Password Protected Locking System Using Arduino

Sriharsha B S\textsuperscript{1}, Zabiullah\textsuperscript{2}, Vishnu S B\textsuperscript{3} and Sanju V\textsuperscript{4}

Abstract - In this current situation, the degree of security is feeble. So there is a lot of robbery, theft going on in and around the world. So, people fear to keep any of their valuables in their homes. Henceforth, many people prefer to keep it in banks. However, in this insecure world even banks are not too safe enough to satisfy people needs. A common man feels his valuables are secured if there is efficiency in security. Hence this project can give effective security in minimal cost.

Index-Terms: Arduino, Servo motor, LCD 16x2, 4x4 Membrane Keypad, Buzzer.

1.0 INTRODUCTION

In this project we are providing enough security to satisfy the user’s needs. The user will be prompted to enter a password to unlock the door. On successful password entry, the door unlocks for a specified amount of time enabling him/her to store or restore his/her valuables. On the other hand, if the user enters an invalid password then corresponding equivalent message will be displayed.

This project “Arduino based password protected locking system” can be used to provide enough security in various places like bank lockers, security doors, BIOS locking in computer etc.

This project uses an arduino kit that consists of ATmega 328 which is one of the most popular microcontrollers that consists of 14 digital pins and 6 analog general purpose pins, EEPROM of capacity 1KB and a ram of 2KB.

2.0 COMPONENTS USED IN DESIGNING

We will be providing the detailed description of every component used in designing this password protected locking system:

2.1 Arduino UNO

This microcontroller is based on the ATmega 328. There are total of 20 pins (0-19) out of which 6 are analog inputs which can also be used as general purpose pins, a ceramic resonator of frequency 16MHz, an USB connection, a power jack and a reset button. It contains everything needed to support a microcontroller. [1].

2.2 LCD

Liquid Crystal Display, which we are using in our project is JHD 1602A. This display consists of 16 columns and 2 rows. The library that is used is <liquidcrystal.h>.

PIN SUMMARY OF LCD 1602A

Pin 1: VSS.
Pin 2: To VDD 5V input.
Pin 3: VL to adjust LCD contrast with the help of 10K potentiometer. Low VL indicates light contrast and high VL indicates dark contrast.
Pin 4: RS for register select. Data registers used for high RS. Similarly, instruction register for low RS.
Pin 4: RS for register select. Data registers used for high RS. Similarly, instruction register for low RS.
Pin 5: R/W signal stands for read/write. When R/W bit is high, it indicates a read operation. If R/W bit is low, it indicates write operation.
Pin 6: Clock Enable- Edge triggering.
Pin 7 to 14: Represents from Bit 0 to Bit 7.
Pin 15: back light Anode.
Pin 16: back light cathode.

\textsuperscript{1}\textsuperscript{3}\textsuperscript{4}\textsuperscript{5}\textsuperscript{6}Department of Computer Science & Engineering, Nitte Meenakshi Institute of Technology, Bangalore-560064, India

\textsuperscript{4}\textsuperscript{5}\textsuperscript{6}Department of Computer Science & Engineering, Muthoot Institute of Technology & Science, Varikoli, Ernakulam, India
2.3 Membrane Keypad
In our project we used 4X4 matrix membrane keypad. This 16 button keypad provides user interface component for Arduino project. This is programmed using the library <keypad.h>. It has the following features:
1. Easy interface to Arduino.
2. Ultra-thin design.
3. Cheap and economical.

SUMMARY about Keypad pins:
1. Maximum operation rating: 24VDC, 30 mA.
2. Insulation Resistance: 100M ohm
3. Interface: 8 pins can be accessed in the form of 4X4 matrix.

