulse width is corresponding to the desired position. The timer from 8051 Microcontroller is used to generate the desire pulse width to control the servo motor movement and angle. 8051 Microcontroller has 2 timers, Timer0 and Timer1 that use the same register which is TMOD. TMOD is an 8 bit register which the lower 4 bits are set aside for timer 0 while the upper 4 bits for Timer1. The crystal frequency of the 8051 Development Board is 11.5092 MHz. To calculate the period, T, time need to complete one cycle is

Furthermore, the formula as shown below is used to calculate the required pulse width

[(FFFF-XXYY+1)] x 1.0851µs = the required pulse width

Where,

XX= data for register TH

YY= data for register TL

For example,

MOV TL0, #056H

MOV TH0, #FBH

This code will make the servo motor operate when interrupt pin INT0 has been triggered.

**Problem statement**

1. Limited I/O ports in 8051 development board

There are limited I/O ports in 8051 development board. In our mini project, we use all the IO ports. Pins of Port A are connected to LCD display, pins of Port C are connected to the keypad and pins of Port B are connected to servo and RS, RW and E . But we still found that have some error in Port B. Solution:

We connect the P1.0, P1.2 and P1.2 to RS, RW and E. This helps to solve the problem of limited I/O ports in our mini project.

1. Servo Motor

The servo motor that we used rotation was off even though we used the values from the datasheet. The variation in the internet resistance of the servo motor was found. Solution:

Adjust the timer values by either adding or subtracting the calculated value.

**5.0 CONCLUSION**

At the end of this mini project, our Home Security System prototype has been developed successfully. In this mini project, we assume that our objectives are been achieved because a home security system is been designed using microprocessor 8051 based on our basic knowledge on microprocessor lab. The knowledge and skills learned during lecture classes and lab sessions have being apply practically in this mini project. Furthermore, we create a friendly user system. Therefore, house owner can easily applied it in their house without needed to study details. We presented to our examiners at the day that we have set and showed all of our programming complete with prototype. Later, the examiners will evaluated it.

**6.0 REFERENCES**

Dr. Syed SahalNazli Alhady Syed Hassan, Encik Ahmad Nazri Ali, Dr. Zaini Bt. Abdul Halim, and Encik Suardi Kaharudin: *Laboratory Manual EEE226: Microprocessor I Semester II:* School of Electrical and Electronic Engineering, University of Science, Malaysia, 2015

Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay: *The 8051 Microcontroller and Embedded Systems*, 2nd edi., Pearson Education Limited, USA, 2006.

<https://electrosome.com/interfacing-keypad-8051-microcontroller-keil-c/>

<https://circuitdigest.com/microcontroller-projects/servo-motor-interfacing-with-8051>

**APPENDICES**

**Appendix A: Assembly Codes of Mini Project**

CPU "8051.TBL"

INCL"8051.INC"

PA: EQU 4000H ; Setup 8255 as input and output port, where Port A address 4000H

PB: EQU 4001H ; Port B address = 4001H

PC: EQU 4002H ; Port C address = 4002H

PCTR: EQU 4003H ; Control address = 4003H

ORG 2000H

MOV SP, #040H ; Load stack pointer at 040H

MOV A, #10000001B ; Port A=Output, Port B=Output, Port C upper=output, PC lower= input

MOV DPTR, #PCTR

MOVX @DPTR, A

; Pre-set 4 digit password was stored at 60H to 64H, which is "4567C"

MOV 60H, #12H ; First digit ‘4’ was stored at 60H

MOV 61H, #22H ; Second digit ‘5’ was stored at 61H

MOV 62H, #42H ; Third digit ‘6’ was stored at 62H

MOV 63H, #14H ; Fourth digit ‘7’ was stored at 63H

MOV 64H, #83H ; Fifth digit ‘C’ was stored at 63H

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Button Display

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BUTTON1: CJNE A, #11H, BUTTON2 ; Check if button "1" on keypad is pressed

