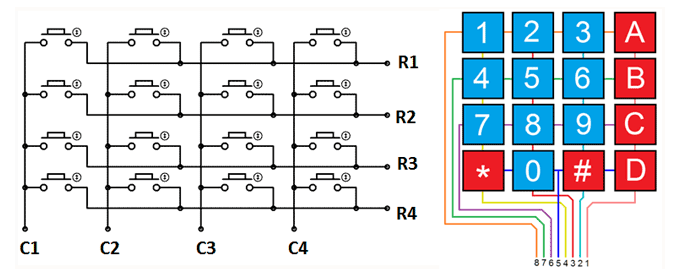
**Guide for Digital Lock + LCD**

1. **Requirements**

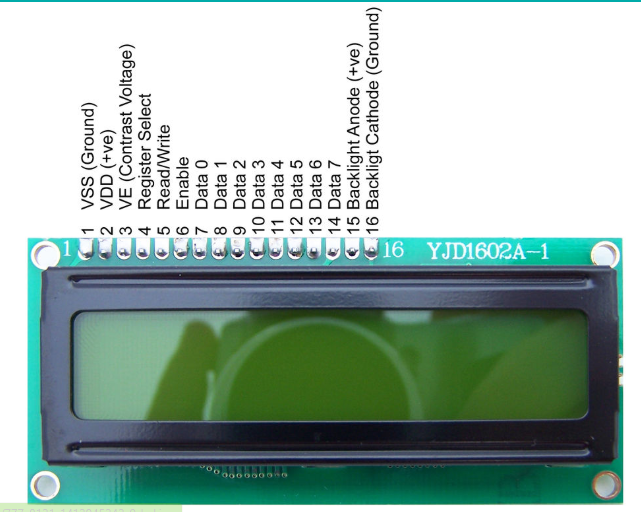
* You have to create a digital lock with Arduino, keypad 4x4 and a servo motor (to move the latch for locking or unlocking)
* Your lock has a default password is 0000
* You lock can change password
* The # key will be used as function key
  + - Press # key less than 3s: call the unlock function
    - Press # key more than 3s: call the change password function
* LCD will be used to dispaly the lock status and password character input (display character as a \* )

1. **Create Circuit**



|  |  |  |
| --- | --- | --- |
| Arduino I/O pin | Keypad | Keypad Pin |
| 6 | R1 | 8 |
| 7 | R2 | 7 |
| 8 | R3 | 6 |
| 9 | R4 | 5 |
| 10 | C1 | 4 |
| 11 | C2 | 3 |
| 12 | C3 | 2 |
| 13 | C4 | 1 |

**LCD**



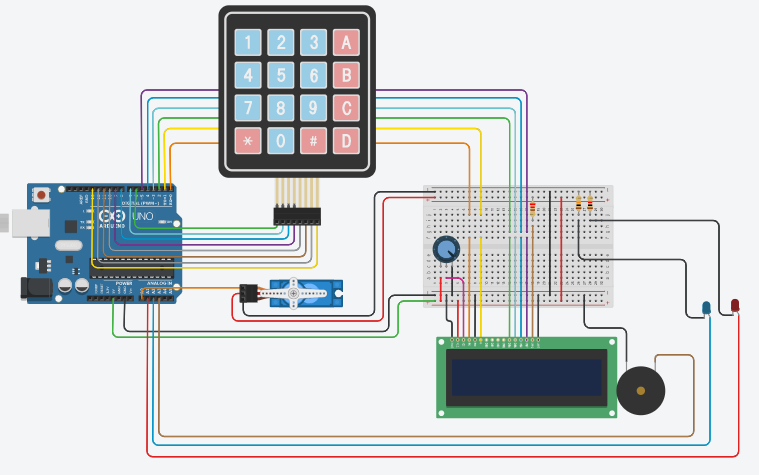
**Pin Configuration**

|  |  |  |
| --- | --- | --- |
| **Pin No:** | **Pin Name:** | **Description** |
| 1 | Vss (Ground) | Ground pin connected to system ground |
| 2 | Vdd (+5 Volt) | Powers the LCD with +5V (4.7V – 5.3V) |
| 3 | VE (Contrast V) | Decides the contrast level of display. Grounded to get maximum contrast. |
| 4 | Register Select | Connected to Microcontroller to shift between command/data register |
| 5 | Read/Write | Used to read or write data. Normally grounded to write data to LCD |
| 6 | Enable | Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgement |
| 7 | Data Pin 0 | Data pins 0 to 7 forms a 8-bit data line. They can be connected to Microcontroller to send 8-bit data.  These LCD’s can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free. |
| 8 | Data Pin 1 |  |
| 9 | Data Pin 2 |  |
| 10 | Data Pin 3 |  |
| 11 | Data Pin 4 |  |
| 12 | Data Pin 5 |  |
| 13 | Data Pin 6 |  |
| 14 | Data Pin 7 |  |
| 15 | LED Positive | Backlight LED pin positive terminal |
| 16 | LED Negative | Backlight LED pin negative terminal |

We used 4 bits mode: D4, D5, D6, D7 for receiving data from UNO board

|  |  |
| --- | --- |
| Arduino I/O pin | LCD Pin |
| 0 | 4 - Register Select |
| 1 | 6 - Enable |
| 2 | 11 - Data 4 |
| 3 | 12 - Data 5 |
| 4 | 13 - Data 6 |
| 5 | 14 - Data 7 |

**SCHEMATIC**



1. **Code the SKETCH**: (My code have no function for LED and Buzzer alarm, you will add to this sketch later)

