جامعة الأزهـــر – غـــــزة

Al Azhar University - Gaza

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

FINAL PROJECT REPORT

prepared by:

Mohammed Jamal Abedellatif **20200565**

SUPERVISED BY

Eng. Siraj sersawi

JANUARY.2022

1'st SEMESTER I 2022/2023

Chapter 1
introduction

1.1 Introduction:

In this experiment , we will interface an **LCD** with a pic **microcontroller**. It is very simple and easy and is commonly used in several electronic products. LCD (Liquid Crystal Display) provides a user-friendly interface and can be very useful for debugging purposes. We will also make a program using Micro C that enable the user to enter a password and check if it is true or not and display the result on the LCD.

1.2 Objectives:

To design a complete circuit that can check if the entered password is correct or not and display the entered password on an LCD.

Chapter 2
Instruments and components

Tools, Instruments and Device:

- PIC Development Board.
- QL2006 PIC Programmer and QL-PROG software.
- Proteus simulation software.
- MikroC programming software.

Components in Proteus simulation:

- PIC16f877A.
- 2*16 LCD.
- 3*4 KEYPAD PHONE.
- OSCILATO CIRCUIT.
- BUZZER.
- GREEN LED.
- RED LED.
- NPN TRANSISTOR.
- RELAY CIRCUIT.
- LAMP (AC).

