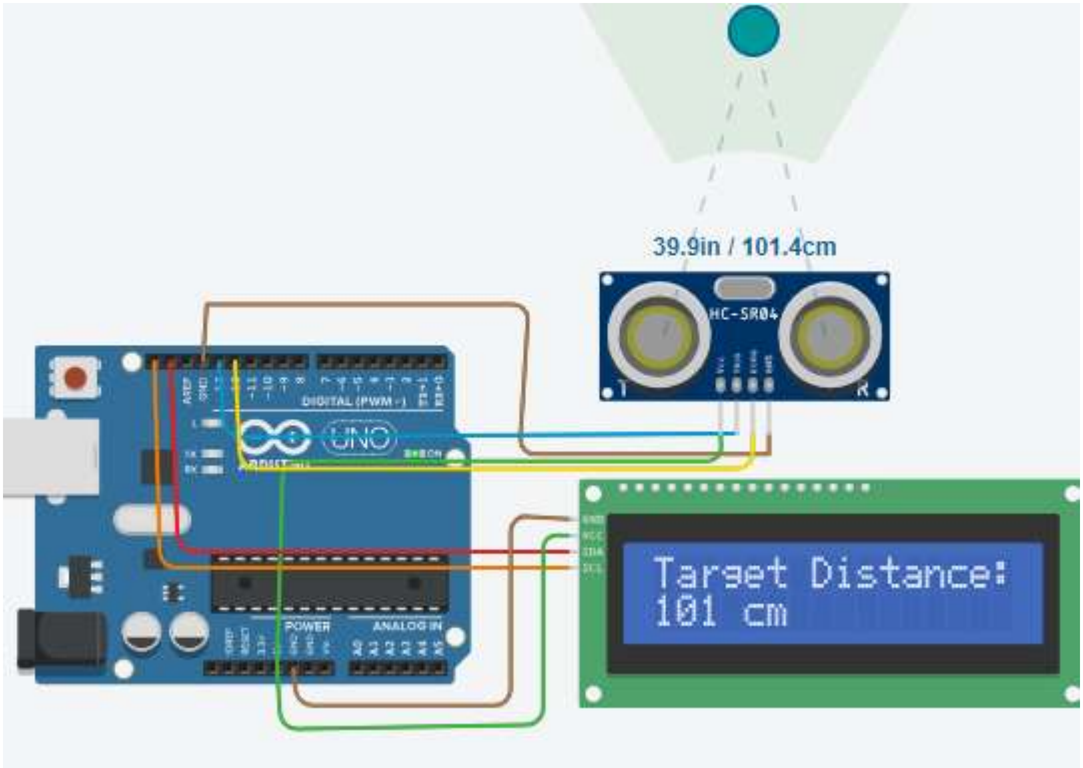
Practical 2

Study and implementation of Buzzer, Switches, LCD, keypad, LDR, Ultrasonic sensors and PWM interfacing with Arduino.

Distance Measurement using HC-SR04:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(32,16,2);  // set the lcd address to 32 for a 16 chars and 2 line display
#define trigPin 13 //Sensor Echo pin connected to Arduino pin 13
#define echoPin 12 //Sensor Trip pin connected to Arduino pin 12
void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  lcd.init();  // initialize the lcd
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Target Distance:"); // Print a message to the lcd.
  Serial.begin(9600); // The baudrate of Serial monitor is set in 9600
  while (!Serial); // Waiting for Serial Monitor
  Serial.println("Target Distance:");
}
```