2.4 Servo Motor: The servo used in the project is SG90 Micro Servo weighing about 9g. This is programmed using the library <servo.h>. It has the following operating conditions:

- Modulation: Analog
- Torque: 25.0 oz-in (1.8kg/cm)
- Speed: 0.12 s/60 deg
- Weight: 0.32 oz (9.8g)
- Motor type: 3 pole
- Gear type: Plastic
- Rotation/Support: Bushing
- Pulse Width: 500-2400 micro-sec

2.5 Potentiometer
In our project we have used a trim potentiometer of 10Kohm resistance in order to adjust the contrast of the Liquid crystal display. It is as shown below:
3.0 IMPLEMENTATION - CONNECTION WIRING SPOT DIAGRAM AND CIRCUIT DIAGRAM

The following table shows the connection wiring spot diagram of our circuit. The entire circuit diagram can be tabulated as shown below. This tabulation is called connection wiring spot diagram. The entire circuit connection is being available in this table. Labeling of the pins as per the above tabulation:

- **D’N’**: D- Digital pins in arduino. N is the pin number
- **A’N’**: A- Analog pins in arduino. N is the pin number
- **LCD’N’**: Liquid Crystal Display pins. N is the pin number
- **DB0-DB7**: Data byte pins from 0-7
- **R/W**: Read/Write
- **VD0 and LED+**: Positive connection of the LCD
- **VD1 and LED-**: Negative connection of the LCD.
- **ROW’N’-**: Rows of the hex keypad. N is the pin number
- **COL’N’-**: Columns of the hex keypad. N is the pin number
- **Motor**: Servo motor Connection
- **MICRO**: Arduino UNO Connection
- **FR**: From Connection
- **TO**: To connection
- **E**: Enable
- **RS**: Reset

The following figure shows the schematic diagram for the project. The components used in design and connections of the project are as follows:
- **ARDUINO UNO**
- **LIQUID CRYSTAL DISPLAY(LCD JHD1602A)**
- **4X4 MEMBRANE KEYPAD**
- **SERVO MOTOR**
- **10K POTENTIOMETER**

Note: all the description of the above components is provided in section 2.1-2.6.

The step by step working is given as below. When the arduino is switched on, the LCD displays the entry screen message by initializing and configuring the LCD pins to arduino.

**Step 1**: Configure the lcd pins.

```c
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
```

**Step 2**: Initialize the correct password

```c
char* pass="A1B2C";
```

The step by step working is given as below. When the arduino is switched on, the LCD displays the entry screen message by initializing and configuring the LCD pins to arduino. Next, the user is asked to prompt a password. Here the correct password is pre-initialized. Next, the user is prompted to enter the password. The user enters the password through a keypad which is read by the arduino. Now the entered password is checked with the predefined password. If the password matches, then the servo motor deflects and the door unlocks for 30s else the buzzer beeps indicating the invalidity of the password.
STEP 3:

```c
myservo.write(90); // The servo motor deflects to an angle of 90
degrees enabling the user to unlock
unlockdoor(); // Unlocks the door for a specified amount of time
currpos=0; // reset the password enabling the user to enter a
new password
myservo.write(0); // after the time exceeds the servo deflects the
angle back to zero degrees.
```

Else, the following set of statements will be executed

```c
myservo.write(0); // Due to the entry of wrong password, the
servo does not deflect and hence the door will // be locked
invalidcode(); // Message of invalidity will be displayed to the
user via lcd and returns to the start
currpos=0; // the password is reset enabling the user to enter a
fresh password
```

In the above case, the door will be unlocked by the movement
of servo to a particular angle or remaining still depending upon
the user's entered password.

**Note:** The entered password by the user is converted into
```
"**" to provide strong privacy.
```  

```c
For(l=0;l<=currpos;++l)
{
    lcd.print ('*');
}
```

Further, the buzzer is provided if the user enters a wrong
password and also if the user exceeds the specified limit. Here
we have given the specified limit to be 20 secs.

**Step 4:**

```c
if(i==21){
digitalWrite(19,HIGH);// buzzer beep
lcd.setCursor(0,0);
```

**5.0 HARDWARE OUTPUT SCREEN SHOTS**

**Figure 9.0. Showing user to enter the password**

In the above figure, Fig 9.0. and Fig 9.1 the LCD displays the
user “ENTER PASSWORD”. The entered password is displayed as ‘*’ on the LCD.

In these figures, fig 10.0., fig 10.1., the user has entered the
correct password.

The servo motor deflects thus unlocking the door.

**Figure 10.0. Showing correct password**

**Figure 10.1. Door unlocks**

**Figure 11.0. Timer activated**

**Figure 11.1. Deadline timer**
In these figures, fig 11.0, fig 11.1., the timer activates automatically for fixed duration. The buzzer beeps if the user exceeds the grace period and deadline timer is activated.