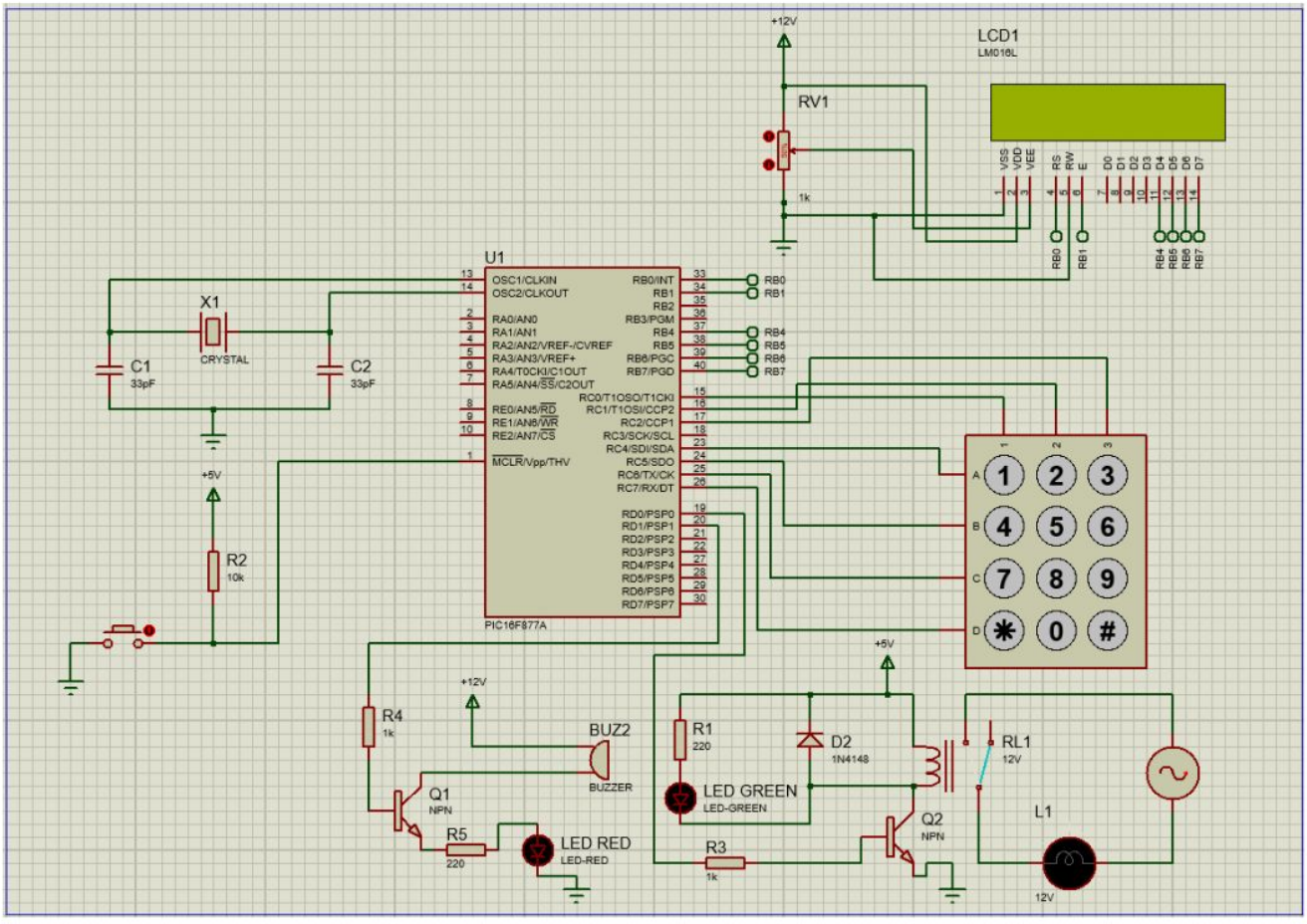
Chapter 3
Experiments

Experimental Procedures:

In this experiment, we are going to make a security system using password. The circuit uses LCD to identify a user who must enter a password to access the system. If the password is correct a Relay will be ON which can turn on any 220 VAC applications through relay contact to show that the password is correct. Also, if the entry is incorrect three times, a buzzer and red LED will be on and the system will be break.

Follow up the following steps:

1. Let's first simulate a kepad interfacing circuit using Proteus software. Open Proteus and connect the circuit as shown in Figure below.



Portc used to interface the keypad and portb used to interface LCD display screen. The relay module output connected at RD0, and the buzzer with led connected at RD1.

2. Open MikroC programming software. Choose MCU type PIC16F877a and crystal value 20MHz.

3. Write the code as shown below. Compile the code to generate hex file.

The idea of the program is to store the security password in an array, and then compare the password entered from the user with the original password (Use the password 12345, so your password is 5 digits).

• The LCD shows a message asking the user to enter the password.

• If the password is correct, then the LCD writes "Correct Password" then "Welcome^_^" and the Relay will be ON.

• If the password is wrong then the LCD writes "Wrong Password" then "Try again..", and the relay will turn Off.

• The user can clear the LCD via the (#) button but the sentence "Enter Password: "

will be written again and all output will turn off.

• If the entry is incorrect three times, a buzzer and red LED will be on and the system will be break.