```
void loop()
{
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(10);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;//distance=duration/58
  lcd.setCursor(0,1); //Set cursor to first column of second row
  lcd.print(""); //Print blanks to clear the row
  lcd.setCursor(0,1); //Set Cursor again to first column of second row
  lcd.print(distance); //Print measured distance
  lcd.print(" cm"); //Print your units.
  Serial.print(distance); //Print measured distance
  Serial.println("cm "); //Print your units.
  delay(250); //pause to let things settle
}
```

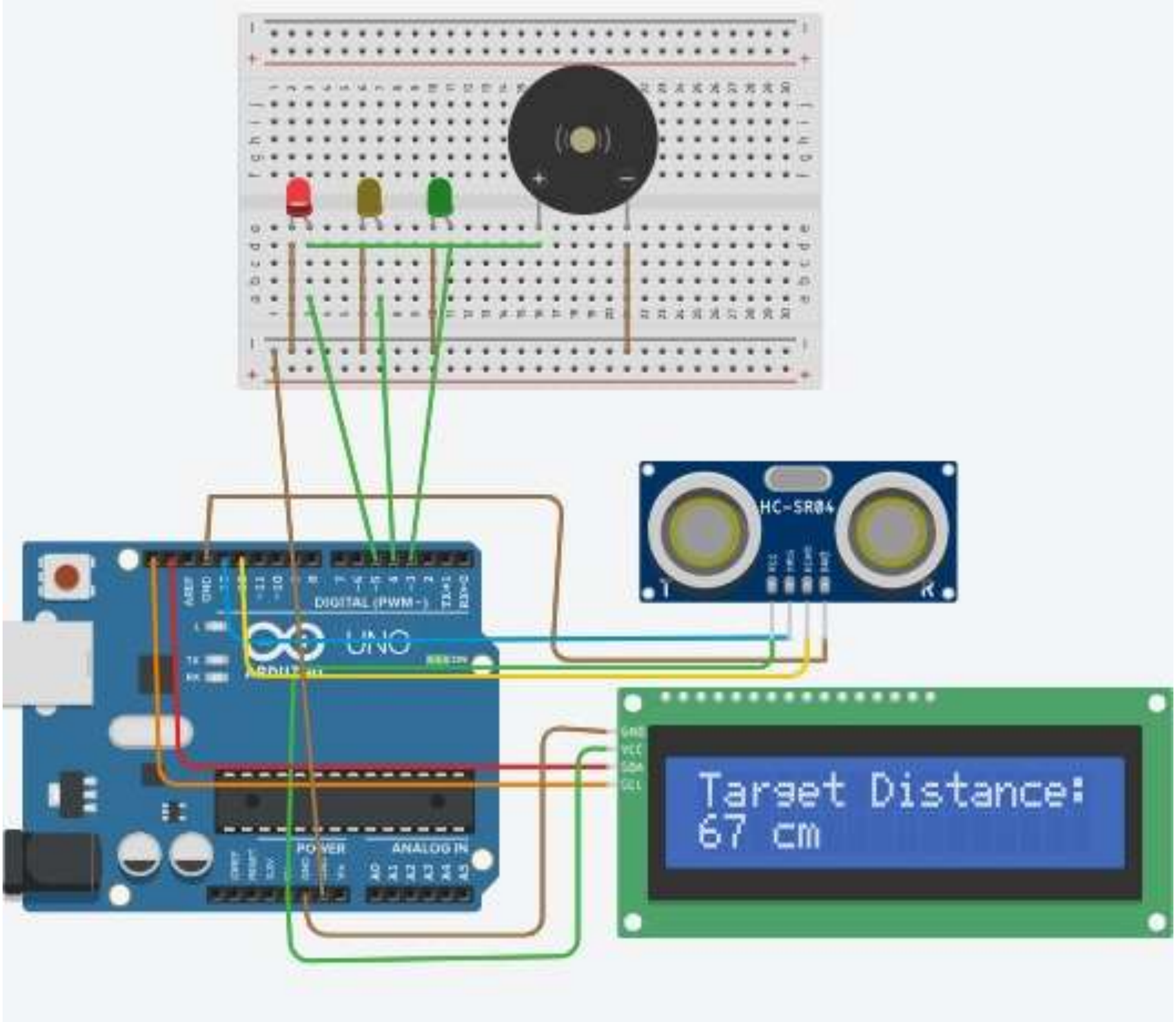


