IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 4, Issue 2, February 2017

Customized Door Lock System

Tushar Diggewadi[1], Shubhangi Patil[2], Noor Mohmmed Khan[3], Anand Gudnavar[4]

U.G. Student, Electronics and Communication, Dr. M.S.S. College of Engg and Tech, Belagavi, India[1, 2, 3]

Head of Department, Computer Science and Engg, Shaikh College of Engg and Tech, Belagavi, India[4]

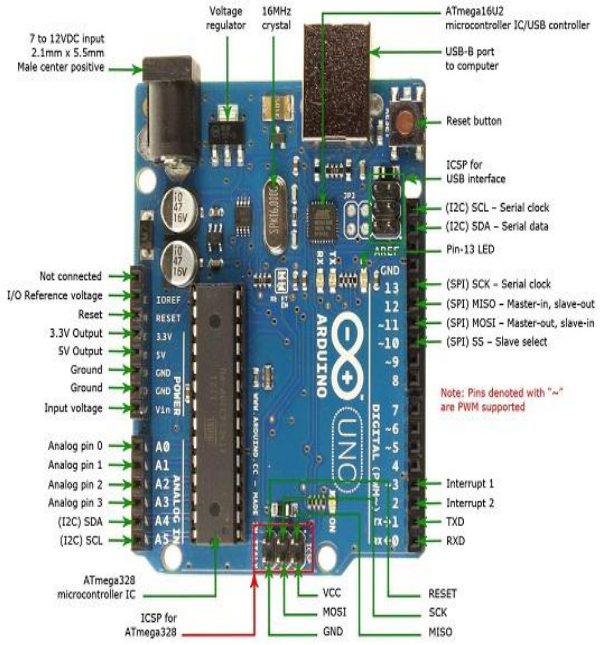
Abstract: Customized door lock system is a replacement to the traditional mechanical locks which most of the houses used earlier. This locking system using Arduino is a electro-mechanical device designed in a software with an ease for the user to set and reset the password of their own choice whenever necessary.

Keywords: Arduino UNO, Digital Door Lock, Interfacing with LCD, Programming in C.

I. INTRODUCTION

What is Arduino?
Arduino UNO is a open source and easy to use microcontroller board which consists of ATmega328p. it has 14 digital pins which can be used as input/output pins (of which 6 ca n be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

To get started connect the USB to the computer or power it with the ac-to-dc adapter or battery.

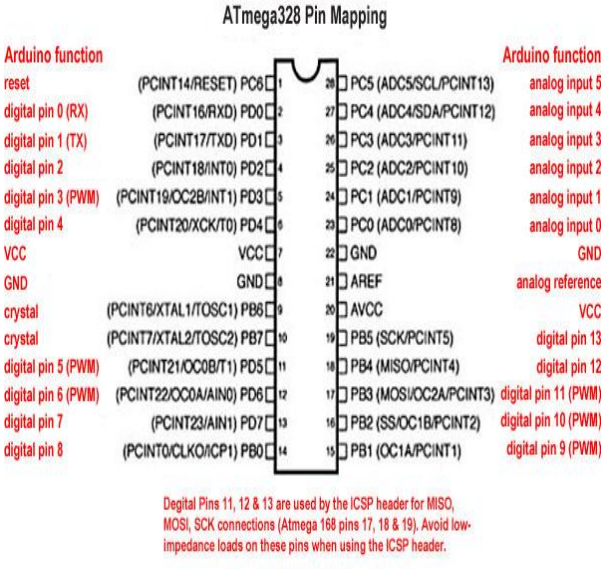


Why Arduino?
Arduino has a simple and accessible user experience. It has been used in many projects and applications. Arduino programming is simple enough to a beginner and flexible enough to an advanced user. It is one of the most used microcontroller board in low cost (prototype) robotics. The advantage of Arduino software (IDE) is that it works on Windows, Mac and Linux. Other IDE's are not supported in other OS like Mac and Linux.