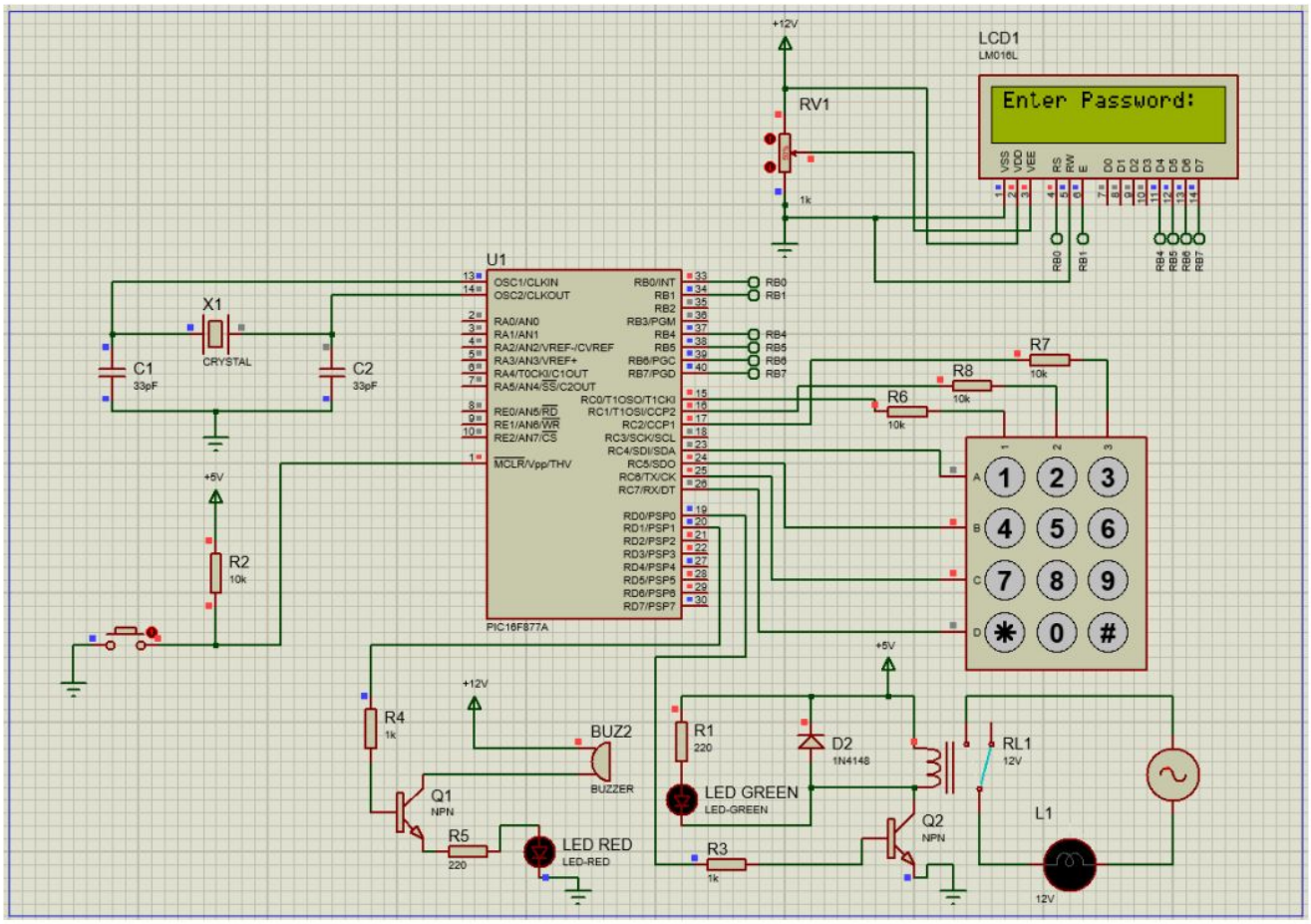
```
// Lcd pinout settings
sbit LCD_RS at RB0_bit;
sbit LCD_EN at RB1_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D4 at RB4_bit;
// Pin direction
sbit LCD_RS_Direction at TRISB0_bit;
sbit LCD_EN_Direction at TRISB1_bit;
sbit LCD_D7_Direction at TRISB7_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB4_bit;

char keypadPort at PORTC; // Configure Keypad connection at port C
char pass[6] = "12345";   // Password = 12345
char pass_input[6];       // Array to read input password
short i = 0, k = 0;       // i "count number if inputs" , k = "Read index of
pressed key"
int Q = 0;                // O "count number for wrong enter"
char key4x3(short x){     // Function to return key number
    char kp;
    switch(x){
    case 1: kp = '1'; break;
    case 2: kp = '2'; break;
    case 3: kp = '3'; break;
    case 5: kp = '4'; break;
    case 6: kp = '5'; break;
    case 7: kp = '6'; break;
    case 9: kp = '7'; break;
```