```
Target Distance:
101 cm
```

Car parking indicator system using HC-SR04:

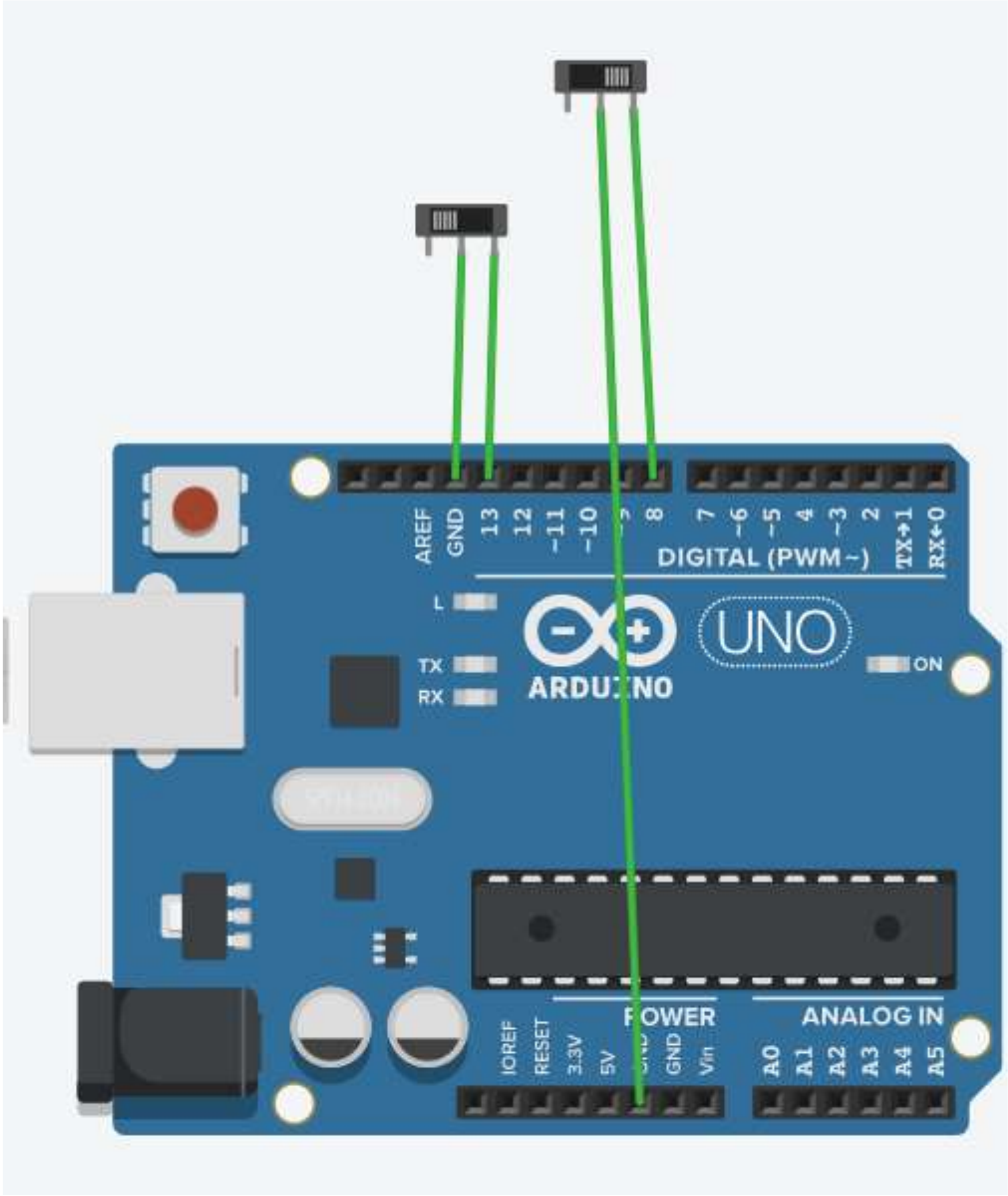
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(32,16,2);  // set the lcd address to 32 for a 16 chars and 2 line display
#define trigPin 13 //Sensor Echo pin connected to Arduino pin 13
#define echoPin 12 //Sensor Trip pin connected to Arduino pin 12
void setup()
{
            int pin[]={5,4,3};
            for(int i=0;i<3;i++)
    {int y=pin[i];
      pinMode(y, OUTPUT);}
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  lcd.init();  // initialize the lcd
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Target Distance:");  // Print a message to the lcd.
  Serial.begin(9600); // The baudrate of Serial monitor is set in 9600
  while (!Serial); // Waiting for Serial Monitor
  Serial.println("Target Distance:");
}
```

```
void loop()
{
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(10);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(10);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/58);
  if (distance<100)
  {
    tone(5, distance, 200);//start buzzer
    digitalWrite(5, HIGH);
    digitalWrite(4, LOW);
    digitalWrite(3, LOW);
  }
  else if (distance>100 && distance <200)
  {
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    digitalWrite(3, LOW);
  }
  else
  {
    digitalWrite(3, HIGH);
    digitalWrite(5, LOW);
    digitalWrite(4, LOW);
  }
  lcd.setCursor(0,1); //Set cursor to first column of second row
  lcd.print(""); //Print blanks to clear the row
  lcd.setCursor(0,1); //Set Cursor again to first column of second row
  lcd.print(distance); //Print measured distance
  lcd.print(" cm"); //Print your units.
  Serial.print(distance); //Print measured distance
  Serial.println("cm "); //Print your units.
  delay(250); //pause to let things settle
}
```