II. TECHNICAL SPECIFICATION

Microcontroller	ATmega328p
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega 328P) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Specification Table of ATmega 328P [1].



IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 4, Issue 2, February 2017

III.ADVANTAGES

1) **Inexpensive** – Arduino is less expensive than other prototyping boards and widely available.
2) **Cross-platform** –Arduino software (IDE) is compatible with Windows, Linux, Mac where as others are limited to Windows.
3) **Simple programming language** – Arduino has a simple programming language which is easy for beginners and flexible enough for advance users.
4) **Open source hardware and software –** Arduino hardware as well as software are open source, where experienced developer can expand the language through C++ libraries. Similarly the plans of Arduino board are published under a Creative Commons license, so experienced circuit designers can make their own version of Arduino.

IV.DIGITAL DOOR LOCK

Digital code lock or digital combination lock is a such a type where a combination of digits or characters or both are used for unlocking the lock. There are separate keys for locking and unlocking the system. The system can be unlocked by pressing the unlock button after entering the correct combination of digits. A hex keypad is used as the input device. The user has to enter the password at the installation time and can change it at any point of time in the future. Locking and unlocking is done in a single press and the password can be changed only if the current password is entered correctly.

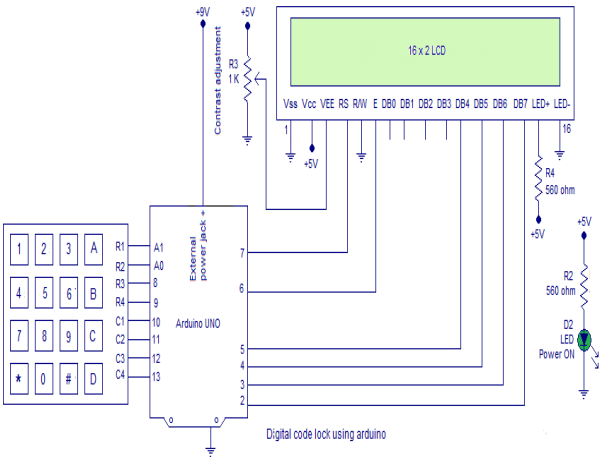


Fig 2- Digital lock code using Arduino

V. INTERFACING HEX KEYPAD TO ARDUINO

Hex keypad is a very important component in Embedded Systems and is usually applicable in code locks, calculators, automation systems or any application that requires a character or numeric input.
Hex keypad has 16 push button switches in a 4X4 matrix form. Typically a hex keypad will have keys for number 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and letters A, B, C, D, *, #. The hex keypad will have 8 connection wires namely R1, R2, R3, R4 and C1, C2, C3, C4 representing the rows and columns respectively. As in, microcontroller or microprocessor the key press is identified by a method called column scanning. Here a particular row is kept low and others are held high. The logic status of each column line is scanned. If a particular column is found low, then that means the key that comes in between that column and row is short (pressed). Then the program registers that key being pressed. Then the same procedure is applied for the subsequent rows and the entire process is repeated. For example if row 1 is kept low and column 1 is found low during scanning, then that means key"1″ is pressed.

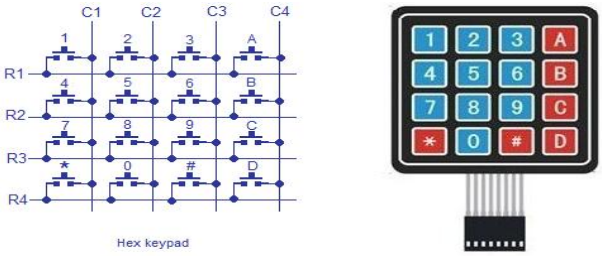


Fig 2.1- Hex keypad

VI. INTERFACING LCD TO ARDUINO

A Liquid Crystal Display commonly abbreviated as LCD is basically a display unit built using Liquid Crystal technology. To display output in our project we need a medium to display our output values and messages. The first basic option would be a 7 segment display which is limited to its own limitations. Hence the next best device is the LCD which is available in different size and specifications out of which 16x2 is the most commonly used. It can display 32 ASCII characters in 2 lines (16 characters in 1 line).