MOV A, #"1" ; Display "1" on LCD if button "1" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #11H ; For purpose of data storage in next step

RET

BUTTON2: CJNE A, #21H, BUTTON3 ; Check if keypad button "2" is pressed

MOV A, #"2" ; Display "2" on LCD if button "2" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #21H ; For purpose of data storage in next step

RET

BUTTON3: CJNE A, #41H, BUTTON4 ; Check if keypad button "3" is pressed

MOV A, #"3" ; Display "3" on LCD if button "3" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #41H ; For purpose of data storage in next step

RET

BUTTON4: CJNE A, #12H, BUTTON5 ; Check if keypad button "4" is pressed

MOV A, #"4" ; Display "4" on LCD if button "4" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #12H ; For purpose of data storage in next step

RET

BUTTON5: CJNE A, #22H, BUTTON6 ; Check if keypad button "5" is pressed

MOV A, #"5" ; Display "5" on LCD if button "5" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #22H ; For purpose of data storage in next step

RET

BUTTON6: CJNE A, #42H, BUTTON7 ; Check if keypad button "6" is pressed

MOV A, #"6" ; Display "6" on LCD if button "6" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #42H ; For purpose of data storage in next step

RET

BUTTON7: CJNE A, #14H, BUTTON8 ; Check if keypad button "7" is pressed

MOV A, #"7" ; Display "7" on LCD if button "7" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #14H ; For purpose of data storage in next step

RET

BUTTON8: CJNE A, #24H, BUTTON9 ; Check if keypad button "8" is pressed

MOV A, #"8" ; Display "8" on LCD if button "8" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #24H ; For purpose of data storage in next step

RET

BUTTON9: CJNE A, #44H, BUTTON0 ; Check if keypad button "9" is pressed

MOV A, #"9" ; Display "9" on LCD if button "9" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #44H ; For purpose of data storage in next step

RET

BUTTON0: CJNE A, #28H, BUTTONA ; Check if keypad button "0" is pressed

MOV A, #"0" ; Display "0" on LCD if button "0" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #28H ; For purpose of data storage in next step

RET

BUTTONA: CJNE A, #81H, BUTTONB ; Check if keypad button "A" is pressed

MOV A, #"A" ; Display "A" on LCD if button "A" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #81H ; For purpose of data storage in next step

RET

BUTTONB: CJNE A, #82H, BUTTONC ; Check if keypad button "B" is pressed

MOV A, #"B" ; Display "B" on LCD if button "B" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #82H ; For purpose of data storage in next step

RET

BUTTONC: CJNE A, #84H, BUTTOND ; Check if keypad button "C" is pressed

MOV A, #"C" ; Display "C" on LCD if button "C" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #84H ; For purpose of data storage in next step

RET

BUTTOND: CJNE A, #88H, BT2 ; Check if keypad button "D" is pressed

MOV A, #"D" ; Display "D" on LCD if button "D" is pressed

ACALL DATAWRT

ACALL DELAY

MOV A, #88H ; For purpose of data storage in next step

RET

BT2: CJNE A, #18H, BT3 ; Check if keypad button "\*" is pressed

ACALL LCD\_ON

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #INVALID ; Display "INVALID" on LCD if button "\*" is pressed

ACALL WORD\_STRING

ACALL DATAWRT

ACALL DELAY

LJMP MAIN

BT3: CJNE A, #48H, BT1 ; Check if keypad button "#" is pressed

ACALL LCD\_ON

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #INVALID ; Display "INVALID" on LCD if button "#" is pressed

ACALL WORD\_STRING

ACALL DATAWRT

ACALL DELAY

LJMP MAIN

BT1: LJMP BUTTON1

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; LCD Interface

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LCD\_ON: MOV A, #38H ; 2 lines and 5x7 matrix

