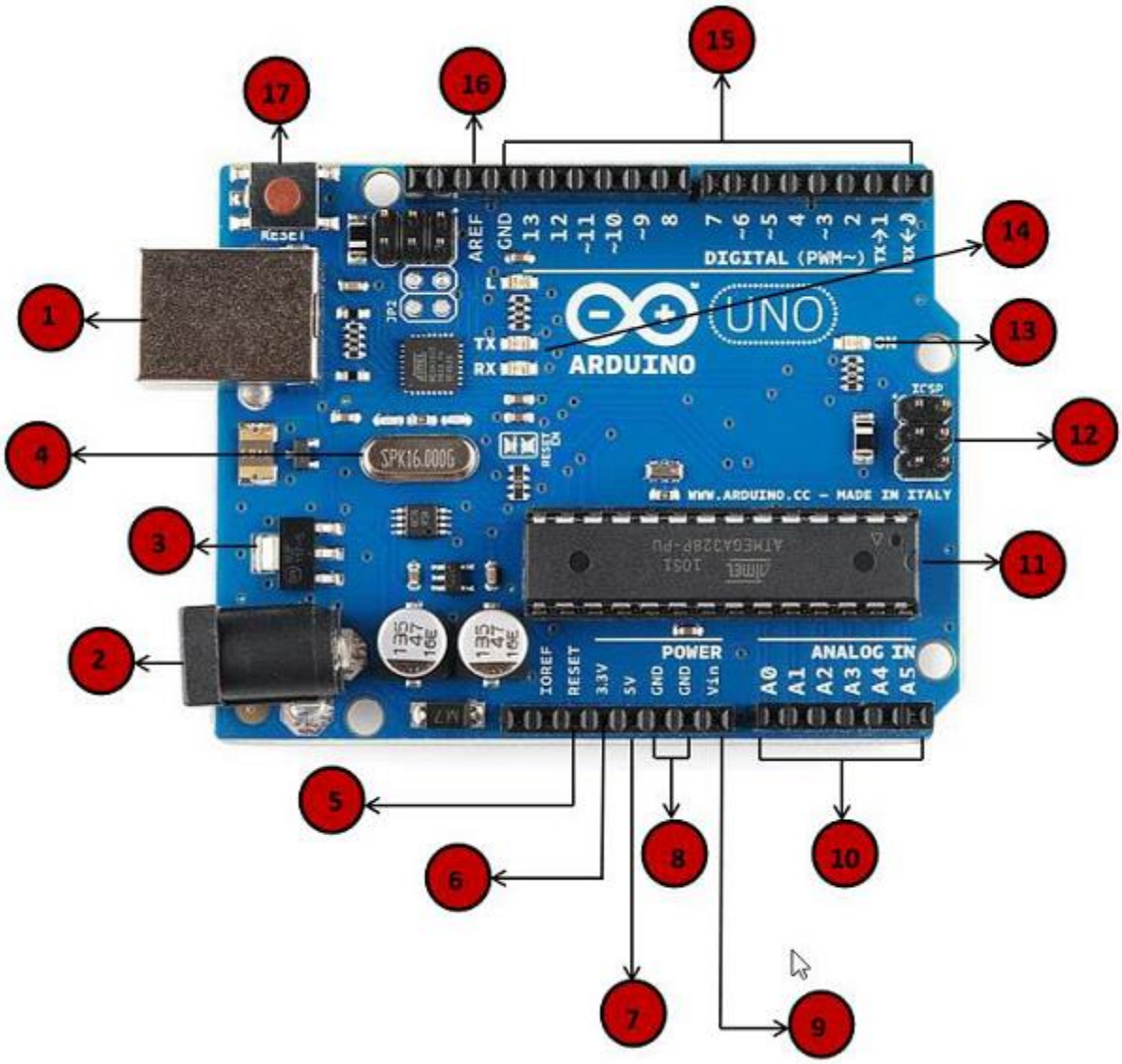
Experiment-1

Aim: Introduction to Arduino Uno Board.
Component: Arduino board.



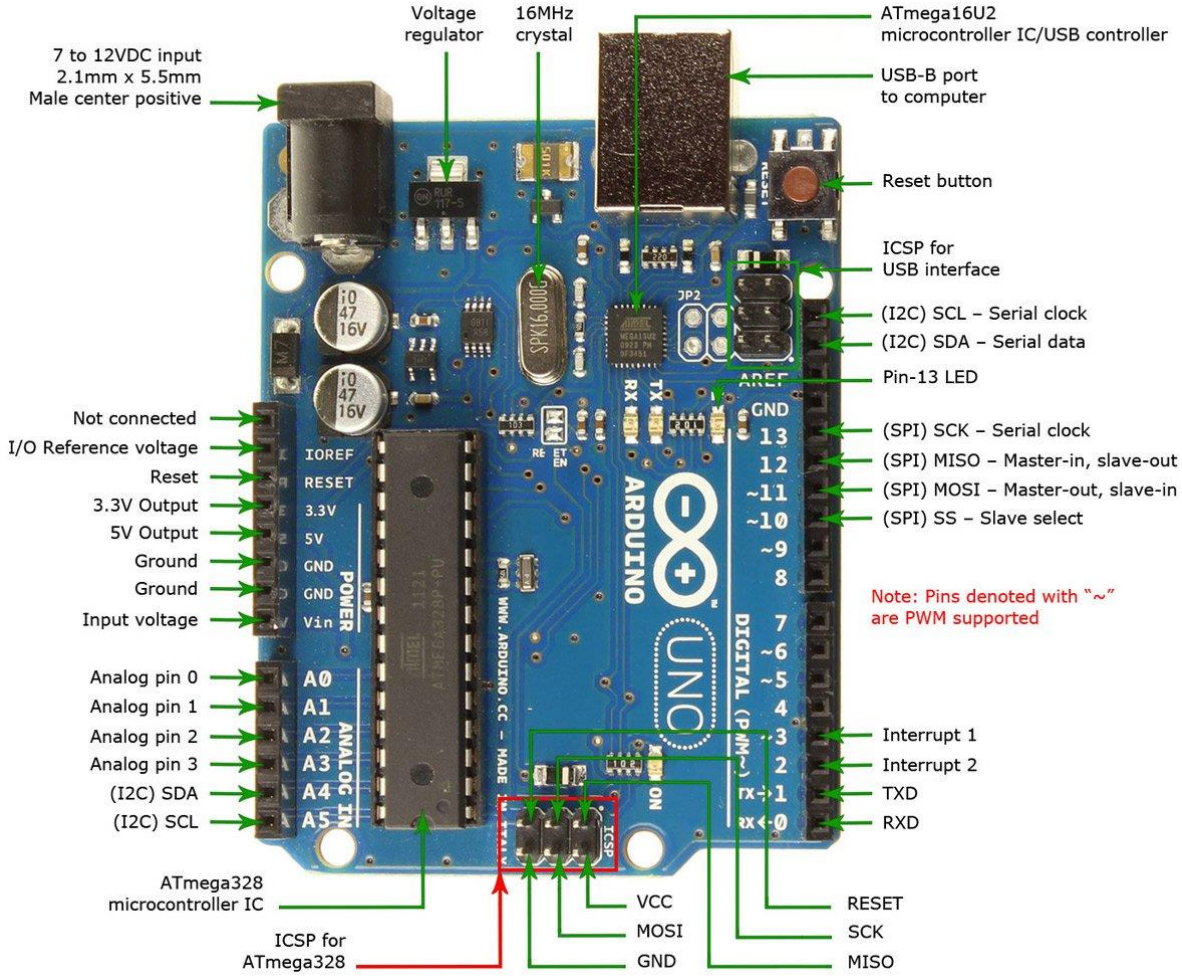
Part Descriptions:

1	**Power USB** Arduino board can be powered by using the USB cable from your computer.
2	**Power (Barrel Jack)**

	Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to **control the voltage given to the Arduino board** and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz. **Time = 1/frequency** or **frequency = 1/time** Frequency of a signal is number of cycles per second. **1K = Kilo = 10^3=1000** **1M = Mega = 10^6=1000000** **1G = Giga = 10^9=1000000000** **1T = Tera = 10^{12} = 1000000000000** **1m = mili = 10^{-3} =** **1micro = micro =10^{-6}=** **1n = neno = 10^{-9} =**
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground: 0v) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.

	• Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different

	modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.



Experiment-2 (Lab-2)

Aim: To blink the LED using arduino programming
Components: Arduino board, connecting wires, LED (Light emitting diode), bread board.
Descriptions:

- LED works as a switch. (Diode as a switch)

- Connect two ends of LED with arduino pins.
- We are having total 14 digital pins and 6 analog pins.
- Use PIN 13 to connect positive end of LED.
- Use Ground pin (0 v) to connect other end of LED.

