



CSE 331.3

Project Report

Title: Password based door lock system using 8051 microcontroller and C

Group No: 01

Group Members: Md. Tareq Mahmud (1811174042)

Shayam Imtiaz Shuvo (1811710642)

Nafis Iqbal (1811250042)

Introduction:

The purpose of our project is to build a password based door lock system using 8051 microcontroller. Password base lock systems are very popular now-a-days because of their simple yet very effective applications. Password based systems are replacing traditional key and lock based system and password based lock systems are also more secure and user friendly than the traditional Key based systems.

Tools:

- **Microcontroller 80C51**
- **2*16 LED Display**
- **Keypad**
- **Motor**
- **Proteus Simulation Tool**
- **Keil uv5**

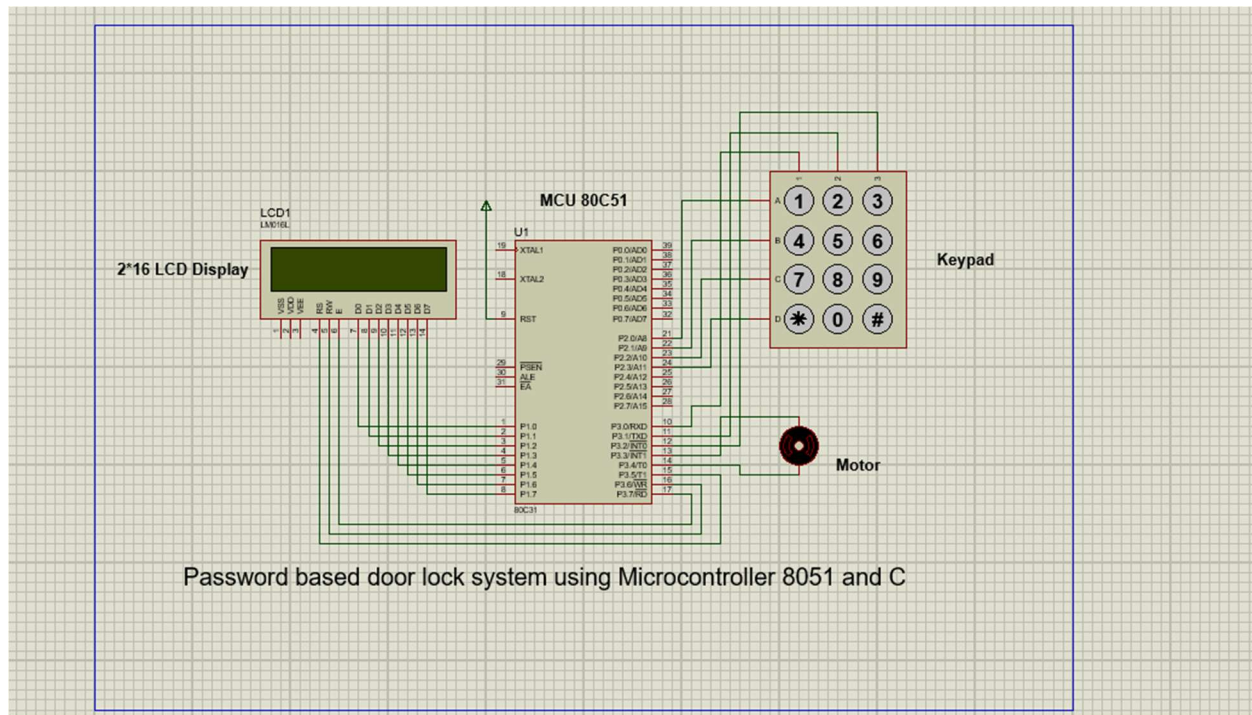


Figure: Schematic Diagram of Password based door lock system

Codes and Project Files:

Proteus simulation file and hex file of embedded c code is being uploaded in github repository, <https://github.com/Tareq69/Doorlocksystemusing8051> . We have used comments in the embedded c code that explains how the code works. Once we are done with project demonstration the repository will be set to public and should be viewed or used for educational purposes only. Embedded C code is also given below:

```
#include <reg51.h>
```

```
sbit r1 = P2 ^ 0;    // Setting Row 1 of keypad
```

```
sbit r2 = P2 ^ 1;    // Setting Row 2 of keypad
```

```
sbit r3 = P2 ^ 2;    // Setting Row 3 of keypad
```

```
sbit r4 = P2 ^ 3;    // Setting Row 4 of keypad
```

```
sbit c1= P3 ^ 0;    // Setting column 1 of keypad
```

```
sbit c2 = P3 ^ 1;           // Setting column 2 of keypad
sbit c3 = P3 ^ 2;           // Setting column 3 of keypad
```

```
sbit motorpin1 = P3 ^ 3;    // setting motorpin 1
sbit motorpin2 = P3 ^ 4;    // setting motorpin 2
```

```
sbit rs = P3 ^ 5;           // setting the registr select pin
sbit rw = P3 ^ 6;           // setting the read/write pin
sbit en = P3 ^ 7;           // setting the enable pin
```

```
unsigned char default_pin[] = {"737271"}; // Predefined Pin
unsigned char user_pin[6]; // Pin entered by the user will be stored in this array
```

```
void delay(unsigned int j)
{
    int a, b;
    for (a = 0; a < j; a++)
    {
        for (b = 0; b < 8; b++)
        {
            }
        }
    }
}
```

```
void lcdCommand(unsigned char A)
```

```
{
    P1 = A;                      // Port 1 will take A as input
    rs = 0;
    rw = 0;                      // writingdata of p1 in LCD
    en = 1;
    delay(1000);
    en = 0;
}
```

```
void lcdData(unsigned char i)
```

```
{
    P1 = i;                      // // Port 1 will take i as input
    rs = 1;
    rw = 0;                      // writingdata of port 1 in LCD
    en = 1;
    delay(1000);
    en = 0;
}
```

```
void display(unsigned char *q)
```

```
{
    int k;
    for (k = 0; q[k] != '\0'; k++) // Until q reaches end of line display the string stored in character
    pointer
    {
        lcdData(q[k]);
    }
    delay(1000);
}
```

```
}
```

```
char keypadInput()
```

```
{
```

```
    int x = 0;
```

```
    while (x == 0)
```

```
    {
```

```
        r1 = 0; // Selecting row 0
```

```
        r2 = 1;
```

```
        r3 = 1;
```

```
        r4 = 1;
```

```
        if (c1 == 0) // selecting element in row 0 and col 0 of keypad
```

```
        {
```

```
            lcdData('*');
```

```
            delay(1000);
```

```
            x = 1;
```

```
            return '1';
```

```
        }
```

```
        if (c2 == 0) // selecting element in row 0 and col 0 of keypad
```

```
        {
```

```
            lcdData('*');
```

```
            delay(1000);
```

```
            x = 1;
```

```
            return '2';
```

```
        }
```

```
        if (c3 == 0) // selecting element in row 0 and col 0 of keypad
```

```
        {
```

```
            lcdData('*');
```

```
            delay(1000);
```

```
            x = 1;
```

```

    return '3';
}
r1 = 1;
r2 = 0;           // Selecting row 1
r3 = 1;
r4 = 1;

if (c1 == 0) // selecting element in row 1 and col 0 of keypad
{
    lcdData('*');
    delay(1000);
    x = 1;
    return '4';
}
if (c2 == 0) // selecting element in row 1 and col 1 of keypad
{
    lcdData('*');
    delay(1000);
    x = 1;
    return '5';
}
if (c3 == 0) // selecting element in row 1 and col 2 of keypad
{
    lcdData('*');
    delay(1000);
    x = 1;
    return '6';
}

r1 = 1;

```

```

r2 = 1;
r3 = 0;           // Selecting row 2
r4 = 1;
if (c1 == 0) // selecting element in row 2 and col 0 of keypad
{
    lcdData('*');
    delay(1000);
    x = 1;
    return '7';
}
if (c2 == 0) // selecting element in row 2 and col 1 of keypad
{
    lcdData('*');
    delay(1000);
    x = 1;
    return '8';
}
if (c3 == 0) // selecting element in row 2 and col 2 of keypad
{
    lcdData('*');
    delay(1000);
    x = 1;
    return '9';
}

r1 = 1;
r2 = 1;
r3 = 1;
r4 = 0; // Selecting row 3

```



```

    if (c1 == 0) // selecting element in row 3 and col 0 of keypad
    {
        lcdData('*');
        delay(1000);
        x = 1;
        return '*';
    }
    if (c2 == 0) // selecting element in row 3 and col 1 of keypad
    {
        lcdData('*');
        delay(1000);
        x = 1;
        return '0';
    }
    if (c3 == 0) // selecting element in row 3 and col 2 of keypad
    {
        lcdData('*');
        delay(1000);
        x = 1;
        return '#';
    }
}

void passwordCheck()
{
    if (default_pin[0] == user_pin[0] && default_pin[1] == user_pin[1] && default_pin[2] ==
user_pin[2] && default_pin[3] == user_pin[3] && default_pin[4] == user_pin[4] &&
default_pin[5] == user_pin[5]) // checking whether the default password matches with the
password user has entered
    {