ACALL COMNWRT

ACALL DELAY

MOV A, #0FH ; Display on, cursor blinking

ACALL COMNWRT

ACALL DELAY

MOV A, #01H ; Clear display screen

ACALL COMNWRT

ACALL DELAY

MOV A, #06H ; Shift cursor to right

ACALL COMNWRT

ACALL DELAY

RET

LINE\_ONE: MOV A, #80H ; Force cursor to first line

ACALL COMNWRT

ACALL DELAY

RET

SND\_LINE: MOV A, #0C3H ; Force cursor to second line

ACALL COMNWRT

ACALL DELAY

RET

SND\_LINE1: MOV A, #0C0H

ACALL COMNWRT

ACALL DELAY

RET

LINE\_PW: MOV A, #0CBH ; Force cursor to 12th place of line 2

ACALL COMNWRT

ACALL DELAY

RET

COMNWRT: MOV DPTR, #PA ; Copy from Accumulator to PA

MOVX @DPTR, A

CLR P1.0 ; RS=0 for command

CLR P1.1 ; R/W=0 for write

SETB P1.2 ; E=1 for high pulse

ACALL DELAY

CLR P1.2 ; E=0 for H-to-L pulse

RET

DATAWRT: PUSH DPH ; To push DPTR (16 bits)

PUSH DPL

MOV DPTR, #PA ; Copy from Accumulator to PA

MOVX @DPTR, A

SETB P1.0 ; RS=1 for data

CLR P1.1 ; R/W=0 for write

SETB P1.2 ; E=1 for high pulse

ACALL DELAY

CLR P1.2 ; E=0 for H-to-L pulse

POP DPL ; To pop DPTR (16 bits)

POP DPH

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Enter Password

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

START: MOV 55H, #3

MAIN: MOV IE, #10000000B ; Disable Interrupt 0 (INT0)

ACALL LCD\_ON ; Start LCD operation

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #WEL\_MSG ; Display welcome message

ACALL WORD\_STRING

ACALL DELAY

ACALL LCD\_ON ; Start LCD operation

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #ENTER\_CODE ; Display instruction to enter password