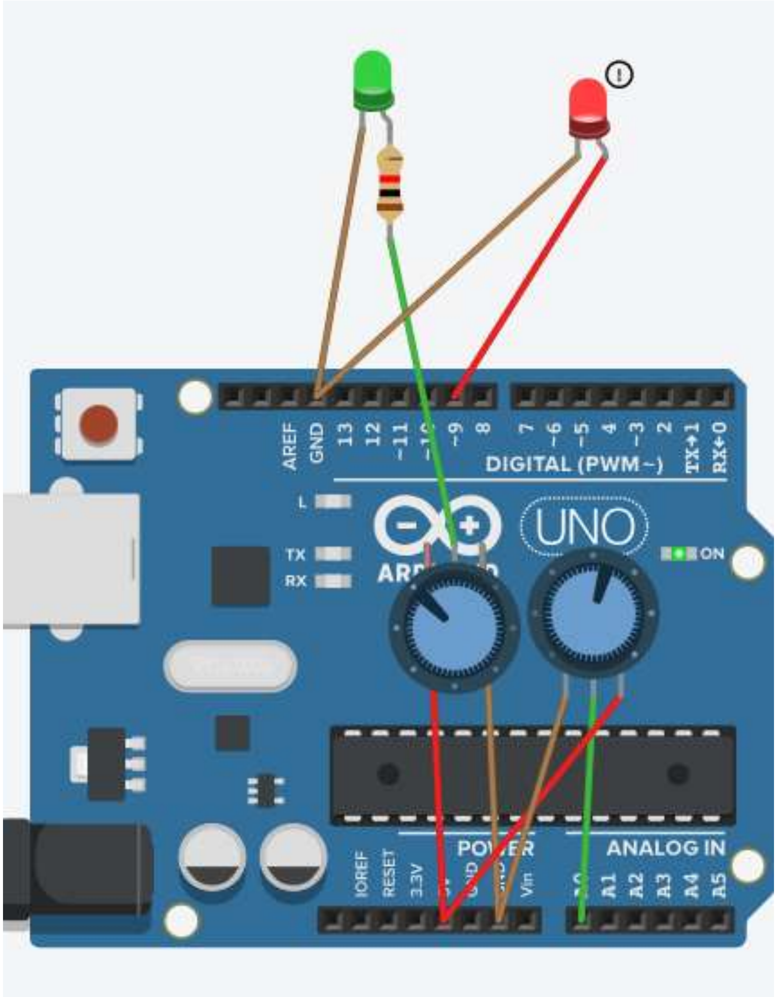
Button-Controlled Increment and Decrement Counter Using Arduino

```
#define buttonp 13
#define buttonn 8
void setup() {
  pinMode(buttonp, INPUT_PULLUP);  // Set pin as input with internal pull-up resistor
  pinMode(buttonn, INPUT_PULLUP);
  Serial.begin(9600);              // Initialize serial communication at 9600 baud
  while (!Serial);                 // Wait for the Serial Monitor to open
}
int counter = 0;  // Initialize counter variable
void loop() {
  if (digitalRead(buttonp) == LOW) {  // Check if the increment button is pressed
    if (counter == 100)               // Reset counter if it reaches 100
      counter = 0;
    else
      counter++;                      // Increment counter
    Serial.println(counter);          // Print the current counter value
    delay(1000);                      // Delay to debounce the button press
  }
  if (digitalRead(buttonn) == LOW) {   // Check if the decrement button is pressed
    if (counter == -100)              // Reset counter if it reaches -100
      counter = 0;
    else
      counter--;                      // Decrement counter
    Serial.println(counter);          // Print the current counter value
    delay(1000);                      // Delay to debounce the button press
  }
}
```