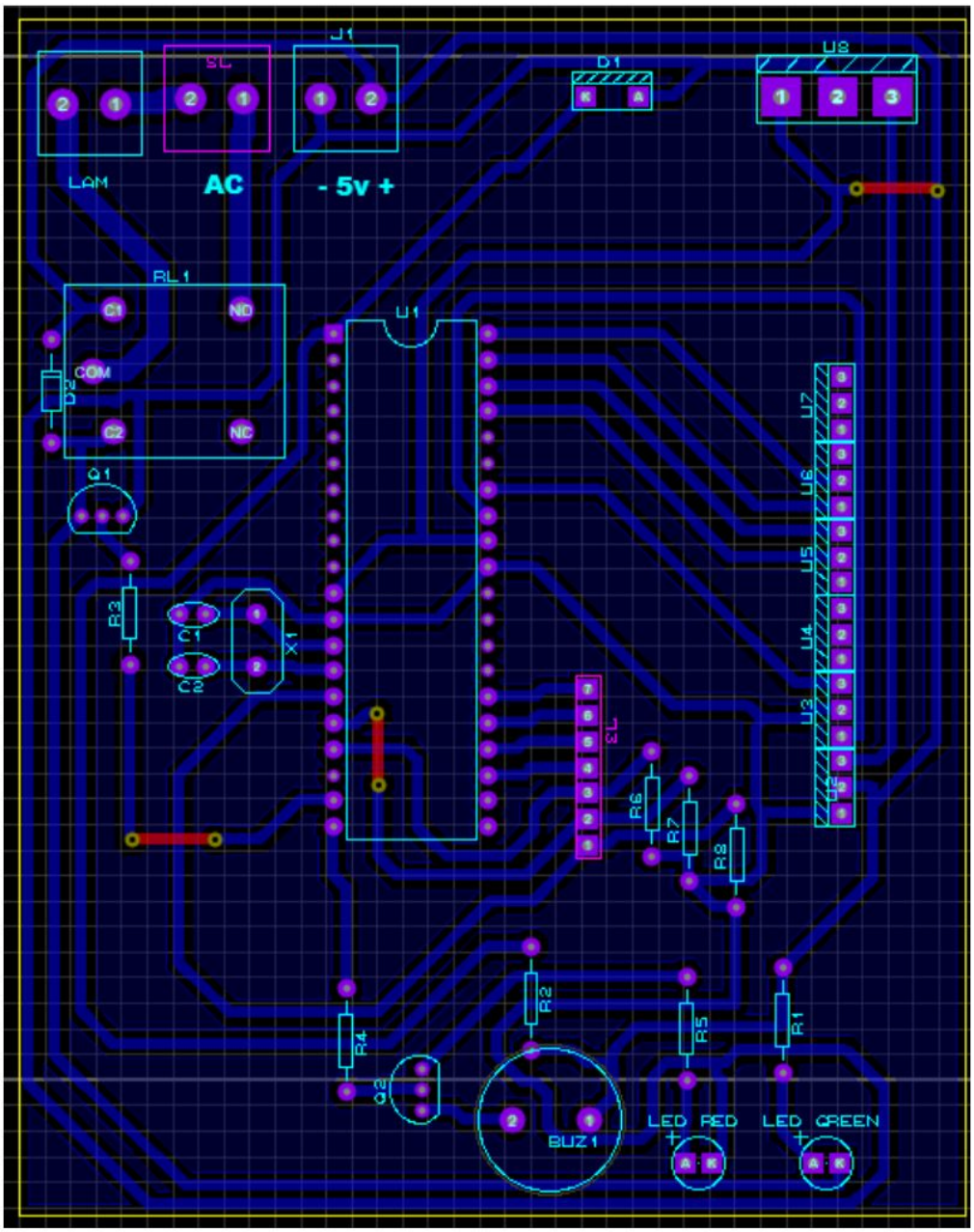
```
    case 10: kp = '8'; break;
    case 11: kp = '9'; break;
    case 13: kp = '*'; break;
    case 14: kp = '0'; break;
    case 15: kp = '#'; break;
    }
return kp;
}
void main(){
    trisd = 0;                          // Define D as output
    RD0_bit = 0;                        // Turn off RD0 portd.f0 = 0;
    RD1_bit = 0;                        // Turn off RD1 portd.f1 = 0;
    Lcd_Init();                         // Inialize LCD screen
    Keypad_Init();                      // Inialize keypad
    Lcd_Cmd(_LCD_CLEAR);                // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF);           // Cursor off
    Lcd_Out(1, 1, "Hello^_^");          // Write text in first row
    Delay_ms(1000);                     // Wait 1 seconds
    Lcd_Cmd(_LCD_CLEAR);                // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF);           // Cursor off
    Lcd_Out(1, 1, "Enter Password: "); // Write text in first row
    while (1){
        do{
            k = Keypad_Key_Click();     // read clicked keypad keys
            delay_ms(100);
            if(k != 0){ //has the user pressed on a key? // if a new key clicked
                if (key4x3(k) == '#'){   // if pressed key ='#' clear screen
                    Lcd_Cmd(_LCD_CLEAR);               // Clear display
                    Lcd_Out(1, 1, "Enter Password: "); // Write text in first row
                    RD0_bit = 0;
                    RD1_bit = 0;
                    i = 0;                             // reset counter
                    Q = 0;                             // reset counter
               }else{ // Store pressed key in input array and print it on lcd sceen
                   pass_input[i] = key4x3(k);
                   lcd_chr(2, i + 1, pass_input[i]); // Write text in second row
               i++;                                 // increment input numbers counter
                }
            }
        }
        while(i < 5); // If input keys = 5 go to next step to check password
        if(strcmp(pass, pass_input) == 0){          // if password is correct