ACALL WORD\_STRING

ACALL LINE\_PW ; Force cursor to 12th place on line 2

MAIN2: ACALL SCAN ; To acquire first input from keypad

ACALL BUTTON1 ; To compare input and display digit on LCD

MOV 65H, A ; Store first input digit into memory location 65H

CLR A

ACALL DELAY\_PRESS

ACALL SCAN ; To acquire second input from keypad

ACALL BUTTON1 ; To compare input and display digit on LCD

MOV 66H, A ; Store second input digit into memory location 66H

CLR A

ACALL DELAY\_PRESS

ACALL SCAN ; To acquire third input from keypad

ACALL BUTTON1 ; To compare input and display digit on LCD

MOV 67H, A ; Store third input digit into memory location 67H

CLR A

ACALL DELAY\_PRESS

ACALL SCAN ; To acquire fourth input from keypad

ACALL BUTTON1 ; To compare input and display digit on LCD

MOV 68H, A ; Store fourth input digit into memory location 68H

CLR A

ACALL DELAY\_PRESS

ACALL SCAN ; To acquire fifth input from keypad

ACALL BUTTON1 ; To compare input and display digit on LCD

MOV 69H, A ; Store fifth input digit into memory location 69H

CLR A

ACALL DELAY\_PRESS

LJMP COMPARE ; To check password program

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Delay

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DELAY: MOV R1, #200

HERE1: MOV R2, #250

HERE2: DJNZ R2, HERE2

DJNZ R1, HERE1

RET

DELAY\_PRESS: MOV R1, #200

HERE3: MOV R2, #255

HERE4: DJNZ R2, HERE4

DJNZ R1, HERE3

RET

DELAY\_BZ: MOV R1, #255

HERE5: MOV R2, #255

HERE6: DJNZ R2, HERE6

DJNZ R1, HERE5

RET

DELAY2: MOV R1, #20

HS1: MOV R2, #100

HS2: MOV R3, #200

HS3: DJNZ R3, HS3

DJNZ R2, HS2

DJNZ R1, HS1

RET

DELAY\_MOTOR: SETB TR0 ; Start Timer 0

AGAIN1: JNB TF0, AGAIN1 ; Monitor Timer 0 Flag

CLR TR0 ; Stop Timer 0

CLR TF0 ; Clear Timer 0 Flag

RET ; Return from subroutine

DELAY\_1SEC: MOV R2,#5

MOV R1,#45

DELAY\_1SEC0: DJNZ R1,DELAY\_1SEC0

DJNZ R2,DELAY\_1SEC0

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Word String on LCD

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WORD\_STRING:

REPEAT: MOV A, #00000000B

MOVC A, @A+DPTR ; Move data into Accumulator

JZ STOP ; Check whether string ends

ACALL DATAWRT

INC DPTR ; Move to the next data

SJMP REPEAT ; Repeat until the end of string

STOP: RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Keypad Interface

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SCAN: MOV A, #11110000B ; 5V to C0, C1, C2 and C3

MOV DPTR, #4002H

MOVX @DPTR, A ; Move Accumulator to Port C Upper

MOV DPTR, #4002H ; Get data input from Port C Lower to Accumulator

MOVX A, @DPTR

ANL A, #00001111B ; AND between Accumulator and input

JZ SCAN ; Scan repeatedly if no input detected

MOV R3, #4 ; Number of column = 4 for input

MOV R4, #00010000B ; To check input from column

SCAN\_NEXT: MOV A, R4 ; Move R4 to Accumulator

MOV DPTR, #4002H ; Move Accumulator to PC Upper

MOVX @DPTR, A

MOV DPTR, #4002H ; Get input from PC Lower to Accumulator

MOVX A, @DPTR

ANL A, #00001111B ; AND between Accumulator and Immediate data #00001111B

JZ NEXT\_COL ; Jump to NEXT\_COL if no input

ORL A, R4 ; OR with R4

RET

NEXT\_COL: MOV A, R4 ; Check the next column

RLC A ; Right shift immediate data in R4

MOV R4, A

DJNZ R3, SCAN\_NEXT ; Decrement R3 and go to SCAN\_NEXT

JMP SCAN ; Go back to SCAN if no input for all 4 columns

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Compare Key in Password

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

COMPARE: MOV A, 65H ; Compare first digit input with correct first digit

XRL A, 60H ; X-OR between immediate data stored in 65H and 60H

JNZ WRONG\_PWD ; Jump to WRONG Programme if incorrect

MOV A, 66H ; Check second digit input

XRL A, 61H

JNZ WRONG\_PWD

MOV A, 67H ; Check third digit input

XRL A, 62H

JNZ WRONG\_PWD

MOV A, 68H ; Check fourth digit input

XRL A, 63H

JNZ WRONG\_PWD

MOV A, 69H ; Check fifth digit input

XRL A, 64H

JNZ WRONG\_PWD

JZ CORRECT\_MSG ; Jump to OPTION Programme if correct

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Password wrong

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WRONG\_PWD: ACALL LCD\_ON ; Display failed attempt on LCD display

ACALL LINE\_ONE ; Move data from WRONG\_CD and display

MOV DPTR, #WRONG\_CD

ACALL WORD\_STRING

BUZZER0: MOV R4, #1

BZ0: MOV DPTR, #PB ; To activate buzzer connect to Port B6 to beep 3 times

MOV A, #11000000B

MOVX @DPTR, A

ACALL DELAY2

MOV A, #00000000B

MOVX @DPTR, A

ACALL DELAY2

DJNZ R4, BZ0

DJNZ 55H, LOOP ; Decrement of memory location 55H and jump to TRIAL if it is not zero