Steps:

- Use tinkercad.com site
- Click on sign in
- Sign in with google accout
- From dashboard click on circuit
- Click on new circuit
- Search arduino from component
- Place arduino
- Search led and breadboard
- Place led on breadboard
- Connect anode of led to pin 1
- Connect cathode of led to gnd pin
- Open code window
- Write the following code.

```
//declaration part is not required here

void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
```

```
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Code:

1. Declaration
2. Setup part:
    - pinmode() is used to define whether pin used is input pin or output pin

3. Main loop part
    - digitalWrite() function is use to write on digital pin.
    - digitalWrite(13, HIGH) means send high value / signal (+5v) on pin no 13.
    - digitalWrite(13, LOW) means send low value / signal (0v) on pin no 13.
    - Delay() is used to generate delay in terms of time
    - Delay(1000) means generate delay of 1000 ms.

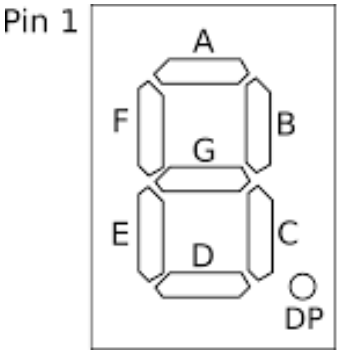
    - Run the simulation and check the output.

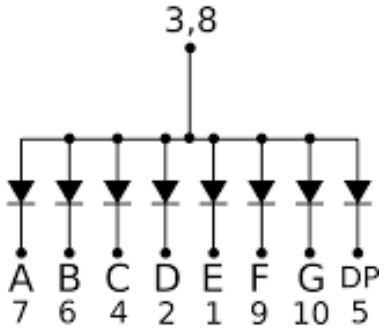
Experiment-3
7-segment LED Display

Aim: to display 0 to 9 numbers one by one on 7 segment LED display

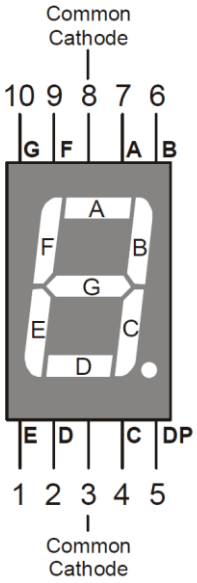
Components: Arduino board, 7 segment LED, wires, Resistor for common ground.

Theory:





Detailed diagram



Pin 7 →A
Pin 6 →B
Pin 4 →C
Pin 2 →D
Pin 1→ E
Pin 9 →F
Pin 10→G

Common cathode: then connect this pin (8) with ground
Common Anode: then connect this pin (8) with +5v

a= 11;
b=12;
c= 2;
d =3;
e = 6;
f = 9;
g= 10;

Lookup Table:

	a	b	c	d	e	f	g
0	High(1)	High	High	High	High	High	Low(0)
1	Low	High	High	Low	Low	Low	Low
2	High	High	Low	High	High	Low	High
3	High	High	High	High	Low	Low	High
4	Low	High	High	Low	Low	High	High
5	High	Low	High	High	Low	High	High
6	High	Low	High	High	High	High	High
7	High	High	High	Low	Low	Low	Low
8	High	High	High	High	High	High	High
9	High	High	High	High	Low	High	High
b	Low	Low	High	High	High	High	High
C	High	Low	Low	High	High	High	Low
d	Low	High	High	High	High	Low	High
F	High	Low	Low	Low	High	High	High

Code:

```
int a= 11;
int b=12;
int c= 2;
int d =3;
int e = 6;
int f = 9;
int g = 10;

void setup()
```

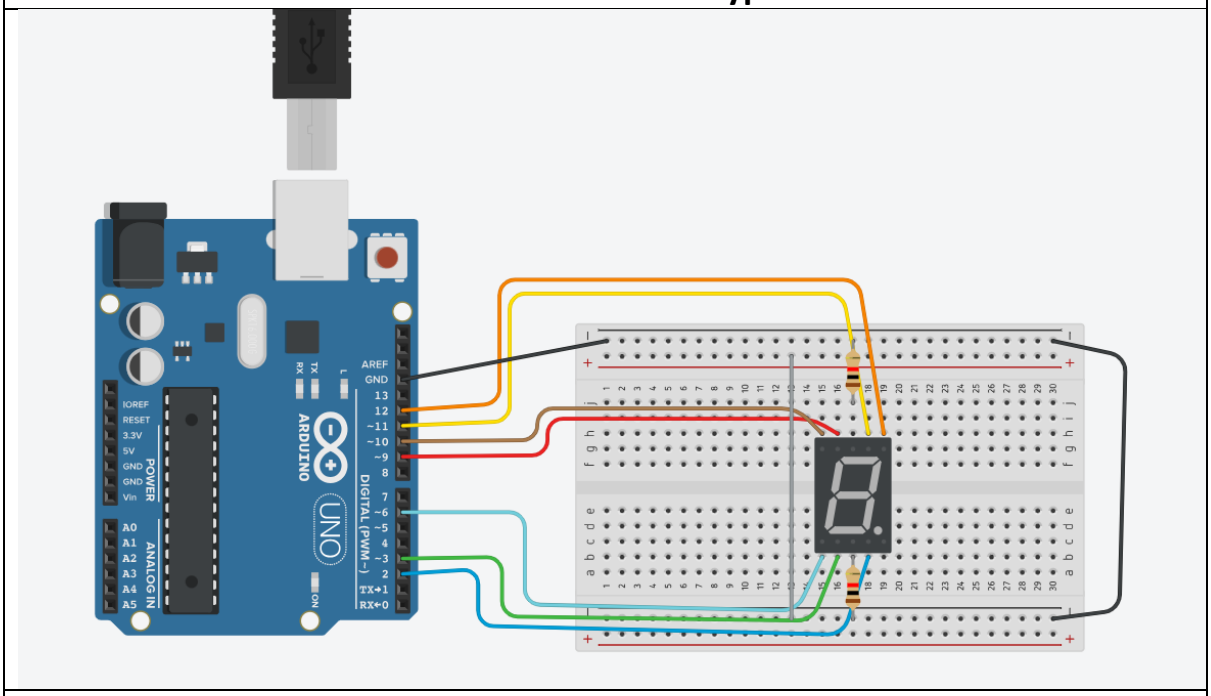
```
{
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
  }
void two()
{
 digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
   digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
   digitalWrite(e,HIGH);
  digitalWrite(f,LOW);
   digitalWrite(g,HIGH);
  delay(1000);
}
void three()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
   digitalWrite(c,HIGH);
  digitalWrite(d,HIGH);
   digitalWrite(e,LOW);
  digitalWrite(f,LOW);
   digitalWrite(g,HIGH);
  delay(1000);
}
void four()
{
  digitalWrite(a,LOW);
  digitalWrite(b,HIGH);
   digitalWrite(c,HIGH);
  digitalWrite(d,LOW);
   digitalWrite(e,LOW);
  digitalWrite(f,HIGH);
   digitalWrite(g,HIGH);
```

```
  delay(1000);
}
void five()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,LOW);
   digitalWrite(c,HIGH);
  digitalWrite(d,HIGH);
   digitalWrite(e,LOW);
  digitalWrite(f,HIGH);
   digitalWrite(g,HIGH);
  delay(1000);
}
void six()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,LOW);
   digitalWrite(c,HIGH);
  digitalWrite(d,HIGH);
   digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
   digitalWrite(g,HIGH);
  delay(1000);
}
void seven()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
   digitalWrite(c,HIGH);
  digitalWrite(d,LOW);
   digitalWrite(e,LOW);
  digitalWrite(f,LOW);
   digitalWrite(g,LOW);
  delay(1000);
}
void eight()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
   digitalWrite(c,HIGH);
```

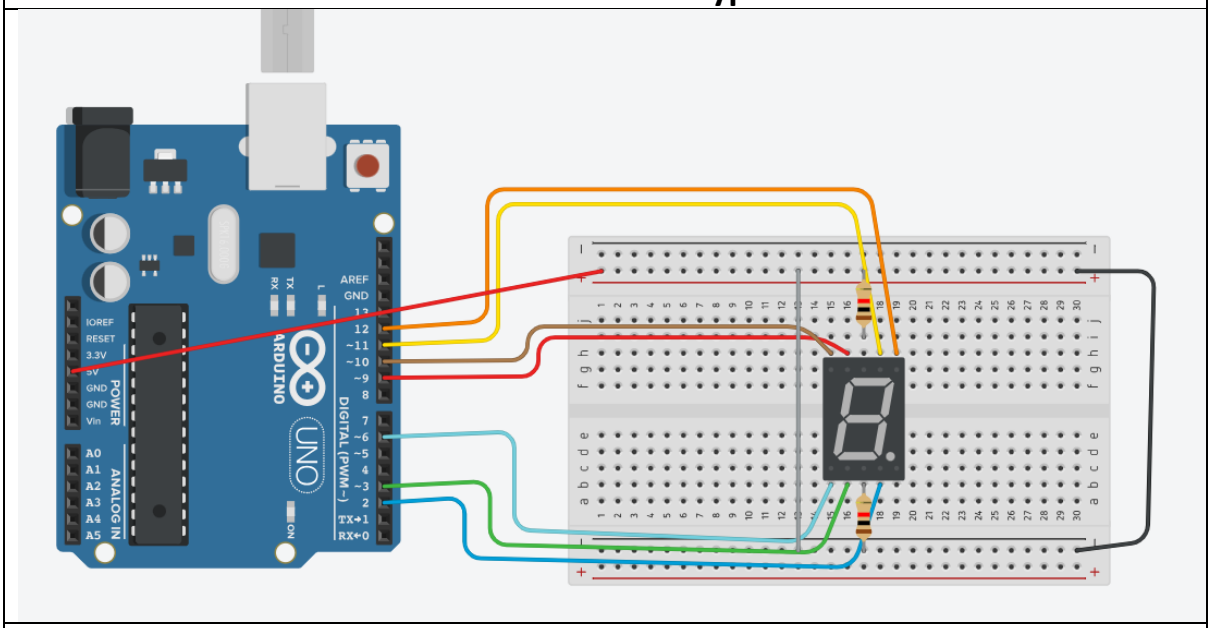
```
  digitalWrite(d,HIGH);
   digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
   digitalWrite(g,HIGH);
  delay(1000);
}
void nine()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
   digitalWrite(c,HIGH);
  digitalWrite(d,LOW);
   digitalWrite(e,LOW);
  digitalWrite(f,HIGH);
   digitalWrite(g,HIGH);
  delay(1000);
}
  void zero()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
   digitalWrite(c,HIGH);
  digitalWrite(d,HIGH);
   digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
   digitalWrite(g,LOW);
  delay(1000);
}