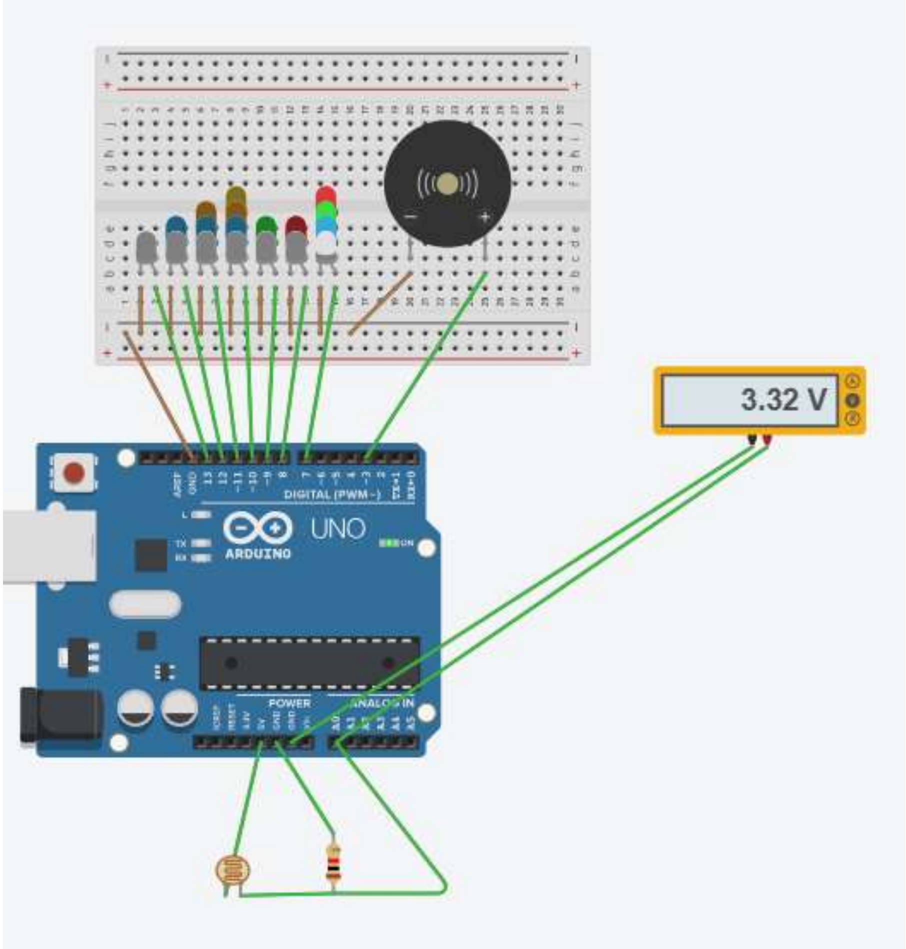

AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN

Interfacing Potentiometer with Arduino:

```
const int potPin = A0;    // Pin connected to the potentiometer
const int ledPin = 9;     // Pin connected to the LED
void setup() {
  pinMode(ledPin, OUTPUT);  // Set the LED pin as an output
}
void loop() {
  // Read the value from the potentiometer (0 to 1023)
  int potValue = analogRead(potPin);
  // Map the potentiometer value to a PWM range (0 to 255)
  int ledValue = map(potValue, 0, 1023, 0, 255);
  // Set the brightness of the LED
  analogWrite(ledPin, ledValue);
  // Small delay for stability
  delay(10);
}
```