#include <Keypad.h>

#include <Servo.h>

#include <string.h>

#include <LiquidCrystal.h>

/\*

Lib: https://github.com/Chris--A/Keypad/archive/master.zip

\*/

// Define some constants and variables

const byte rows = 4;

const byte columns = 4;

int holdDelay = 700;

int n =3;

int state = 0;

char key = 0;

int pos = 0;

//Default password is 0000

String default\_passwd = "0000";

//variable to store the user input for password

String input\_passwd = "";

//Define keys for lock and unlock or change password function

char lock\_key = '\*';

char unlock\_key = '#';

char change\_pass\_key = '-'; //press \* key for more than 3 second

// Create an instance for servo motor

Servo servo\_A0;

//Define characters matrix

char keys[rows][columns] =

{

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'},

};

//Define pins for every row of keypad

byte rowPins[rows] = {6, 7, 8, 9};

//Define pins for every column of keypad

byte columnPins[columns] = {10, 11, 12, 13};

// Create an instance for our keypad

Keypad keypad = Keypad(makeKeymap(keys), rowPins, columnPins, rows, columns);

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(0, 1, 2, 3, 4, 5);

// Define function for key

char function\_key(int n)

{

char temp = keypad.getKey();

if ((int)keypad.getState() == PRESSED)

{

if (temp != 0) {key = temp;}

}

if ((int)keypad.getState() == HOLD)

{

state++;

state = constrain(state, 1, n);

delay(holdDelay);

}

if ((int)keypad.getState() == RELEASED)

{

key += state;

state = 0;

}

delay(100);

//Serial.println(key);

return key;

}

// Define function input\_password

String input\_password(int num\_char)

{

String passwd = "";

//Serial.print("Input password: ");

do

{

char temp = keypad.getKey();

if (temp != 0)

{

//Serial.print(temp);

LCD\_display(passwd.length(), 1, "\*");

passwd += temp;

}

delay(100);

}

while (passwd.length() < num\_char);

//Serial.println();

return passwd;

}

// Define function change\_password

String Change\_password(int num\_char, String current\_passwd)

{

//Authenticate the old password:

//Serial.print("Input old password: ");

LCD\_display(0, 0, "OLD PASSWORD:");

String old\_passwd = input\_password(num\_char);

if (old\_passwd != current\_passwd)

{

lcd.clear();

LCD\_display(0, 0, "WRONG PASSWORD!");

//Serial.println("Password does not match! Nothing changes");

return current\_passwd;

}

//New password

//Serial.print("Input new password: ");

lcd.clear();

LCD\_display(0, 0, "NEW PASSWORD:");

String new\_passwd = input\_password(num\_char);

//Confirm passwd

//Serial.print("Input new password again: ");

lcd.clear();

LCD\_display(0, 0, "CONFIRM PASSWORD");

String confirm\_passwd = input\_password(num\_char);

if (confirm\_passwd == new\_passwd)

{

//Serial.println("Password has changed!!!");

lcd.clear();

LCD\_display(0, 0, "CHANGED PASSWORD");

return confirm\_passwd;

}

else

{

//Serial.println("Password does not match! Nothing changes");

lcd.clear();

LCD\_display(0, 0, "NOTHING CHANGES!");

return current\_passwd;

}

}

void Unlock()

{

//Serial.print("Input password: ");

lcd.clear();

LCD\_display(0, 0, "INPUT PASSWORD:");

input\_passwd = input\_password(4);

if (input\_passwd == default\_passwd)

{

lcd.clear();

LCD\_display(0, 0, "OPENNING!!!");

//Unlock by servo\_A0

for (pos = 0; pos <= 180; pos += 1)

{

// tell servo to go to position in variable 'pos'

servo\_A0.write(pos);

// wait 15 ms for servo to reach the position

delay(15); // Wait for 15 millisecond(s)

}

delay(3000); //open door 3s then close

//lock by servo\_A0

lcd.clear();

LCD\_display(0, 0, "CLOSING!!!");

for (pos = 180; pos >= 0; pos -= 1)

{

// tell servo to go to position in variable 'pos'

servo\_A0.write(pos);

// wait 15 ms for servo to reach the position

delay(15); // Wait for 15 millisecond(s)

}

}

else

{

//Serial.println("Wrong password!");

LCD\_display(0, 0, "WRONG PASSWORD!");

}

//Reset input\_passwd

input\_passwd = "";

key = 0;

}

// Define LCD display function

void LCD\_display(int column, int line, String message)

{

// set the cursor to column 0, line 1

// (note: line 1 is the second row, since counting begins with 0):

lcd.setCursor(column, line);

// print the number of seconds since reset:

lcd.print(message);

}

void setup()

{

//Setting serial with baudrate 9600

//Serial.begin(9600);

// connect signal pin of servo to pin number 4 on Uno

servo\_A0.attach(A0);

servo\_A0.write(pos);

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

LCD\_display(0, 0, "Welcome!");

}

void loop()

{

lcd.clear();

LCD\_display(0, 0, "STATUS: LOCKED!");

LCD\_display(0, 1, "UNLOCK: PRESS #");

while (key !=unlock\_key & key != change\_pass\_key)

{

key = function\_key(n);

//lcd.setCursor(8, 1);

// print the key

//lcd.print(key);

}

if (key == unlock\_key)

{

Unlock();

}

if (key == change\_pass\_key)

{

default\_passwd = Change\_password(4, default\_passwd);

delay(2000);

key =0;

}

}