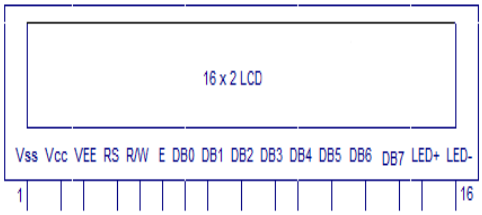


Fig 2.2(a) - LCD

The RS pin of the LCD module is connected to digital pin 12 of the Arduino. R/W pin of the LCD is grounded. Enable pin of the LCD module is connected to digital pin 11 of the Arduino. Here the LCD module and Arduino are interfaced in the 4-bit mode. This means only four of the digital input lines (DB4 to DB7) of the LCD are used. Digital lines DB4, DB5, DB6 and DB7 are interfaced to digital pins 5, 4, 3 and 2 of the Arduino. A 1kΩ resistor is for the contrast of the display. 560Ω resistor R1 limits the current through the back light LED. The Arduino can be powered through the external power jack provided on the board. +5V (5 Volts) required in some other parts of the circuit can be tapped from the 5V source on the Arduino

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 4, Issue 2, February 2017

board. The Arduino can be also powered from the PC through the USB port.

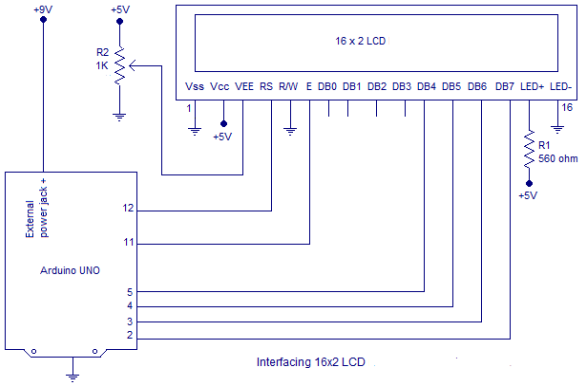


Fig 2.2(b) – Interfacing Arduino with LCD

VII. INTERFACING LCD TO ARDUINO

I will be explaining important points about the program only. As mentioned before, you are supposed to know the codes of interfacing hex keypad and LCD module.

Installation – You will be asked to input **5** digits as password at the initial boot/reset of the device. The first 5 digits you input at installation will be saved as your set or initial password**.** The device will go locked after setting password.

Key A – for unlocking the device. Input correct password and press A for Unlocking.

Key B – for locking any time. Just press B and you will see the locked message.

Key C – for changing the password. Input the correct password and Press C. You will see message asking to ENTER NEW PASSWORD. Enter **5** digits as password. The first 5 digits you enter will be saved as new password.

Exceptions – You cannot use keys 'A', 'B' and 'C' inside the password combination. These 3 keys are control keys of the device with specific functions. The program checks for these key presses (at the password setting time – you may see the SetPassword() function and look the condition to check for invalid key press**)** and identifies them as invalid keys. You will have to input 5 new digits as password after an invalid key press.

pass[6] – is the array used to save and hold the user defined password.

check[6] – is the array used to collect & hold user input. This user input data (in check[] array) is compared with pass[] array to authenticate password.

entry – is the variable used to identify initial entry point of the program. User is asked to set a 5 digit password at installation of lock. Hence we need a variable to identify entry and loop 5 times to collect **5** digits and save them to pass[] array. The same variable is later made use of to change password. When the key for changing password (here 'C') is pressed, this variable is simply assigned a zero value (the initial state of variable). This forces the program control to re enter the password setting loop of the program.

key_id – is the variable used to identify a key press and perform some actions in the program (that should happen only on a key press). By default this variable is set zero initial value. Whenever a key is pressed in key pad, this variable will be assigned a value =1**.** You may check the keyscan() function to see this. This simple trick helps to identify a key press and perform various actions on that key press (based on the value of key press). This variable is set to zero at different points in the program (to prevent the value 1 in key_id variable being identified as a false key press). You may check them as well.