void loop()
{
 zero();
  // one
  digitalWrite(a,LOW);
  digitalWrite(b,HIGH);
   digitalWrite(c,HIGH);
  digitalWrite(d,LOW);
   digitalWrite(e,LOW);
  digitalWrite(f,LOW);
   digitalWrite(g,LOW);
```

```
  delay(1000);
  two();
  three();
  four();
  five();
  six();
  seven();
  eight();
  nine();
 }
void C()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,LOW);
   digitalWrite(c, LOW);
  digitalWrite(d,HIGH);
   digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
   digitalWrite(g, LOW);
  delay(1000);
}
void E()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,LOW);
   digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
   digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
   digitalWrite(g,HIGH);
  delay(1000);
}

void loop()
{
 C();
  E();
}
```

Common cathode type



Common anode type



Common anode code

```
int a= 11;
int b=12;
int c= 2;
int d =3;
int e = 6;
int f = 9;
int g = 10;


void setup()
{
  pinMode(a, OUTPUT);
```

```
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);

}
void one()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,LOW);
   digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
   digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
   digitalWrite(g,HIGH);
  delay(1000);
}
void two()
{
 digitalWrite(a,LOW);
  digitalWrite(b,LOW);
   digitalWrite(c,HIGH);
  digitalWrite(d,LOW);
   digitalWrite(e,LOW);
  digitalWrite(f,HIGH);
   digitalWrite(g,LOW);
  delay(1000);
}
void three()
{
  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
   digitalWrite(c,LOW);
  digitalWrite(d,LOW);
   digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
   digitalWrite(g,LOW);
  delay(1000);
}
void four()
{
  digitalWrite(a,HIGH);
  digitalWrite(b,LOW);
   digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
   digitalWrite(e,HIGH);
  digitalWrite(f,LOW);
   digitalWrite(g,LOW);
  delay(1000);
```

```
}
void five()
{
   digitalWrite(a,LOW);
  digitalWrite(b,HIGH);
   digitalWrite(c,LOW);
  digitalWrite(d,LOW);
   digitalWrite(e,HIGH);
  digitalWrite(f,LOW);
   digitalWrite(g,LOW);
  delay(1000);
}
void six()
{
  digitalWrite(a,LOW);
  digitalWrite(b,HIGH);
   digitalWrite(c,LOW);
  digitalWrite(d,LOW);
   digitalWrite(e,LOW);
  digitalWrite(f,LOW);
   digitalWrite(g,LOW);
  delay(1000);
}
void seven()
{
   digitalWrite(a,LOW);
  digitalWrite(b,LOW);
   digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
   digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
   digitalWrite(g,HIGH);
  delay(1000);
}
void eight()
{
  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
   digitalWrite(c,LOW);
  digitalWrite(d,LOW);
   digitalWrite(e,LOW);
  digitalWrite(f,LOW);
   digitalWrite(g,LOW);
  delay(1000);
}
void nine()
{
  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
   digitalWrite(c,LOW);
  digitalWrite(d,LOW);
```

```
   digitalWrite(e,HIGH);
  digitalWrite(f,LOW);
   digitalWrite(g,LOW);
  delay(1000);
}
  void zero()
{
   digitalWrite(a,LOW);
  digitalWrite(b,LOW);
   digitalWrite(c,LOW);
  digitalWrite(d,LOW);
   digitalWrite(e,LOW);
  digitalWrite(f,LOW);
   digitalWrite(g,HIGH);
  delay(1000);
}

void loop()
{
 zero();
  one();
  two();
  three();
  four();
  five();
  six();
  seven();
  eight();
  nine();

}
```

Experiment-4

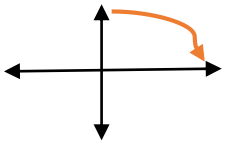
Aim: To interface potentiometer with arduino and glow LED if potentiometer value is above 300 points.

Components: Arduino board, connecting wires, LED (Light emitting diode), bread board, potentiometer

Theory:

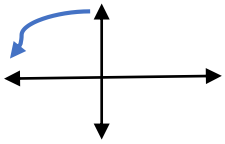
Potentiometer:

Clockwise:



- If I will rotate the button in direction of clock from north to east then it is clockwise direction.
-

Anticlockwise:



- If I will rotate the button in opposite direction of clock from north to West then it is anti-clockwise direction.
- Potentiometer is a device which is used to get variable resistor value.
- It is available with specific range.

Steps:

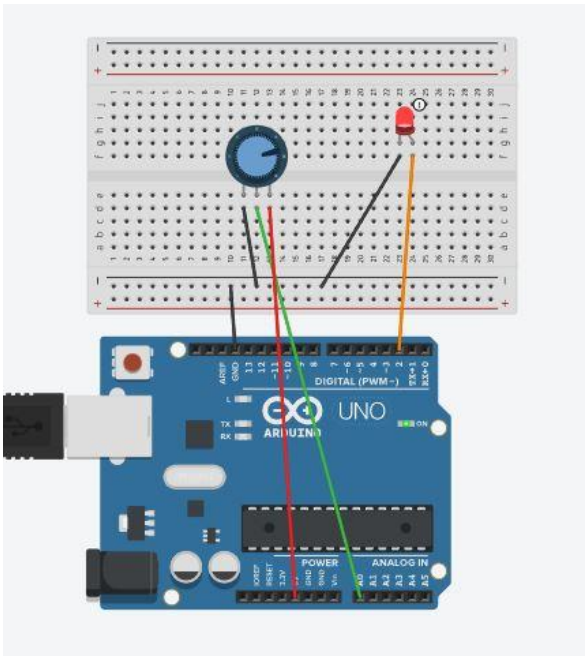
- Use tinkercad.com site
- Click on sign in
- Sign in with google account
- From dashboard click on circuit
- Click on new circuit

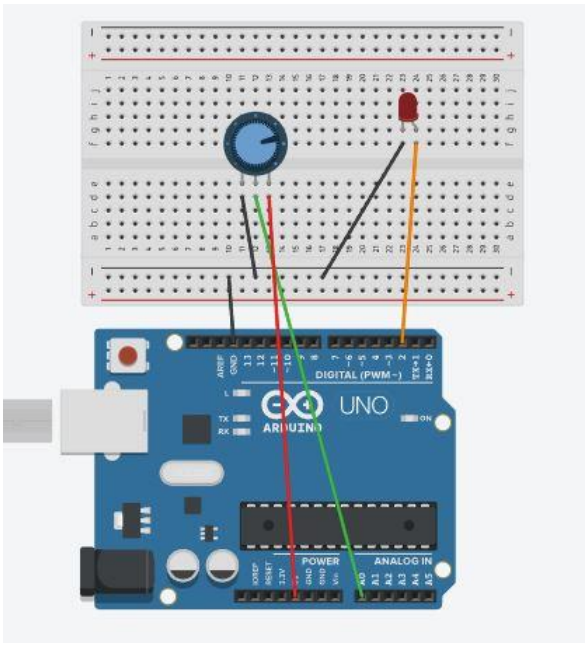
- Search arduino from component
- Place arduino
- Search led, breadboard, potentiometer.
- Place potentiometer on bread board.
- Place LED on bread board.
- Connect one end of LED to Digital pin 2 through resistor of 1kohm.
- Connect other end of LED to ground.
- Open code window
- Write the following code.

Code:

```
const int sensorMin = 0;
const int sensorMax = 800;

void setup()
{
  pinMode(2,OUTPUT);
}

void loop()
{
  // read the sensor:
  int sensorReading = analogRead(A0);

  if (sensorReading>=300)
  {
  digitalWrite(2, HIGH);
  delay(500);
  }
  else
  {
  digitalWrite(2, LOW);
  delay(500);
  }

