



North South University

Department of Electrical & Computer Engineering

Project Report

Course Name: Microprocessor Interfacing & Embedded System

Course Code: CSE 331

Semester: Summer 2020

Section: 06

Faculty Name: Rishad Arfin (Rsf)

Project Title: Password-based door lock system using 8051/PIC microcontroller

Submitted By-

Group No: 05

Group Memebers:

1. K.M.H. Mubin - 162 1802 042
2. Miti Mostari Patwary – 1610073042

Submission Date: September 30, 2020

Objective:

Safety is the most crucial concern of human. We always try to keep our things between ourselves. For this reason, we still use various methods to lock our precious items like a locked diary. And when it comes to our daily life, we are more serious. In the modern age, there are so many ways to lock the door; one of them is password-based lock system—a system where you are the only one to know how to access it. It saves our daily life from the various malicious problem like a thief. This system will give us the security that we want. To make our life more secure, we are going to build the password-based door lock system. This system is easy to assemble and very easy to use in our daily life. Anyone can use it to secure themselves.

Applications:

The password-based lock system can install any door of any rooms. This system also can be integrated with the existing system. This electric combination lock system uses a five-digit password. The system collects five-digit user input and compares the user input with the preset password inside the program. If the password matches, access will be granted, and if not match the entry will be denied. The system can be used at residential places to ensure better safety. It can be used at organizations to ensure authorized access to highly secured places.

Block Diagram:

In this diagram every block is represent as a part or function of the program or device.

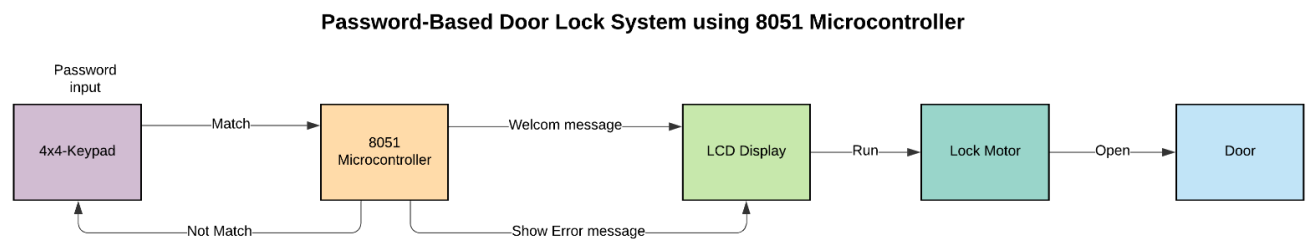


Fig: Password Based Door Lock System Block Diagram

Working Procedures:

1. The main component in the circuit is 8051 microcontrollers. This control everything in the device.
2. We connect the 4x3 keypad with the microcontroller. In the keypad there is 4 rows which is indicated the letters (A-D) and the 3 columns which is indicated the number (1-3).
3. In the 8051 controller, pin P2.0 to P2.3 are connected to the keypad rows and pin P3.0 to P3.2 are connected to the keypad columns.
4. Then we connect the door lock motor pins in the P3.3 and P3.4. This motor only work when the password is right.

5. LCD's points are connected with P3.5 to P3.7 in 8051. LCD help us to show the inputs and error messages.
6. All devices are connected with each other. In the simulation we use the hex code to run the program. The hex code is generated from the embedded C code. This code only works on any 8051-microcontroller system.
7. To run the simulation, we set the clock frequency at 11.0592 MHZ for 8051. This is allowed us to run the simulation without getting any errors.
8. We pre-define the password 12345 for the program.
9. If user input wrong password it will give another chance to input the password again with an error message. When the password is right it will show the welcome message and lock motor will move to open the door.

Schematic Circuit Diagram:

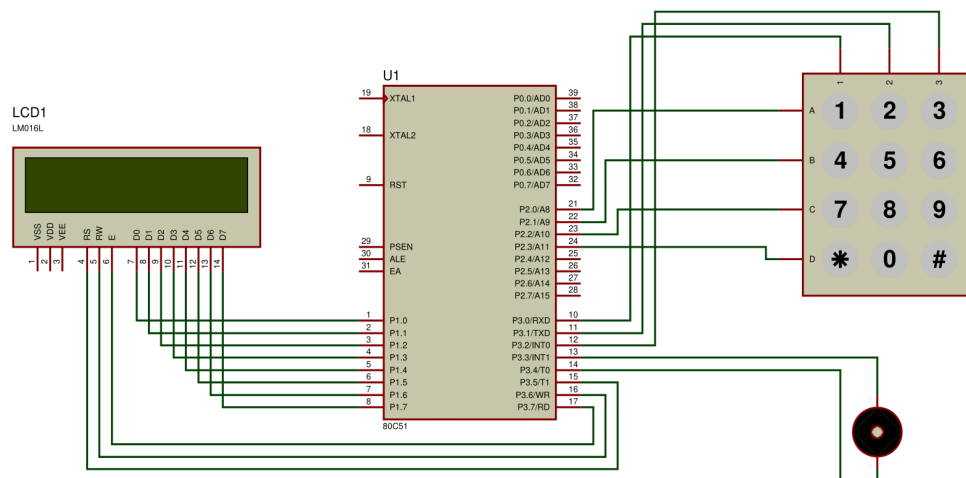


Fig: Password base Door locking system schematic diagram

Project Code:

```
1. // password based door lock system in 8051 microprocessor
2.
3. // for 8051 family
4. #include <reg51.h>
5.
6. // connected pins
7. // keypad rows
8. sbit keyrow1 = P2 ^ 0;
9. sbit keyrow2 = P2 ^ 1;
10. sbit keyrow3 = P2 ^ 2;
11. sbit keyrow4 = P2 ^ 3;
12. //keypad column
13. sbit keycolumn1 = P3 ^ 0;
14. sbit keycolumn2 = P3 ^ 1;
15. sbit keycolumn3 = P3 ^ 2;
16.
17. // motor pins
18. sbit motorpin1 = P3 ^ 3;
19. sbit motorpin2 = P3 ^ 4;
20.
21. // led pins
22. sbit rs = P3 ^ 5;
23. sbit rw = P3 ^ 6;
24. sbit en = P3 ^ 7;
25.
26. //functions
27. void lcdcmd(unsigned char);
28. void lcddat(unsigned char);
29. void lcddisplay(unsigned char *q);
30. char keypad();
31. void check();
32. void delay(unsigned int);
33. unsigned char pin[] = {"12345"};
34. unsigned char Epin[5];
35.
36. // main function
37. void main()
38. {
39.     lcdcmd(0x0F); //decimal value: 15
40.     lcdcmd(0x38); //decimal value: 56
41.     lcdcmd(0x01); //decimal value: 1
42.
43.     while (1)
44.     {
45.         unsigned int i = 0;
46.         lcdcmd(0x80); //decimal value: 128
47.         lcddisplay("ENTER PIN NUMBER");
48.         delay(1000);
49.         lcdcmd(0xc0); //decimal value: 192
50.         while (pin[i] != '\0')
51.         {
52.             Epin[i] = keypad();
53.             delay(1000);
54.             i++;
55.         }
56.         check();
57.     }
58. }
```

```

59.
60. //delay function
61. void delay(unsigned int j)
62. {
63.     int a, b;
64.     for (a = 0; a < j; a++)
65.     {
66.         for (b = 0; b < 10; b++)
67.             ;
68.     }
69. }
70.
71. // lcd commands functions
72. void lcdcmd(unsigned char A)
73. {
74.     P1 = A;
75.     rs = 0;
76.     rw = 0;
77.     en = 1;
78.     delay(1000);
79.     en = 0;
80. }
81.
82. //lcd data function
83.
84. void lcddat(unsigned char i)
85. {
86.     P1 = i;
87.     rs = 1;
88.     rw = 0;
89.     en = 1;
90.     delay(1000);
91.     en = 0;
92. }
93.
94. //lcd display charecters function
95.
96. void lcddisplay(unsigned char *q)
97. {
98.     int k;
99.     for (k = 0; q[k] != '\0'; k++)
100.    {
101.        lcddat(q[k]);
102.    }
103.    delay(10000);
104. }
105.
106. // assign keypad character value function
107.
108. char keypad()
109. {
110.     int x = 0;
111.     while (x == 0)
112.     {
113.         // assign values for first row
114.         keyrow1 = 0;
115.         keyrow2 = 1;
116.         keyrow3 = 1;
117.         keyrow4 = 1;
118.         if (keycolumn1 == 0)
119.         {

```

```

120.         lcdat('*');
121.         delay(1000);
122.         x = 1;
123.         return '1';
124.     }
125.     if (keycolumn2 == 0)
126.     {
127.         lcdat('*');
128.         delay(1000);
129.         x = 1;
130.         return '2';
131.     }
132.     if (keycolumn3 == 0)
133.     {
134.         lcdat('*');
135.         delay(1000);
136.         x = 1;
137.         return '3';
138.     }
139.     // assign values for second row
140.     keyrow1 = 1;
141.     keyrow2 = 0;
142.     keyrow3 = 1;
143.     keyrow4 = 1;
144.
145.     if (keycolumn1 == 0)
146.     {
147.         lcdat('*');
148.         delay(1000);
149.         x = 1;
150.         return '4';
151.     }
152.     if (keycolumn2 == 0)
153.     {
154.         lcdat('*');
155.         delay(1000);
156.         x = 1;
157.         return '5';
158.     }
159.     if (keycolumn3 == 0)
160.     {
161.         lcdat('*');
162.         delay(1000);
163.         x = 1;
164.         return '6';
165.     }
166.
167.     // assign values for third row
168.     keyrow1 = 1;
169.     keyrow2 = 1;
170.     keyrow3 = 0;
171.     keyrow4 = 1;
172.     if (keycolumn1 == 0)
173.     {
174.         lcdat('*');
175.         delay(1000);
176.         x = 1;
177.         return '7';
178.     }
179.     if (keycolumn2 == 0)
180.     {