Note:-
col_scan – is the actual variable that gets activated to a low on key press (hence helps in identifying key press). But this variable is actually a part of the key pad interfacing program (version 2).

count – is the variable used to iterate the index of check[count] (user input array). count variable is initialized to 1. Each user input will be saved to check[] array in order of the increment of count variable.

temp_press – is a temporary variable to hold the value of key press. The value of key press is assigned to temp_press variable as a return result of the keypress() function. keypress() is the function defined to identify value of key press.

lcd_count – is a simple counter variable used to iterate the column position of LCD module. This variable helps to display user input data successively in row 2 of LCD module.
i, j, flag – are just dummy variables used in the program. i, j are used as counter variables inside for loop. flag is used to hold status of checkPassword() subroutine (the function used to compare user input data and the set password). A decision is made based on the value inside flag variable.

VIII. SUBROUTINES USED IN THE PROGRAM

SetPassword() – is the subroutine used to set user defined password. This subroutine is very dependent on the **"**Password Setting Loop**"** written inside the main program.

This password setting loop will be iterated at installation of the device (that is at the boot or reset of the device) for first **5** key presses. This first 5 key press will be used to set the password. These key presses will be saved to pass[] array. As mentioned earlier, entry is the variable

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 4, Issue 2, February 2017

used to iterate the loop **5** times. key_id is the variable used identify key press.

Note:- The same **"**Password Setting Loop**"** is made use of for Changing the Password as well. When key 'C' is pressed, the current password is checked for. If the input password is matching with current set password, then entry variable will be assigned to zero value. This will simply transfer the control of the program to enter the **"**Password Setting Loop**"** again.

keyscan() – is the subroutine to scan keypad for a key press. This subroutine is basically same as the version 2 code of interfacing hex keypad to Arduino. I have added some lines of code needed for this code lock. Apart from that, the lines of code in this subroutine is same as that of interfacing keypad. keyscan() subroutine scans for a key press (whenever the function is called from Main program or from other sub routines like SetPassword()) and identifies the row and column of the pressed key. If key '1' is pressed, keyscan() identifies that key at row 1 and column 1 is pressed. Similarly if key '6' is pressed, the keyscan() identifies a key is pressed at row 2 and column 3. Whenever a key is pressed, another subroutine named keypress() is invoked within the keyscan() routine. This keypress() routine is used to identify the value of key press (say '1', '2', '3' or 'A', 'C' or 'D' etc)

keypress() – as mentioned above is the subroutine to identify value of key press. The keyscan() routine identifies which row and column of key pad is pressed. This row and column number is passed to keypress() routine as parameters (using variable values of i and j).

checkPassword() – is the subroutine to check user input password against the set user defined password. The user input data (password to cross check) is collected in the check[] array. This is compared against the set password inside pass[] array. A for loop is used for comparing. If each digit inside the arrays match, flag variable will remain zero. If any mismatch occurs, the flag will be set to 1 and loop will break.
So that's all you need to know about the program. Read the program below.

IX. PROGRAM

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(7,6,5,4,3,2);
int row[]={A1,A0,8,9};
int col[]={10,11,12,13};
int i,j,lcd_count,count=1,key_id=0,flag,entry=0;
int col_scan;
char temp_press;
char check[6],pass[6];

void setup()
{
lcd.begin(16,2);
for(i=0;i<=3;i++)
{
pinMode(row[i],OUTPUT);
pinMode(col[i],INPUT);
digitalWrite(col[i],HIGH);
}
lcd.print("SET 5 Digit PASS");
}

void loop()
{
 while(entry<=4)
{
SetPassword();
}

key_id=0;
keyscan();
if(key_id==1)
{
 check[count]=temp_press;
 count++;
if(temp_press=='A')
{
  checkPassword();
if(flag==0)
{
lcd.setCursor(0,0);
lcd.print("UNLOCKED");