}
```

Screenshots:





Experiment-5

Aim: To interface Gas senor (Smoke sensor) with arduino and to glow the LED if smoke value is higher.

Components: Arduino board, connecting wires, LED (Light emitting diode), Buzzer, bread board, resistor, Gas sensor.

Descriptions:

(Smoke sensor)

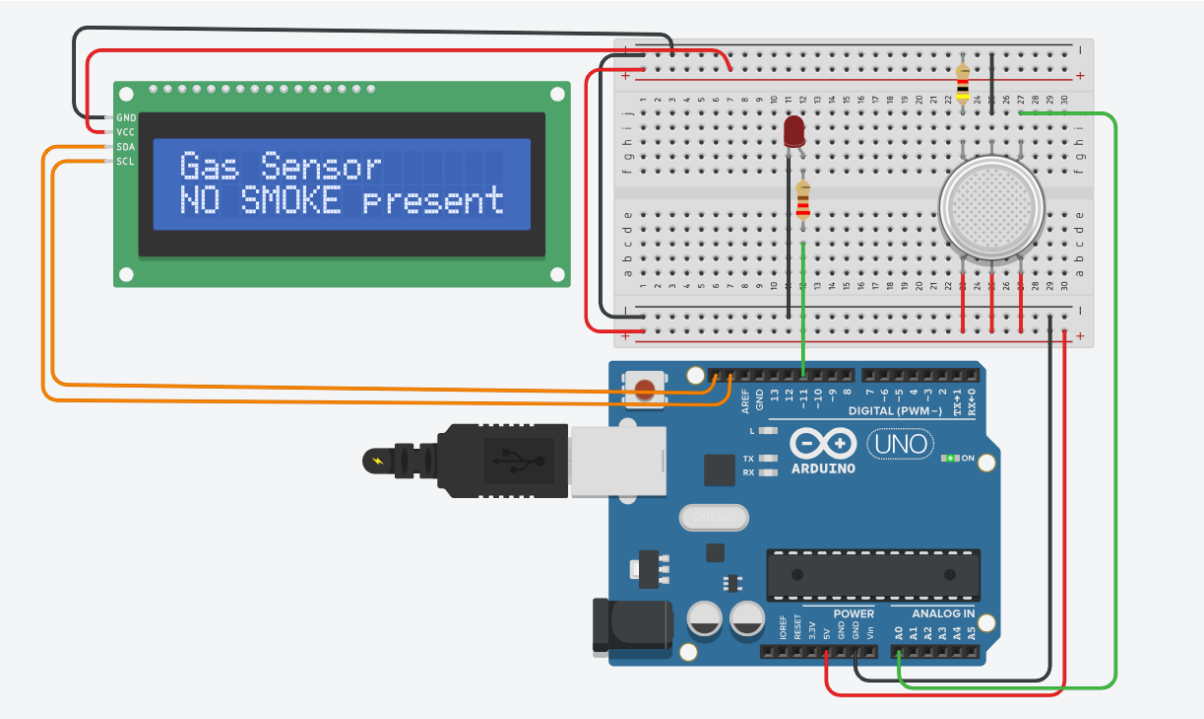
Theory:

- Smoke sensors are useful in designing fire safety applications.
- Fire alarm is the main application of smoke sensor.
- It is used in industry to maintain the fire levels.
- It is also used in application like car safety system.
- It is used for safety purpose near petrol pump or any other gas station.
- It can be used in multipurpose applications like safety at home, hospital or any other remote places.
- Any sensor must have at least 3 pins.[VCC(+5v), Ground (0v), Data line]
- Gas sensor also has supply, ground and data pins.
- Gas sensor input is of analog type and it must be connected with analog pin of arduino board.

Steps:

- Use tinkercad.com site
- Click on sign in
- Sign in with google accout
- From dashboard click on circuit
- Click on new circuit
- Search arduino from component

- Place arduino
- Search led , breadboard, resistor and gas sensor.
- Place gas sensor on bread board.
- Connect VCC of the gas sensor to +5v or vcc of arduino.
- Connect ground of gas sensor to Gnd of arduino.
- Connect data pin of gas sensor to A0 pin of arduino.
- Place LED on bread board.
- Connect one end of LED to Digital pin 2 through resistor of 1kohm.
- Connect other end of LED to ground.
- Open code window
- Write the following code.

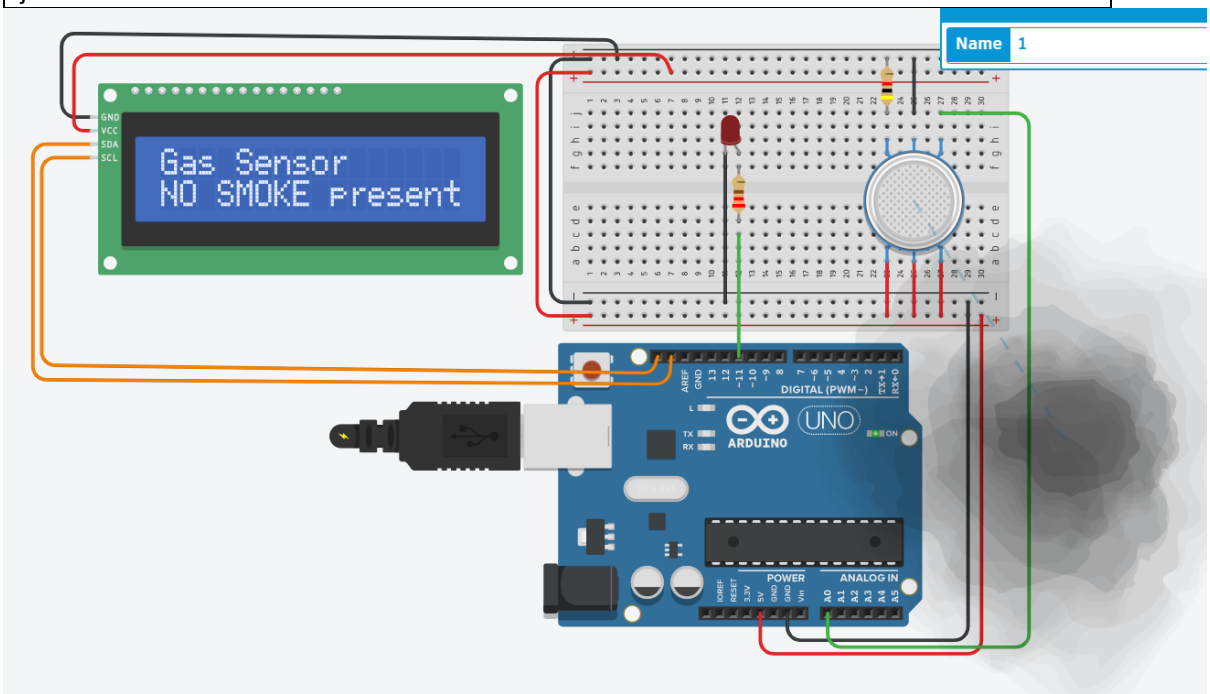


Code:

Code for interfacing Gas sensor with Arduino

```
#include <Adafruit_LiquidCrystal.h>
Adafruit_LiquidCrystal lcd(0);

int LED = 11;
int sensor = A0;

void setup()
{
  Serial.begin(9600);
  pinMode(LED,OUTPUT);
  lcd.begin(16,2);
  lcd.setCursor(0,0);
```

```
  lcd.print("Gas Sensor");
}

void loop()
{
  float sensorValue;
  sensorValue = analogRead(sensor); // read analog input pin 0