WRG\_3T: ACALL LCD\_ON ; Display "WRONG 3 TIMES" on LCD display if memory location 55H is zero

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #WRONG\_3T

ACALL WORD\_STRING

LOOP: LJMP MAIN

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Correct Option

:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CORRECT\_MSG: ACALL LCD\_ON

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #CRT\_MSG

ACALL WORD\_STRING

MENU\_OPTION: ACALL LCD\_ON ; Display first option to unlock case

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #OPEN\_OPT

ACALL WORD\_STRING

ACALL SND\_LINE ; Display second option to reset password

MOV DPTR, #RESET\_OPT

ACALL WORD\_STRING

ACALL SCAN ; Acquire input from keypad

CJNE A,#00011000B, NOT\_EQL

SJMP COMPARE\_1 ; To show invalid other than "\*" and "#" is pressed

NOT\_EQL: CJNE A,#01001000B, MENU\_OPTION ; Only "\*" and "#" will jump to COMPARE\_1

COMPARE\_1: CJNE A, #18H, COMPARE\_2 ; Check if keypad button "\*" is pressed

ACALL LCD\_ON ; Display instruction of entering new password

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #RESET\_PW

ACALL WORD\_STRING

ACALL LINE\_PW ; Display new password at 12th place of line 2

ACALL SCAN ; Move first digit entered into memory location 50H

ACALL BUTTON1

MOV 50H, A

ACALL DELAY\_PRESS

ACALL SCAN ; Move second digit entered into memory location 51H

ACALL BUTTON1

MOV 51H, A

ACALL DELAY\_PRESS

ACALL SCAN ; Move third digit entered into memory location 52H

ACALL BUTTON1

MOV 52H, A

ACALL DELAY\_PRESS

ACALL SCAN ; Move fourth digit entered into memory location 53H

ACALL BUTTON1

MOV 53H, A

ACALL DELAY\_PRESS

ACALL SCAN ; Move fourth digit entered into memory location 54H

ACALL BUTTON1

MOV 54H, A

ACALL DELAY\_PRESS

ACALL LCD\_ON ; Display success in reset password

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #RESET\_OK

ACALL WORD\_STRING

MOV 60H, 50H ; Move new password to pre-stored password memory location

MOV 61H, 51H

MOV 62H, 52H

MOV 63H, 53H

MOV 64H, 54H

JMP MENU\_OPTION ; Goes back to OPTION programme

COMPARE\_2: CJNE A, #48H, COMPARE\_1 ; Check if keypad button "#" is pressed

ACALL LCD\_ON ; Display execution of unlocking door procedure

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #CASE\_OPEN

ACALL WORD\_STRING

ACALL SND\_LINE1

MOV DPTR, #WLCM\_HOME

ACALL WORD\_STRING

MOV 55H, #3 ; Reset number of trial back to 3 again

LJMP TIMER

LJMP MAIN

; \*\*\*\*\*\*\*\*\*\*\*\*\* Menu Display on LCD \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WEL\_MSG: DFB "GROUP14 PROJECT ",0

ENTER\_CODE: DFB "SECURITY CODE :",0

INVALID: DFB " --DO AGAIN-- ",0

WRONG\_CD: DFB " TRY AGAIN !",0

CRT\_MSG: DFB " ACCEPTED !",0

OPEN\_OPT: DFB "PRESS # TO OPEN",0

RESET\_OPT: DFB "OR \* TO RESET",0

RESET\_PW: DFB "INSERT NEW CODE :",0

RESET\_OK: DFB "RESET COMPLETED",0

CASE\_OPEN: DFB "OPENED!",0

WLCM\_HOME: DFB "WELCOME HOME!",0

WRONG\_3T: DFB "YOU FORGOT PSSWRT!",0

CNTDWN: DFB "COUNTING DOWN TO",0

SEC1: DFB " 01 SECOND ",0

SEC2: DFB " 02 SECONDS ",0

SEC3: DFB " 03 SECONDS ",0

SEC4: DFB " 04 SECONDS ",0

SEC5: DFB " 05 SECONDS ",0

FORGET: DFB "AUTOMATIC LOCK",0

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*Countdown Timer\*\*\*\*\*\*\*\*\*\*\*\*