}
else
{
lcd.setCursor(0,0);
lcd.print("WRONG PASSWORD");
delay(200);
lcd.clear();
lcd.print("LOCKED");
}
count=1;
}

else if(temp_press=='C')
{
checkPassword();
if(flag==0)
{
lcd.setCursor(0,0);
lcd.print("ENTER NEW PASS");
key_id=0;
entry=0;
}else{
lcd.setCursor(0,0);
lcd.print("WRONG PASSWORD");
}
count=1;
}

else if(temp_press=='B')
```

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 4, Issue 2, February 2017

```
{
lcd.setCursor(0,0);
lcd.print("LOCKED");
count=1;
}
}
}

void SetPassword()
{
keyscan();
if(key_id==1)
{
if(temp_press=='A'||temp_press=='C'||temp_press=='B')
{
lcd.setCursor(0,0);
lcd.print("INVALID KEYS");
entry=0;
}
else

{
  pass[entry]=temp_press;

}
}
key_id=0;
if(entry==5)
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("PASSWORD SET & LOCKED");
}}

char keyscan
{
for(i=0; i<=3; i++)
{
digitalWrite(row[0],HIGH);
digitalWrite(row[1],HIGH);
digitalWrite(row[2],HIGH);
digitalWrite(row[3],HIGH);
digitalWrite(row[i],LOW);
for(j=0; j<=3; j++)
{
col_scan=digitalRead(col[j]);
if(col_scan==LOW)
{
  key_id=1;
  entry++;
temp_press=keypress(i,j);
lcd.setCursor(lcd_count++,1);
lcd.print(temp_press);
if(temp_press=='A'||temp_press=='C'||temp_press=='B')
{
lcd_count=0;
lcd.clear();
}
delay(300);
break;
}}

}}

char keypress(int i, int j)
{
if(i==0&&j==0)
{
return('1');
}
if(i==0&&j==1)
{
return('2');
}
if(i==0&&j==2)
{
return('3');
}
if(i==0&&j==3)
{
return('A');
}
if(i==1&&j==0)
{
return('4');
}
if(i==1&&j==1)
{
return('5');
}
if(i==1&&j==2)
{
return('6');
}
if(i==1&&j==3)
{
return('B');
}
if(i==2&&j==0)
{
return('7');
}
if(i==2&&j==1)
{
return('8');
}
if(i==2&&j==2)
{
return('9');
}
if(i==2&&j==3)
{
return('C');
}
if(i==3&&j==0)
{
return('*');
}
```

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 4, Issue 2, February 2017

```
if(i==3&&j==1)
{
return('0');
}
if(i==3&&j==2)
{
return('#');
}
if(i==3&&j==3)
{
return('D');
}
}

void checkPassword()
{
flag=0;
for(i=1;i<=5&&flag==0;i++)
{
if(check[i]==pass[i])
{
flag=0;
}
else
{
flag=1;
}}}
```

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our Principal **Dr. Basavraj G. Katageri** for endowing us with his valuable time and knowledge. We would like to thank our Head of the Department **Prof. S.B. Kulkarni** for his keen interest and concern towards our publication.

CONCLUSION

Arduino based door locking system is used where we need more security. It can also be used for lockers and protective doors. The system comprises of a number keypad and that is connected to the Arduino board. This board monitor's the keypad and if somebody enters the password it will check with the pre entered password and unlock the door if its correct. Hence the system allows only that person who knows the password.

REFERENCES

[1] " Arduino Tutorials"-https://www.arduino.cc/tutorial
[2] Getting started with Arduino, Mazzimo Banzi (co-founder), 2nd edition 2011,O'Reilly.
[3] "Arduino Door Lock with Password"-https://instructable.com/id/Arduino-door-lock-with-password/
[4] "Interface LCD to Arduino"-https://alselectro.wordpress.com/tag/interface-lcd-to-arduino/
[5] "Digital code lock using Arduino with LCD display "-https://www.circuitstoday.com/advanced-digital-code-lock-using-arduino.