  if(sensorValue >= 470)
  {
    digitalWrite(LED,HIGH);
    Serial.print(sensorValue);
    Serial.println(" |SMOKE DETECTED");
    lcd.setCursor(0,1);
    lcd.print("SMOKE Detected  ");
  }
  else
  {
          digitalWrite(LED,LOW);
    Serial.println("Sensor Value: ");
    Serial.println(sensorValue);
    lcd.setCursor(0,1);
    lcd.print("NO SMOKE present");
  }
  delay(1000);
}
```



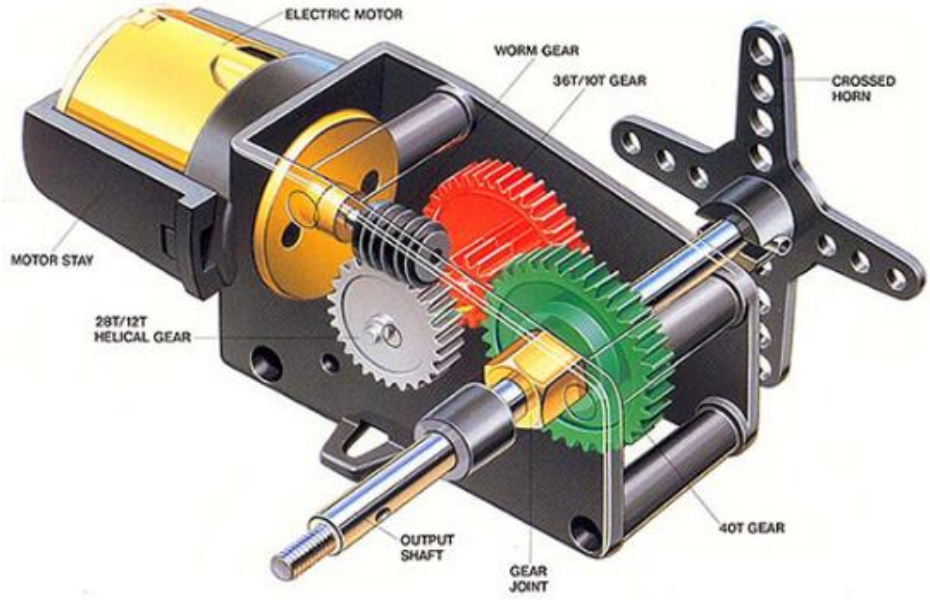
Experiment-6

Aim: To interface Servomotor with arduino controller and implement following functionality.

1. Rotate the servo motor from 0 to 180 degree in forward direction and 180 to 0 degree in reverse direction.
2. Use potentiometer (variable resistor) which controls the position of servomotor.

Components:

- Arduino board,
- connecting wires,
- Servo motor,
- bread board,
- Potentiometer



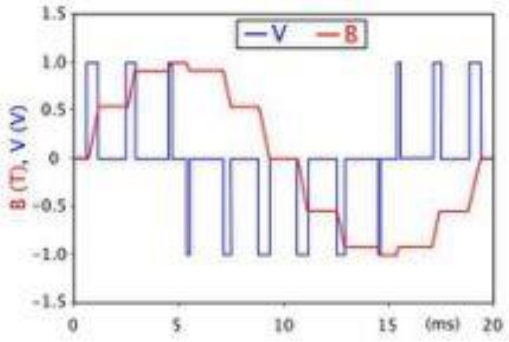
Descriptions:

- There are four main parts of servomotor
  - DC motor
  - Gearing set
  - Control circuit
  - Position sensor
- It is used in almost all robotics applications like
  - Robotics arm
  - Positioning robot
  - Helping robot
  - Robotics boat

- It is used in electronics devices like DBD player or blue ray disc player.
- They are used in automobile industries.
- It is used in electronics toys.
- It is used in aerospace industry.
- It is used in mechanical industry with CNC machine.
- It is used in elevators also.
- Servomotor works with PWM technique.

PWM: Pulse Width Modulation

- **<u>Modulation??</u>**
- [Low frequency signal (paper)+ High frequency signal(Stone)] →modulated signal (the process is modulation)
- [Modulated signal - High frequency signal(Stone)] →low frequency signal (paper) (the process is demodulation)
- In PWM width of pulse changes according to source signal.



- Servomotor is having three wires.
    - Power (Vcc, +5v)
    - Ground
    - Control
- Ground wire is connected with ground (gnd) of arduino.
- Power wire is connected with +5 v of arduino.
- Control wire is connected with Digital Pin-3 of arduino.

Connection of part-I

Ground→gnd
Power→5 v
Control→Pin3

Connection of part-II

Ground→gnd
Power→5 v
Control→Pin3
Ground of pot→gnd
Power of pot→5 v
Control of pot→A0 analog pin

Steps:

- Use tinkercad.com site
- Click on sign in
- Sign in with google accout
- From dashboard click on circuit
- Click on new circuit
- Search arduino from component
- Place arduino.
- Search servo motor, breadboard, potentiometer and add to sketch.
- Make the connection as per sketch shown in the figure.
- Write the following code.

Code:

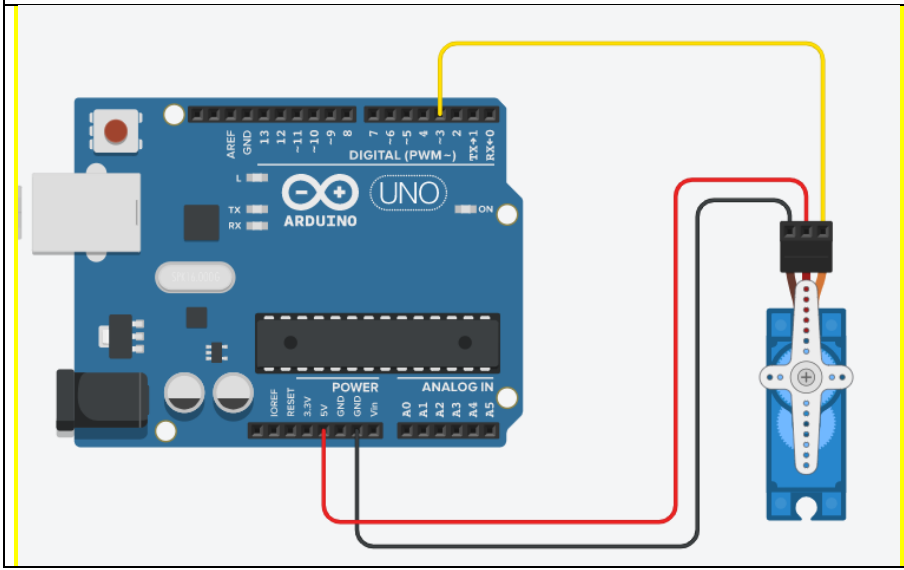
- Install library for servomotor
- Include header file for servo.
- #Include <Servo.h>

Part-I

Code for interfacing servomotor with Arduino

```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards
int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(3);  // attaches the servo on pin 3 to the servo object
}
```

```
void loop()
{
  for (pos = 0; pos <= 180; pos ++) { // 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
  }
  for (pos = 180; pos >= 0; pos --) { // 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
  }
}
```

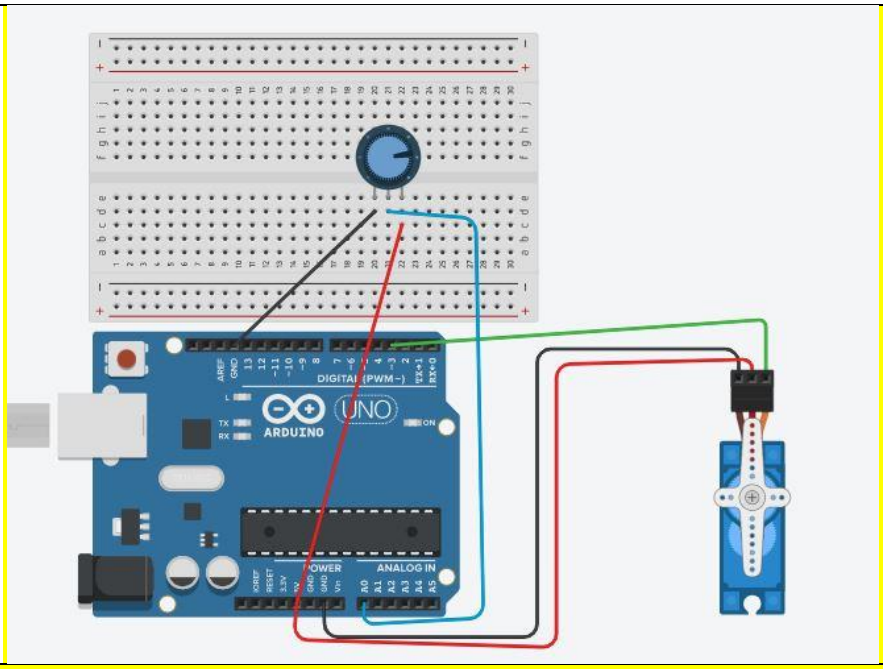


Part-II

Code (sketch) for controlling the position of servomotor using potentiometer.

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo

int val;    // variable to read the value from the analog pin

void setup() {
  myservo.attach(3);  // attaches the servo on pin 3 to the servo
}
```

```
void loop()
{
  val = analogRead(A0); // reads the value of the potentiometer
//(value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180);     // scale it to use it with the
//(value between 0 and 180)
  myservo.write(val);                  // sets the servo position

  delay(15);                           // waits for the servo to get there
}
```