Implementing buzzer with LDR, & Multimeter to indicate different intensity of brightness to make graph of LED and different tone of buzzer using Arduino:

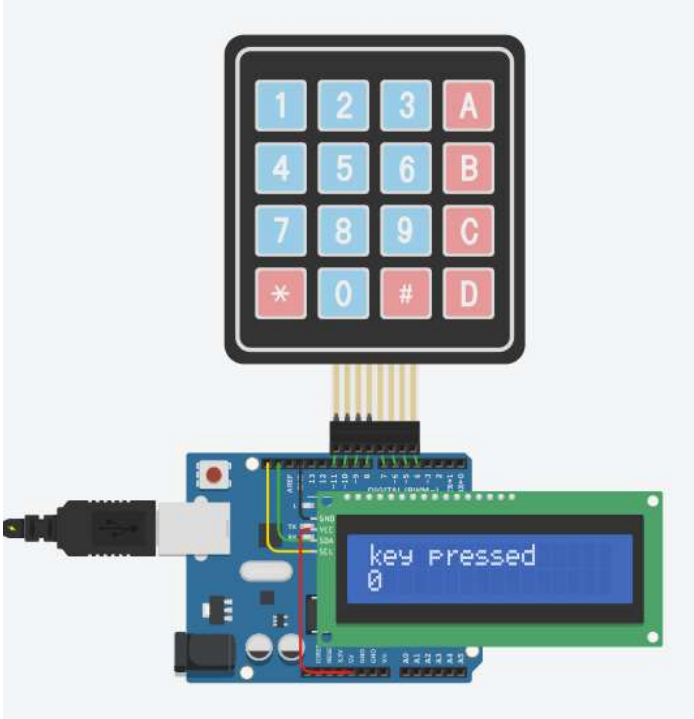


```
int pin[]={13,12,11,10,9,8,7,3};

void setup()

{

        Serial.begin(9600);

        for(int i=0;i<8;i++)

   {int y=pin[i];

     pinMode(y, OUTPUT);}

}

int x=0;

void loop()

{

  int z = analogRead(A0);

  int y = pin[(z/100)%7];

  Serial.println(z/100);

  digitalWrite(y, HIGH);

  tone(3, 500*y,500);//turn buzzer on

  delay(50);

  noTone(3);//turn buzzer off

  digitalWrite(y, LOW);

  delay(50);

  x++;

}
```

Basic Keypad implementation using Arduino:

```
#include <Adafruit_LiquidCrystal.h>
Adafruit_LiquidCrystal lcd_1(0);
#include <Keypad.h>
const byte ROWS = 4; // Four rows
const byte COLS = 4; // Four columns
char keys[ROWS][COLS] = { // Define the Keymap
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
byte rowPins[ROWS] = {11,10,9,8};
// Connect keypad COL0, COL1, COL2 and COL3 to these Arduino pins.
byte colPins[COLS] = {7,6,5,4};
// Create the Keypad
Keypad kpd = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
void setup()
{
  lcd_1.begin(16, 2);
  lcd_1.print("key pressed");
  lcd_1.setBacklight(1);
  Serial.begin(9600);
}
```