**Figure 12. Door locks.**

In the Fig 12., the door locks when the timer crosses the deadline time. The user is prompted to enter the password once more and the process continues. Fig 9.0.

**Figure 13.0. Incorrect password entry.**

In these figures, fig 13.0., fig 13.1., the user has entered an incorrect password. As a result the door remains locked and the user is prompted to enter the password once more.

**Figure 13.1. The door remains locked**

6.0 CONCLUSION

This project is effective in providing enough security as long as the password is not shared. In future this “Arduino based password security locking system” can be provided maximum security by the above enhancements in order to completely satisfy user’s needs. Hence, a common man can afford to purchase such locking system in minimal cost to keep his valuables safely without any worries.

**ACKNOWLEDGEMENT**

The project on the locking system would be tedious without the help and support of the under mentioned people. Firstly, we would like to thank the Department of Computer Science & Engineering, Nitte Meenakshi Institute of Technology, Bangalore for providing us with the opportunity to perform and apply our knowledge by developing a ‘Password Protected Locking System Using Arduino’.

We would like to express our deep gratitude to Dr. N R Shetty, Advisor, Nitte Meenakshi Institute of Technology, Prof. H C Nagaraj, Principal, Nitte Meenakshi Institute of Technology, Dr. Venkatesh, Head of Department, Bangalore for their enthusiastic encouragement and willingness to provide us their valuable time so generously.

Lastly, we would like to thank our families and friends who provided us with resources and their encouragement which lead to the successful completion of this project.

**REFERENCES**

[4]. Getting started with Arduino, Mazzimo Banzi(co-founder), 2nd edition 2011, O’Reilly  
[6]. Introduction to Arduino, a piece of cake, Alan G Smith, September 30 2011  
[7]. M. Margolis, Arduino Cookbook, O’reilly, CA, USA, 2011  
[9]. The working principle of an Arduino, Abuja, Electronics, Computer and Computation (ICECCO), 2014 11th International Conference, IEEE  
[10]. An RF Based relay for control and communication for systems for unmanned ground vehicle and air vehicle, 2nd international conference on computing for sustainable global development, BVICIAM, New Delhi, India, 03/2015
Table 1. Connection wiring spot diagram

<table>
<thead>
<tr>
<th>Motor</th>
<th>Micro</th>
<th>LCD</th>
<th>Keypad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FR</td>
<td>TO</td>
<td>FR</td>
</tr>
<tr>
<td>1</td>
<td>-ve Gnd</td>
<td>3.3V</td>
<td>1 VD1</td>
</tr>
<tr>
<td>2</td>
<td>Pulse</td>
<td>MICRO A2</td>
<td>5V</td>
</tr>
<tr>
<td>3</td>
<td>+ve 5V</td>
<td>Vin</td>
<td>3 WD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 RS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6 E</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7 DB0</td>
</tr>
<tr>
<td>A0</td>
<td>KP COL2</td>
<td>8 DB1</td>
<td>ROW4 MICRO D4</td>
</tr>
<tr>
<td>A1</td>
<td>9 DB2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>MOTOR Pulse</td>
<td>10 DB3</td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>Gnd</td>
<td>11 DB4</td>
<td>MICRO D11</td>
</tr>
<tr>
<td>A4</td>
<td>12 DB5</td>
<td>MICRO D10</td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>Buzzer +ve</td>
<td>13 DB6</td>
<td>MICRO D9</td>
</tr>
<tr>
<td>N/C</td>
<td>14 DB7</td>
<td>MICRO D8</td>
<td></td>
</tr>
<tr>
<td>GND 100K POT</td>
<td>LCDVD1,MOTOR NEG, Buzzer -ve</td>
<td>15 LED+ RES-5V-100KPOT</td>
<td></td>
</tr>
<tr>
<td>D0</td>
<td>16 LED-</td>
<td>Gnd</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>KPCOL3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>KPCOL4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D4</td>
<td>KPROW4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D5</td>
<td>KPROW3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D6</td>
<td>KPROW2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D7</td>
<td>KPROW1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8</td>
<td>LCDB7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D9</td>
<td>LCD DB6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D10</td>
<td>LCD DB5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D11</td>
<td>LCD DB4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D12</td>
<td>LCD E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D13</td>
<td>LCD R5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>