            Lcd_Cmd(_LCD_CLEAR);                    // Clear display
            Lcd_Cmd(_LCD_CURSOR_OFF);               // Cursor off
            Lcd_Out(1, 1, "Correct Password");      // Write text in first row
            Lcd_Out(2, 1, "Welcome^_^");           // Write text in second row
            RD0_bit = 1;                            // Turn on relay
            delay_ms(2000);                         // Wait 2 seconds
            Q = 0;                                  // reset counter
        }else{                                      // If password is wronge
            Lcd_Cmd(_LCD_CLEAR);                    // Clear display
            Lcd_Cmd(_LCD_CURSOR_OFF);               // Cursor off
            Lcd_Out(1, 1, "Wrong Password!");       // Write text in first row
            Lcd_Out(2, 1, "Try again..");           // Write text in second row
            RD0_bit = 0;                            // Turn off relay
            delay_ms(2000);                         // Wait 2 seconds
            Q++;                                    // count +1
```

```
        }
        if( Q == 3 ){
           Lcd_Cmd(_LCD_CLEAR);                    // Clear display
           Lcd_Cmd(_LCD_CURSOR_OFF);               // Cursor off
           Lcd_Out(1, 1, "warning!!");             // Write text in first row
           Lcd_Out(2, 1, "system break");          // Write text in second row
           RD1_bit = 1;                            // Turn on D1
           break;
        }else{
           Lcd_Cmd(_LCD_CLEAR);                    // Clear display
           Lcd_Cmd(_LCD_CURSOR_OFF);               // Cursor off
           Lcd_Out(1, 1, "Enter Password: ");      // Write text in first row
           i = 0;
        }
    }
}
```

4. Add hex file to simulation circuit in Proteus. Run and test the code as shown below.



5. Then using proteus to design PCB layout as shown below.



6. connect portc Pins in microcontroller to keypad columns and row pins in kit. Then, connect portb pins in microcontroller to LCD screen pins. Finally, connect RD0 in microcontroller to relay control pin in kit, and RD1 to buzzer and led pin in PCB.