```

```

    delay(1000);
    lcdCommand(0x01); // Clearing the display
    lcdCommand(0x80); // displaying line 1 of lcd

    display("Access Granted");
    delay(1000);
    motorpin1 = 1; // Motor will be on and move in clockwise direction once
    motorpin2 = 0;
    lcdCommand(0xc0);          // displaying line 2 of lcd
    display("DOOR OPENED");
                                motorpin1 = 1; // Motor will be on and move in clockwise
direction once
    motorpin2 = 0;
    lcdCommand(0x01);          // Clearing the display
}
else
{
    lcdCommand(0x01);          // Clearing the display
    lcdCommand(0x80);          // selecting line 1 of lcd
    display("Access denied");
    delay(1000);
    lcdCommand(0x01); // Clearing the display
}
}

void main()
{
    lcdCommand(0x0F);          // Turning the LCD on

```

```
    lcdCommand(0x38); // 2 lines will be displayed in  
the lcd
```

```
    lcdCommand(0x01); // Clearing the display
```

```
while(1)
```

```
{
```

```
    unsigned int i = 0;
```

```
    lcdCommand(0x80); // displaying line 1 of lcd
```

```
    display("Enter your pin");
```

```
    delay(1000);
```

```
    lcdCommand(0xc0); // displaying line 2 of lcd
```

```
    while (default_pin[i] != '\0') // Reading default pin untill it reaches end of line
```

```
    {
```

```
        user_pin[i] = keypadInput(); // KeypadInput is stored in user_pin array
```

```
        delay(1000);
```

```
        i++;
```

```
    }
```

```
    passwordCheck(); // Checking the default pin and user input pin
```

```
}
```

```
}
```

Work Procedures and discussion:

- we have used MCU8051, A 2*16 lcd display, A 4*3 keypad and a motor to do this simulation.
- We connected the lcd display with port 1 of mcu, Rows of the 4*3 keypad with port 2.0 to port 2.3 and the columns of 4*3 keypad with port 3.0 to port 3.2. Motor pins are connected with port 3.3 and 3.4 of mcu8051. An internal oscillator that is inside the 8051 is used to give the clock cycle which will execute the instructions of the hex file. No power connection is needed as we are doing the simulation on proteus. This completes all the necessary connection between the devices to do the simulation.
- After establishing all the connection with mcu and the other devices we wrote the c code in keil uv5. This code contains all the instructions that will be given to the mcu 8051 to run the simulation.
- At the beginning of the code, we have included the reg51.h header file which includes all the library functions and necessary codes that is used to write code for Microcontroller 8051.
- Then, we set all the connections of keypad, lcd and motor and the ports of mcu 8051 using the setb command. Then we declare one initialized array of 6 characters which holds the predefined pin “737271”. Then we declare another array of length 6 but it is uninitialized as it will be used to check the input password and predefined password later on.
- Next, We define 6 functions named as Delay(), lcdCommand(), lcdData(), lcdDisplay(), keypadInput(), Passwordcheck(). lcdCommand function is used to give certain lcdCommand, lcdData function is used to pass data in the lcd, lcdDisplay is used to display strings as character input and it has maximum 16 character displaying capability per line and only for 2 lines because we are using a 2*16 lcd. Delay function is used to create delay in the code. The delay time unit is in millisecond (ms). KeypadInput() function is used to return the character inputted in the keypad and lastly Passwordcheck() function is used to match the check whether the input password and predefined password is same or not. If password is same we display “Access granted” in line 1 and in line 2 we display “Door opened”. If passwords do not match we display “Access denied”.
- Lastly, we have the main function where we use the predefined lcdCommand function to turn the lcd on. Then we run an infinite loop and take user input and pass it into the uninitialized array of length 6. Finally, we call the passwordcheck() function to check whether the input password is correct or not.

While doing this project we got familiarized with proteus simulation tool, Keil uv5 ide. We also learned how to use the lcd, keypad and motors to build digital systems. We learned how to write c code for programming mcu8051 and how to generate hex file from the c code and how to use it with the proteus simulation file to run the simulation successfully. We faced a difficulty when the motor that we used in the proteus simulation was not working. But after running proteus in admin mode the problem resolved. The full project was done with the help of resources found in online. The link of the tutorial is, <https://www.youtube.com/watch?v=oB4Qgi60oAY&t=1634s> . We had no difficulties other than the motor not working. Everything else worked fine. It would have been better if we could do this project offline, in university campus but due to covid we had no other option than doing the simulation online in proteus as the lab was not held offline, We look forward to implement this project offline if possible.

Contribution Table:

Name	
Md. Tareq Mahmud	Proteus simulation file, Embedded c code, report
Nafis Iqbal	Embedded c code, report
Shayam Imtiaz Shuvo	Embedded c code, report