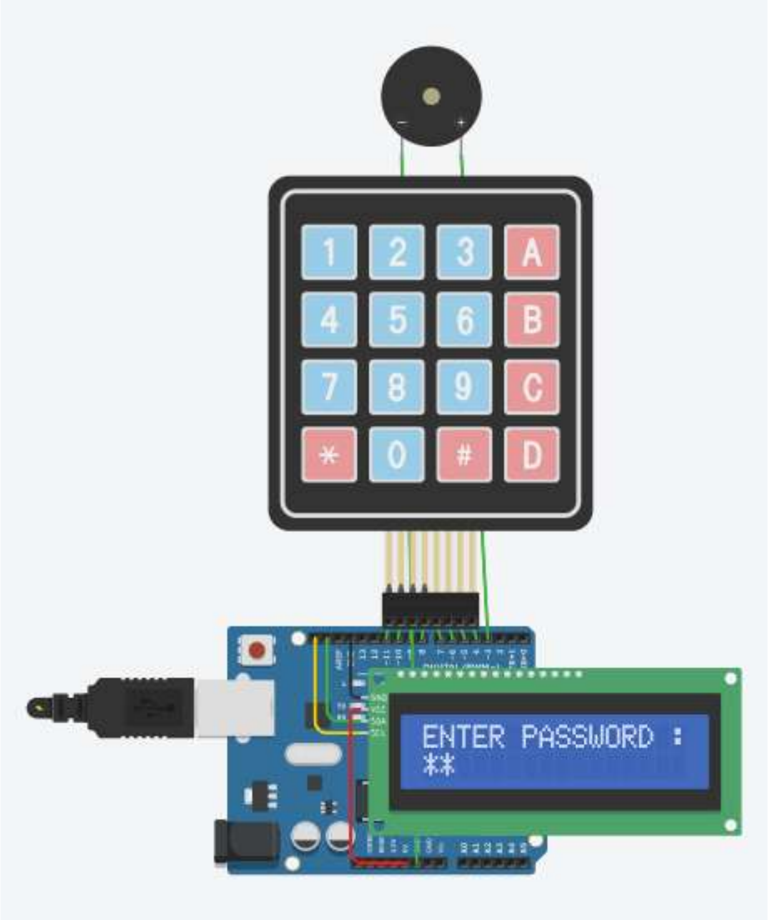
```
void loop()
{
  lcd_1.setCursor(0, 1);
  char key = kpd.getKey(); // get key pressed
  if (key) {
    Serial.println(key);
        lcd_1.print(key);
        delay(500); // Wait for 500 millisecond(s)
  }
}
```



Password matching using Keypad and Arduino uno:

```
#include <Adafruit_LiquidCrystal.h>

#include <Keypad.h>

const byte ROWS = 4; // Four rows

const byte COLS = 4; // Four columns

char keys[ROWS][COLS] = { // Define the Keymap

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

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

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

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

};

// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.

byte rowPins[ROWS] = {11,10,9,8};

// Connect keypad COL0, COL1, COL2 and COL3 to these Arduino pins.

byte colPins[COLS] = {7,6,5,4};

Keypad kpd = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

Adafruit_LiquidCrystal lcd_1(0);

void setup()

{  lcd_1.clear();

  lcd_1.begin(16, 2);

  lcd_1.print("ENTER PASSWORD :");

  lcd_1.setBacklight(1);

  Serial.begin(9600);

  pinMode(3, OUTPUT);}

char pass[] = "1111"; //password
```

```
int passLength = strlen(pass);
//char k[passLength + 1]; // +1 for null terminator
char k[5];
int i = 0;
int t=0;
void loop()
{
  lcd_1.setCursor(i, 1);
  char key = kpd.getKey();
  if (key != NO_KEY) {
    k[i] = key;
    k[i + 1] = '\0'; // null terminate the string
    lcd_1.print(key);
    i++;
    if (i == passLength) {
      if (strcmp(pass, k) == 0) {
        Serial.println("CORRECT PASSWORD");
        lcd_1.clear();
        lcd_1.print("CORRECT PASSWORD");
        delay(2000);
        lcd_1.clear();
        i=0;
        setup();
      }
else {
        // password incorrect, reset i and display error message
        i = 0;
        t++;
        lcd_1.setCursor(i, 1);
        lcd_1.print("Error!");
        delay(1000); // wait 1 second before clearing the display
        lcd_1.clear();
        lcd_1.print("ENTER PASSWORD :");
        while (t>=3)
         tone(3, 1000*t,500); // buzzer alarm
      }
    }
  }
}
```



ENTER PASSWORD :
**