TIMER: MOV TMOD, #00000001B ; Timer 0 is used in mode 1

ACALL DELAY

MOV IE, #10000001B ; Enable Interrupt 0 (INT0)

SETB TCON.0

ACALL MOTOR\_OD ; To open the door

ACALL LCD\_ON ; Clear the LCD to display countdown timer

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR, #CNTDWN ; Display countdown message

ACALL WORD\_STRING

ACALL DELAY

MOV R7,#01H

CD5: ACALL SND\_LINE

MOV DPTR,#SEC5 ; Display 5th second

ACALL WORD\_STRING

ACALL DELAY\_1SEC

CJNE R7, #0H, CD4 ; Check whether interrupt (INT0) was activated or not

ACALL MOTOR\_CD ; To close the case if it was once activated and back to MAIN programme

LJMP MAIN

CD4: ACALL SND\_LINE

MOV DPTR,#SEC4 ; Display 4th second

ACALL WORD\_STRING

ACALL DELAY\_1SEC

CJNE R7, #0H, CD3 ; Check whether interrupt (INT0) was activated or not

ACALL MOTOR\_CD ; To close the case if it was once activated and back to MAIN programme

LJMP MAIN

CD3: ACALL SND\_LINE

MOV DPTR,#SEC3 ; Display 3rd second

ACALL WORD\_STRING

ACALL DELAY\_1SEC

CJNE R7, #0H, CD2 ; Check whether interrupt (INT0) was activated or not

ACALL MOTOR\_CD ; To close the case if it was once activated and back to MAIN programme

LJMP MAIN

CD2: ACALL SND\_LINE

MOV DPTR,#SEC2 ; Display 2nd second

ACALL WORD\_STRING

ACALL DELAY\_1SEC

CJNE R7, #0H, CD1 ; Check whether interrupt (INT0) was activated or not

ACALL MOTOR\_CD ; To close the case if it was once activated and back to MAIN programme

LJMP MAIN

CD1: ACALL SND\_LINE

MOV DPTR,#SEC1 ; Display last second

ACALL WORD\_STRING

ACALL DELAY\_1SEC

WARNING: ACALL LCD\_ON ; Display warning on LCD display

ACALL LINE\_ONE ; Force cursor on LCD to first line

MOV DPTR,#FORGET

ACALL WORD\_STRING

ACALL BUZZER1

ACALL MOTOR\_CD

LJMP MAIN

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Servo Motor \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MOTOR\_OD: MOV R5, #40

MOTOR\_OD1: MOV A, #00000001B ; Setup motor at PB0

MOV DPTR, #PB

MOVX @DPTR, A

MOV TL0, #0D6H ; -90° with 0.6ms high pulse

MOV TH0, #0FDH

ACALL DELAY\_MOTOR

MOV A, #00000000B ; Clear accumulator

MOV DPTR, #PB

MOVX @DPTR, A

MOV TL0, #029H ; 19.4ms low pulse

MOV TH0, #0BAH

ACALL DELAY\_MOTOR

DJNZ R5, MOTOR\_OD1

RET

MOTOR\_CD: MOV R5, #40

MOTOR\_CD1: MOV A, #00000001B ; Setup motor at PB0

MOV DPTR, #PB

MOVX @DPTR, A

MOV TL0, #000H ; +90° with 2.5ms high pulse

MOV TH0, #0F7H

ACALL DELAY\_MOTOR

MOV A, #00000000B

MOV DPTR, #PB

MOVX @DPTR, A

MOV TL0, #0FFH ; 17.5ms low pulse

MOV TH0, #0C0H

ACALL DELAY\_MOTOR

DJNZ R5, MOTOR\_CD1

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; INT 0 and INT 1

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ORG 3FF0H ; Starting address for Interrupt 0 = 3FF0H

