**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

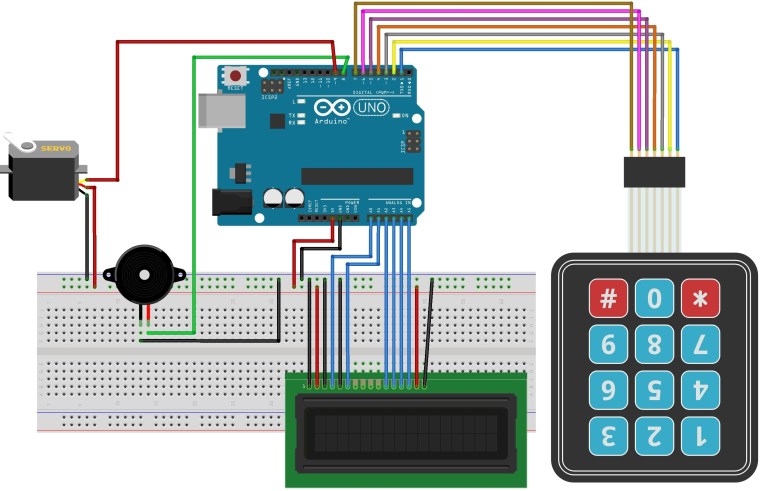
COL’N’- Columns of the hex keypad. N is the pin number

Motor- Servo motor Connection

MICRO- Arduino Connection

E- Enable

RS: Reset



**CODE :**

#include <Keypad.h>

#include <LiquidCrystal.h>

#include <Servo.h>

Servo myservo;

LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);

#define Password\_Lenght 7 // Give enough room for six chars + NULL char

int pos = 0; // variable to store the servo position

char Data[Password\_Lenght]; // 6 is the number of chars it can hold + the null char = 7

char Master[Password\_Lenght] = "123456";

byte data\_count = 0, master\_count = 0;

bool Pass\_is\_good;

char customKey;

const byte ROWS = 4;

const byte COLS = 3;

char keys[ROWS][COLS] = {

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

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

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

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

};

bool door = true;

byte rowPins[ROWS] = {1, 2, 3, 4}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {5, 6, 7}; //connect to the column pinouts of the keypad

Keypad customKeypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS); //initialize an instance of class NewKeypad

void setup()

{

myservo.attach(9);

ServoClose();

lcd.begin(16, 2);

lcd.print(" Arduino Door");

lcd.setCursor(0, 1);

lcd.print("--Look project--");

delay(3000);

lcd.clear();

}

void loop()

{

if (door == 0)

{

customKey = customKeypad.getKey();

if (customKey == '#')

{

lcd.clear();

ServoClose();

lcd.print(" Door is close");

delay(3000);

door = 1;

}

}

else Open();

}

void clearData()

{

while (data\_count != 0)

{ // This can be used for any array size,

Data[data\_count--] = 0; //clear array for new data

}

return;

}

void ServoOpen()

{

for (pos = 180; pos >= 0; pos -= 5) { // goes from 0 degrees to 180 degrees

// in steps of 1 degree

myservo.write(pos); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

} }

void ServoClose()

{

for (pos = 0; pos <= 180; pos += 5) { // goes from 180 degrees to 0 degrees

myservo.write(pos); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

}

}

void Open()

{

lcd.setCursor(0, 0);

lcd.print(" Enter Password")

customKey = customKeypad.getKey();

if (customKey) // makes sure a key is actually pressed, equal to (customKey != NO\_KEY)

{

Data[data\_count] = customKey; // store char into data array

lcd.setCursor(data\_count, 1); // move cursor to show each new char

lcd.print(Data[data\_count]); // print char at said cursor

data\_count++; // increment data array by 1 to store new char, also keep track of the number of chars entered

}

if (data\_count == Password\_Lenght - 1) // if the array index is equal to the number of expected chars, compare data to master

{

if (!strcmp(Data, Master)) // equal to (strcmp(Data, Master) == 0)

{

lcd.clear();

ServoOpen();

lcd.print(" Door is Open");

door = 0;

}elsE{

lcd.clear();

lcd.print(" Wrong Password");

delay(1000);

door = 1;

}

clearData();

}

}