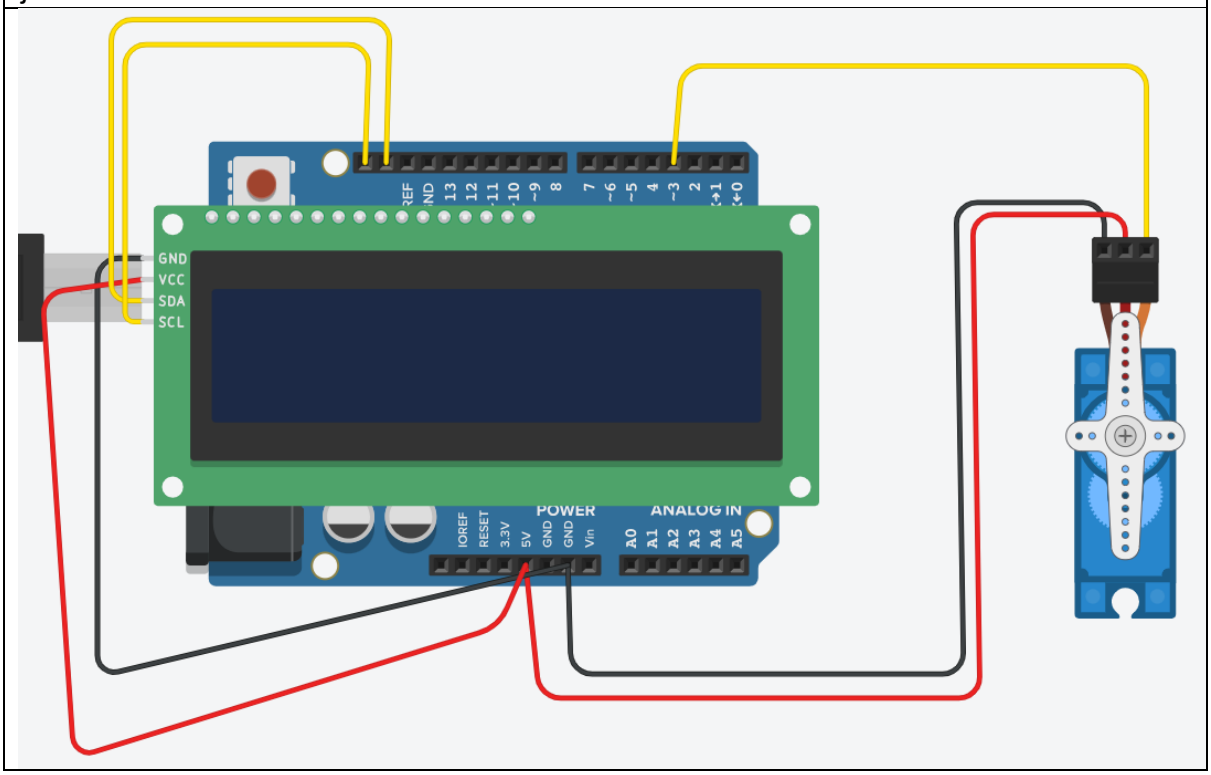


Part-III

Servomotor with I2C LCD

```
#include <Adafruit_LiquidCrystal.h>
#include <Servo.h>
Servo myservo;
Adafruit_LiquidCrystal lcd_1(0);

int pos = 0;

void setup()
{
```

```
myservo.attach(3);
lcd_1.begin(16,2);
lcd_1.print("Servo Angle");
}

void loop()
{
for (pos = 0; pos <= 180; pos ++)
{
myservo.write(pos);
lcd_1.setCursor(7,1);
lcd_1.print(pos);
delay(100);

}

for (pos = 180; pos >= 0; pos --)
{
myservo.write(pos);
lcd_1.setCursor(7,1);
lcd_1.print(pos);
delay(100);
}
}
```

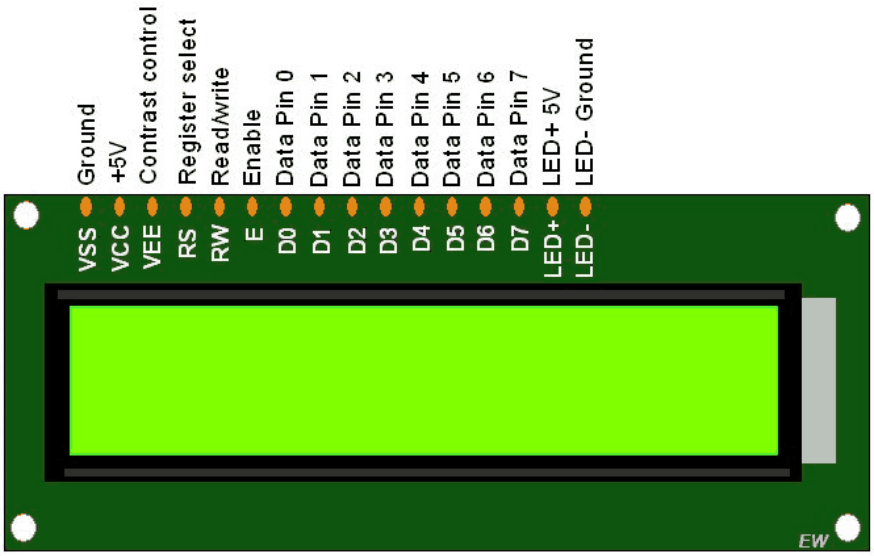


Experiment-7

Aim: Interface (16 x 2) LCD with arduino. (Without I2C)

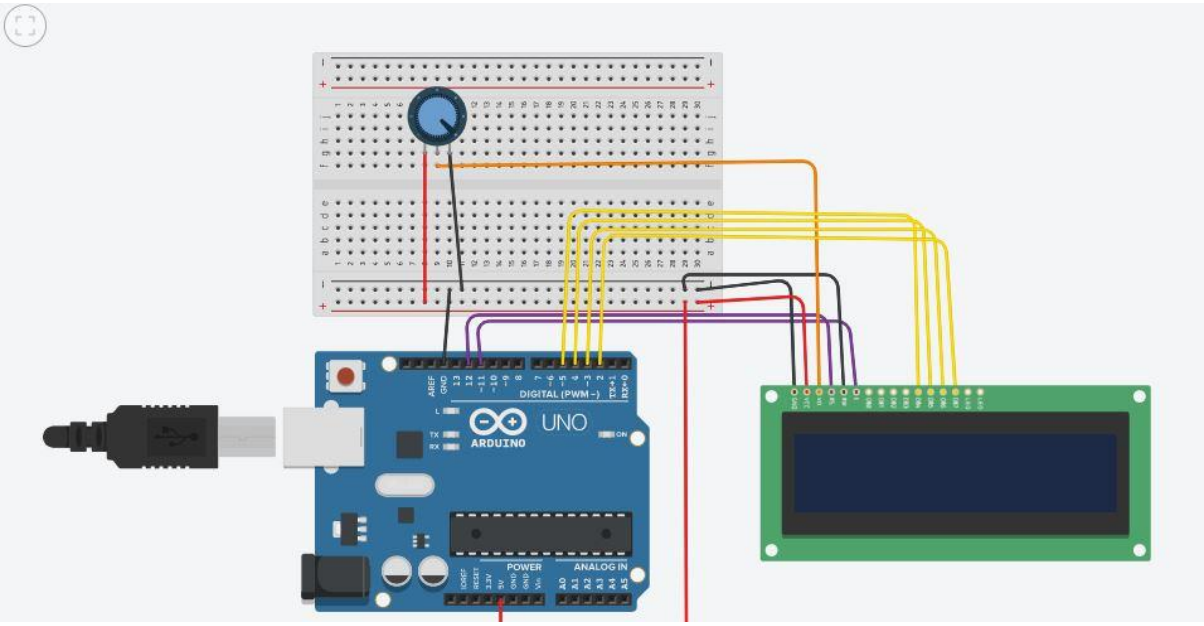
1. Print “You are the best” message on 1st line.
2. Display Time counter on 2nd line.
3. Scroll the text to left.
4. Scroll the text to right.

LCD Interfacing with Arduino:



(16 x 2) LCD

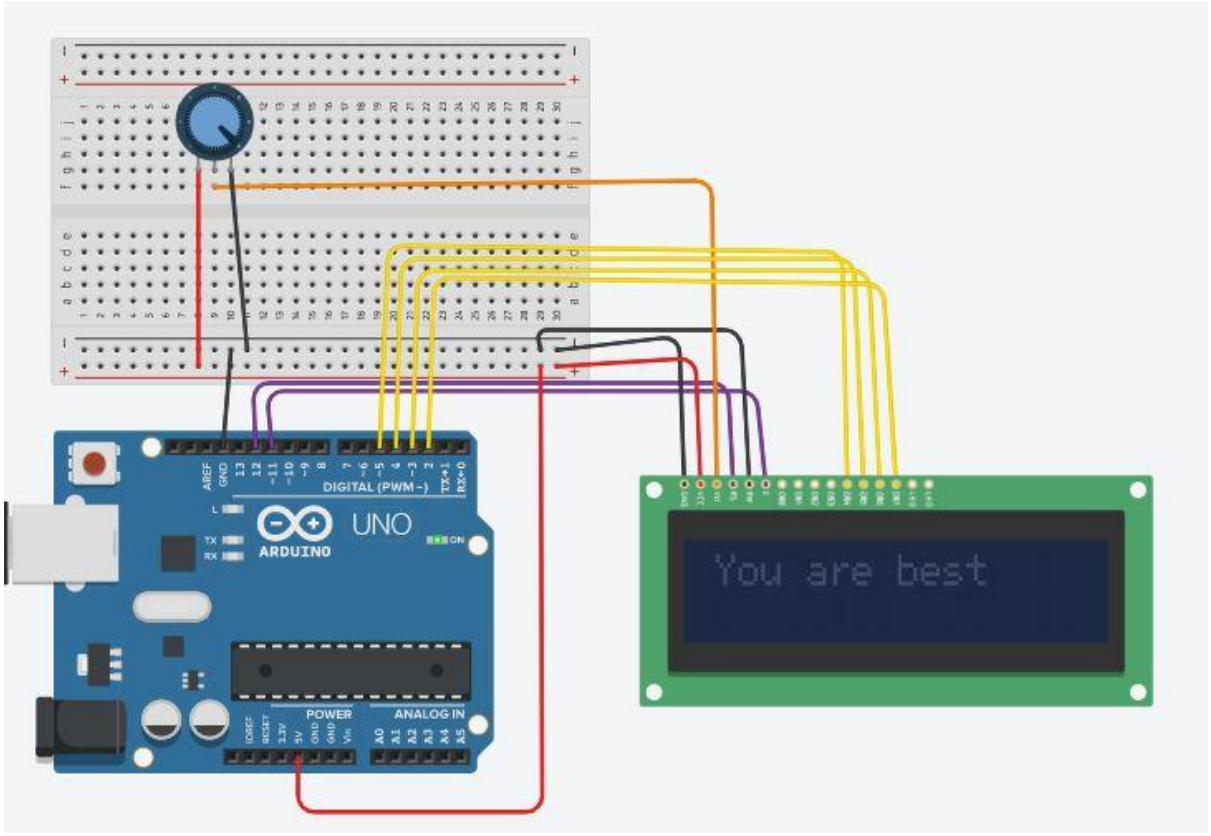
LCD Program:



LiquidCrystal()

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
  lcd.begin(16,2); // it is used to on or initialize the LCD
  lcd.print("You are best!");
}

void loop() {}
```

Output Message on LCD:



Program to Blink the cursor on LCD

```
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
void setup() {
   lcd.begin(16, 2);
   lcd.print("You are best");
}

void loop() {
  lcd.blink();
  delay(500);
  lcd.noBlink();
  delay(500);
}
```

LCD Program to print the alphabet starting with ‘a’.
1. If the alphabet is ‘m’ then change the writing direction from right to left.
2. If the alphabet is ‘s’ the change the writing direction from left to right.
3. If alphabet is ‘z’ then start again with ‘a’ from initial position

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int thisChar = 'a';
void setup()
{
  lcd.begin(16, 2);
  lcd.cursor();
}

void loop()
{
// print the character
  lcd.write(thisChar);
  delay(1000);
  thisChar++;

  // reverse directions at 'm':
  if (thisChar == 'm') {
    // go right for the next letter
```

```
    lcd.rightToLeft();
  }
  // reverse again at 's':
  if (thisChar == 's') {
    // go left for the next letter
    lcd.leftToRight();
  }
  // reset at 'z':
  if (thisChar > 'z') {
    // go to (0,0):
    lcd.home();
    // start again at 0
    thisChar = 'a';
  }
}
```

LCD Program for Autoscroling the text

```
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup()
{
    lcd.begin(16, 2);
}

void loop()
{
  // set the cursor to (0,0):
  lcd.setCursor(0, 0);
  // print from 0 to 9:
  Int i;
  for (i = 0; i < 10; i++)
 {
    lcd.print(i);
    delay(500);
  }
  // set the cursor to (16,1):
  lcd.setCursor(16, 1);
```

```
  // set the display to automatically scroll:
  lcd.autoscroll();
  // print from 0 to 9:
  for (i = 0; i < 10; i++) {
    lcd.print(i);
    delay(500);
  }
  // turn off automatic scrolling
  lcd.noAutoscroll();
  // clear screen for the next loop:
  lcd.clear();
}
```

LCD Program for Custom Characters

```
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// make some custom characters:
byte heart[8] = {
  0b00000,
  0b01010,
  0b11111,
  0b11111,
  0b11111,
  0b01110,
  0b00100,
  0b00000
};

byte smiley[8] = {
  0b00000,
  0b00000,
  0b01010,
  0b00000,
  0b00000,
  0b10001,
  0b01110,
```

```
  0b00000
};

byte frownie[8] = {
  0b00000,
  0b00000,
  0b01010,
  0b00000,
  0b00000,
  0b00000,
  0b01110,
  0b10001
};

byte armsDown[8] = {
  0b00100,
  0b01010,
  0b00100,
  0b00100,
  0b01110,
  0b10101,
  0b00100,
  0b01010
};

byte armsUp[8] = {
  0b00100,
  0b01010,
  0b00100,
  0b10101,
  0b01110,
  0b00100,
  0b00100,
  0b01010
};

void setup() {
  // initialize LCD and set up the number of columns and rows:
  lcd.begin(16, 2);
```

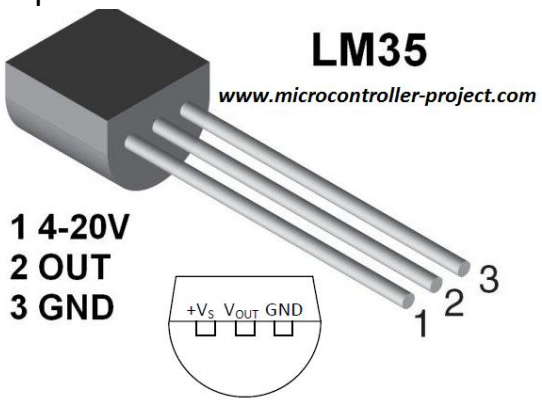
```
  // create a new character
  lcd.createChar(0, heart);
  // create a new character
  lcd.createChar(1, smiley);
  // create a new character
  lcd.createChar(2, frownie);
  // create a new character
  lcd.createChar(3, armsDown);
  // create a new character
  lcd.createChar(4, armsUp);

  // set the cursor to the top left
  lcd.setCursor(0, 0);

  // Print a message to the lcd.
  lcd.print("I ");
  lcd.write(byte(0)); // when calling lcd.write() '0' must be cast as a byte
  lcd.print(" Arduino! ");
  lcd.write((byte)1);

}

void loop() {
  // read the potentiometer on A0:
  int sensorReading = analogRead(A0);
  // map the result to 200 - 1000:
  int delayTime = map(sensorReading, 0, 1023, 200, 1000);
  // set the cursor to the bottom row, 5th position:
  lcd.setCursor(4, 1);
  // draw the little man, arms down:
  lcd.write(3);
  delay(delayTime);
  lcd.setCursor(4, 1);
  // draw him arms up:
  lcd.write(4);
  delay(delayTime);
}
```

Experiment 8
(Temperature Sensor)

AIM: Interface LM35 sensor with arduino board. Write a sketch for

1. To read the temperature
2. To Display the temperature in Celsius
3. To Display the temperature in Fahrenheit.

1. LM35 is popular temperature sensor.



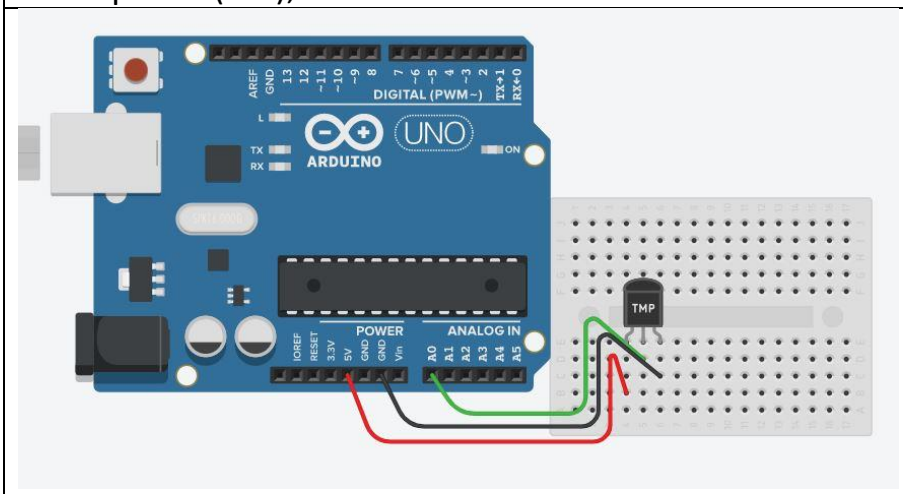
Steps (Algorithm)

1. Read the input value from temp sensor.
2. Use A0 pin to read the value.
3. Store the answer in "tmp" variable.
4. The value read from analog sensor is 10 bit number=2^{10}=
5. Convert this 10 bit number in the range of 5 v
6. Voltage = (temp)* 5v / 1024 means for temp=0 →voltage = 0 and for temp=1024→voltage = 5 v = 5000 mili volt
7. Again convert this voltage into millivolts by multiplying the voltage with 1000. Milivolt = Voltage*1000.
8. To find the temperature use following equation
   Temp (in celcius) = (millivolt – 500)/10
9. To convert temp into Fernheat (F) use following equation.
   $F = °C \times (9/5) + 32$
10. Display temp in C.
11. Display temp in F.