MOV R7,#0H ; Clear Accumulator

RETI ; Return to the place where it was interrupted

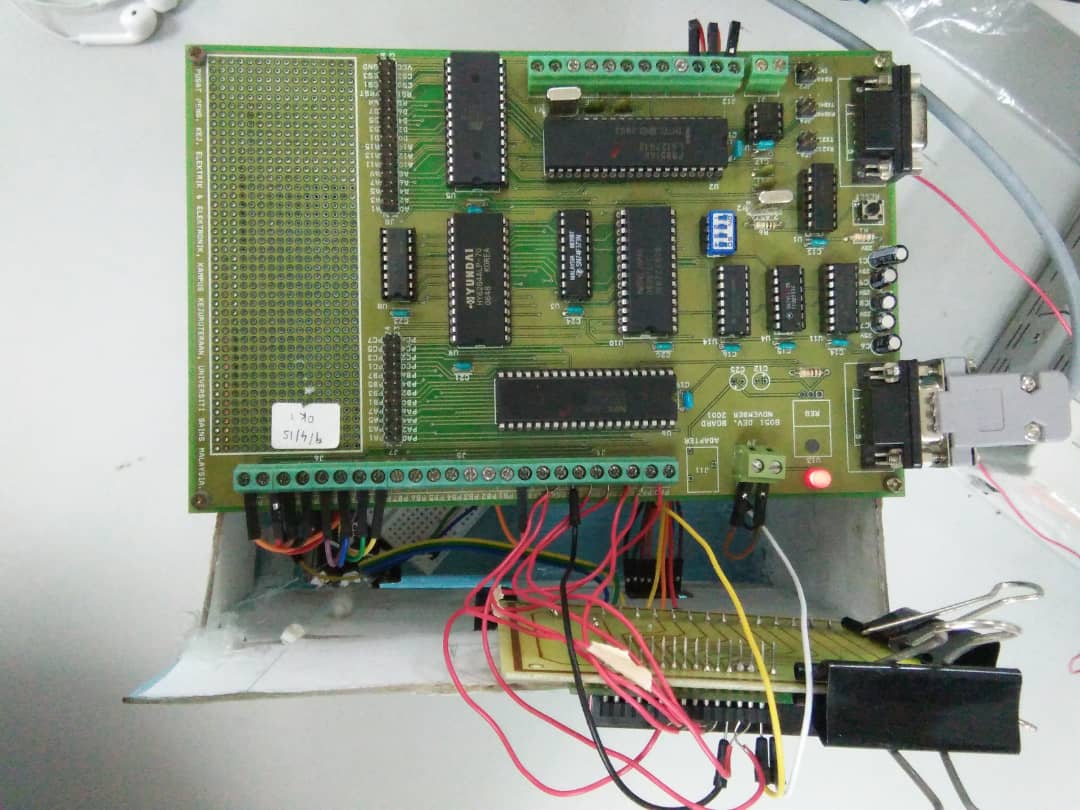
ORG 3FF6H ; Starting address for Interrupt 0 = 3FF6H