```

```

181.         lcddat('*');
182.         delay(1000);
183.         x = 1;
184.         return '8';
185.     }
186.     if (keycolumn3 == 0)
187.     {
188.         lcddat('*');
189.         delay(1000);
190.         x = 1;
191.         return '9';
192.     }
193.
194.     // assign values for forth row
195.     keyrow1 = 1;
196.     keyrow2 = 1;
197.     keyrow3 = 1;
198.     keyrow4 = 0;
199.
200.     if (keycolumn1 == 0)
201.     {
202.         lcddat('*');
203.         delay(1000);
204.         x = 1;
205.         return '*';
206.     }
207.     if (keycolumn2 == 0)
208.     {
209.         lcddat('*');
210.         delay(1000);
211.         x = 1;
212.         return '0';
213.     }
214.     if (keycolumn3 == 0)
215.     {
216.         lcddat('*');
217.         delay(1000);
218.         x = 1;
219.         return '#';
220.     }
221. }
222. }
223.
224. // password check function and run the door motor
225.
226. void check()
227. {
228.     // compare the input value with the assign password value
229.     if (pin[0] == Epin[0] && pin[1] == Epin[1] && pin[2] == Epin[2] && pin[3] ==
Epin[3] && pin[4] == Epin[4])
230.     {
231.         delay(1000);
232.         lcdcmd(0x01); //decimal value: 1
233.         lcdcmd(0x81); //decimal value: 129
234.         // show pin is correct
235.         lcddisplay("PIN CORRECT");
236.         delay(1000);
237.         // door motor will run
238.         motorpin1 = 1;
239.         motorpin2 = 0;
240.         lcdcmd(0xc1); //decimal value: 193

```

```

241.          // show the door is unlocked
242.          lcddisplay("DOOR OPENED");
243.          delay(10000);
244.          motorpin1 = 1;
245.          motorpin2 = 0;
246.          lcdcmd(0x01); //decimal value: 1
247.      }
248.      else
249.      {
250.          lcdcmd(0x01); //decimal value: 1
251.          lcdcmd(0x80); //decimal value: 128
252.          lcddisplay("WRONG PIN");
253.          delay(1000);
254.          lcdcmd(0x01); //decimal value: 1
255.      }
256.  }
257.
258.  // end

```

Discussion:

In the program first we import the 8051-family header file which contain all the necessary classes. Then we assign every connected pin to the variables.

In keypad we assign variable as keyrow1 to keyrows4 for keypad rows and keycolumn1 to keycolumn3 for keypad columns. Then we connected the motor pins and lcd pins in the variable called motor pins and rs, rw and en.

After assign all the connected pins to the variables. We declare functions for every single task. The main function calls the other functions for execute the program.

Delay function is used for delay the program execution for few seconds. It helps us to execute the program smoothly. We delay the execution for only 1000ms = 1secs.

The lcdcmd function is help us to control the lcd. It will control the current flow for the lcd screen and also help to perform Read/Write Operation.

The lcdat function control the lcd data pins. In the LCD there is total 8 pins D0 to D7 Pins used to send Command or data to the LCD.

The lcdisplay function is used for display user input characters and also assign the values for every keypad character values. First, we select a single keypad row and assign the values for the key. Also encoded with the asterisk (*) character, for pin protection.

The check function is used for compare the user input with the assign password number. If the user input match the assign password, it will show the "pin correct" message and give the signal to run the lock motor. After lock motor will run, again it will show the "door open" message. On the other hand, if the user input password is wrong, it will show the "wrong pin" message. Then it will delay few second and give another chance to input the password again.

The main function is the mother function of the program. Main function will call other function in other to perform the task. At first it will call the lcdcmd function to control the lcd, then it will call the

lcdisplay function for display the messages in the lcd screen. It will delay for few seconds. After that it will run loop until the correct pin enter. To check the password is correct it will call check function, then execute the program.

To simulate the program circuit, we use the proteus simulation. All the code is written in embedded C code, which is not directly executed by the 8051 microcontrollers. To run the program first we need to build the program and convert the program in hex file. The hex file is supported by the 8051 microcontrollers. Every 8051 controller has a clock frequency for perform the task. To run the program, we use 11.0592 MHZ clock frequency. It will run the simulation smoothly without any kind of errors.

Contribution:

Name	Contribution
K.M.H.Mubin	Schematic circuit diagram, code, report
Miti Mostari Patwary	Proposal writing