MOV R6,#0H ; Clear Accumulator

RETI

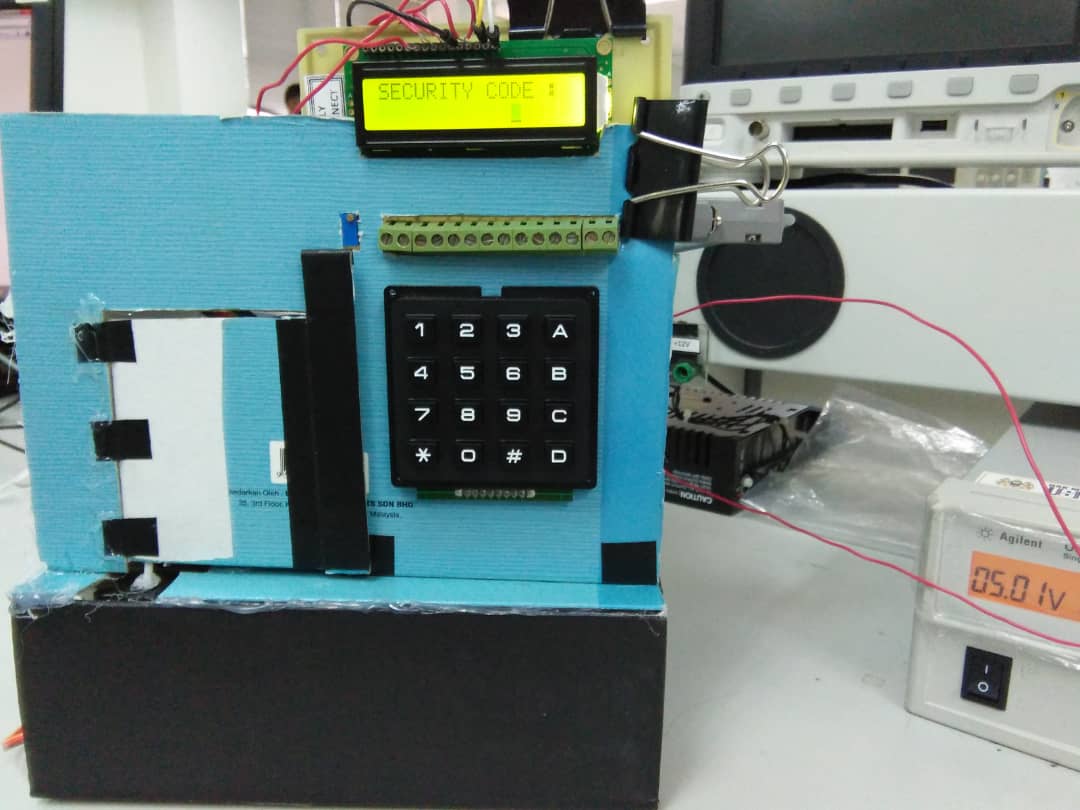
END

**Appendix B: Photo of The Mini Project.**

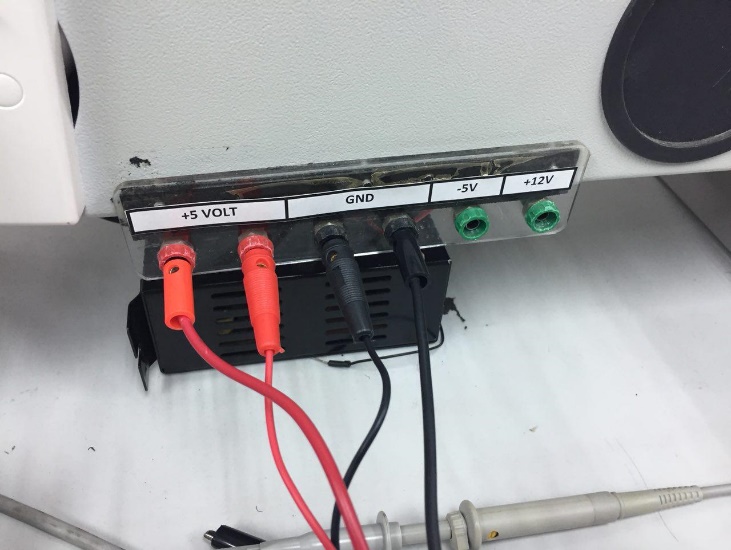
****

****

**Figure B.1:** The prototype of Home Security System.

**** 

**Figure B.2:** Working principal of Home Security System.



**FIGURE B.3:** Supply for 8051, Motor,Keyppad, LCD and breadboard.