Program or sketch for temperature sensor

char degree = 176; //ASCI value of Degree
void setup()
{
pinMode(A0,INPUT);

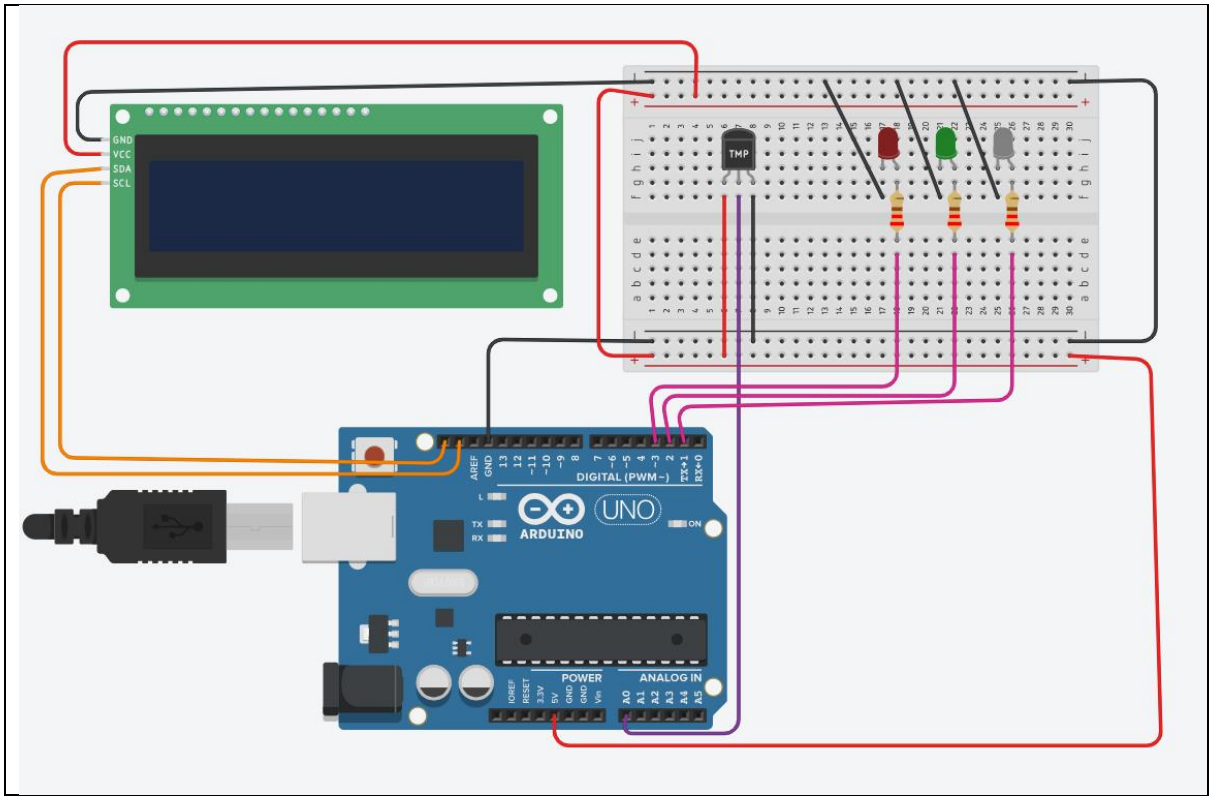
```
Serial.begin(9600);
}
void loop()
{
int tmp = analogRead(A0);
float voltage = (tmp * 5.0)/1024;
float milliVolt = voltage * 1000
float tmpCel = (milliVolt-500)/10
float tmpFer = (((tmpCel*9)/5)+32
Serial.print("10bit number(0-1023): ");
Serial.println(tmp);
Serial.print("voltage: ");
Serial.print(voltage);
Serial.println("V");
```



Temperature sensor with I2C LCD and 3 Leds

```
#include <Adafruit_LiquidCrystal.h>
Adafruit_LiquidCrystal lcd(0);
int redled=3;
int greenled=2;
int yellowled=1;

void setup()
{
pinMode(redled, OUTPUT);
pinMode(greenled, OUTPUT);
pinMode(yellowled, OUTPUT);
Serial.begin(9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
```

```
lcd.print(" Temperature ");
}

void loop()
{
 int sensordata;
 float voltage;
 float temp;
 float tempf;
 sensordata = analogRead(A0);
 voltage = sensordata *(5.0 / 1024.0);
 temp = (voltage - 0.5)*100;
 tempf = (temp*(9/5)) +32;
 Serial.println("Temperature Value is");
 Serial.print(temp);
 lcd.setCursor(3,1);
 lcd.print(temp);
 lcd.print(" C");
 Serial.println(" C");
 Serial.print(tempf);
 Serial.println(" F");
 if(temp > 40)
 {
   digitalWrite(redled, HIGH);
   digitalWrite(greenled, LOW);
   digitalWrite(yellowled, LOW);
 }
  else if (temp >10)
  {
   digitalWrite(redled, LOW);
   digitalWrite(greenled, HIGH);
   digitalWrite(yellowled, LOW);
  }
  else
  {
   digitalWrite(redled, LOW);
   digitalWrite(greenled, LOW);
   digitalWrite(yellowled, HIGH);
  }
}
```


GND
VCC
SDA
SCL
TMP
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

Experiment 9

Ultrasonic Sensor

Aim: To interface Ultrasonic sensor with arduino and make three LEDs on /off based on the distance of an object.

1. distance >350 cm then LED1=OFF, LED2=OFF, LED3=OFF
2. distance >250 cm and distance <350 cm then LED1=ON, LED2=OFF, LED3=OFF
3. distance >150 cm and distance <250 cm then LED1=ON, LED2=ON, LED3=OFF
4. distance >0 cm and distance <150 cm then LED1=ON, LED2=ON, LED3=ON

Components:

Arduino board,
Ultrasonic sensor,
Bread board,
Connecting wires,
3 LEDs,
3 resistors

Theory:

Ultrasonic sensor interfacing with arduino:

Ultra: It refer to sound which cannot be listen by human but animals like bat and dog can hear it.

Sonic: It refers to sound

- Generally human use the eye for finding the distance of an object which is not an accurate method.
- Birds like bat can emit the sound signal and receives it back and find the distance.
- Velocity = distance / time
- Time = distance / velocity
- Velocity in space or air = 3×10^8 m/s

- If we estimate the time between sending and receiving the signal then we can able to find distance.
- Ultrasonic sensor can be used in many application where we need to find the distance of an object, detecting motion, counting objects etc.
- Ultrasonic sensor must have two parts
    - A transmitter which transmits the signal which human cannot hear.
    - A Receiver which receives the reflected signal bounced back from nearby objects.
- If the object is very close to the sensor, the signal comes back quickly.
- If the object is far away from the sensor, the signal takes longer to come back
- If objects are too far away from the sensor, the signal takes so long to come back (or is very weak when it comes back) that the receiver cannot detect it.

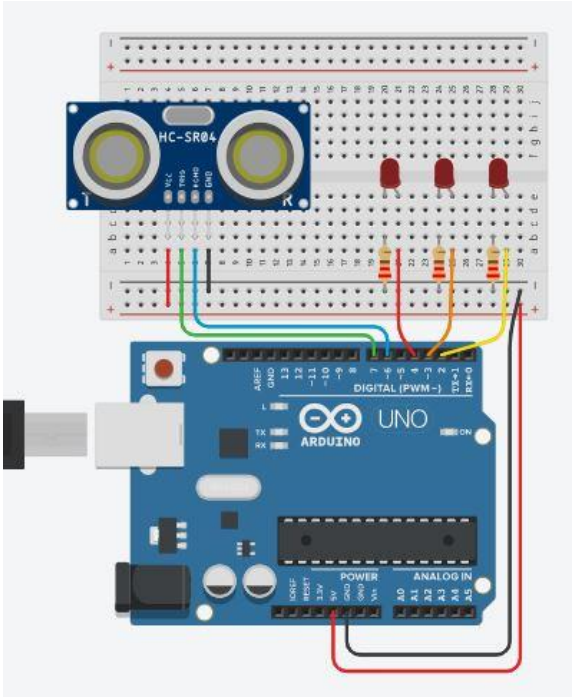
- It is used to find the objects.
- It also find the distance of the object from arduino.
- It is based on principle of transmitting and receiving ultrasonic sound waves.
- It send the waves in the space and waits for its reflection.
- It calculates the time difference between transmitted waves and received (reflected) waves.
- Calculate the distance based on tie difference.
- Distance (cm) = 0.0172* Time duration (micro second)
- Ultrasonic senor has 4 pins
    - Vcc→+5v
    - Ground
    - Trigg pin
    - Echo pin

Steps:

- Use tinkercad.com site
- Click on sign in
- Sign in with google accout
- From dashboard click on circuit
- Click on new circuit
- Search arduino from component
- Place arduino.
- Search Ultrasonic sensor , breadboard, LEDs, resistors, bread board and add to sketch.

- Make the connection as per sketch shown in the figure.
- Write the following code.

Schematic:



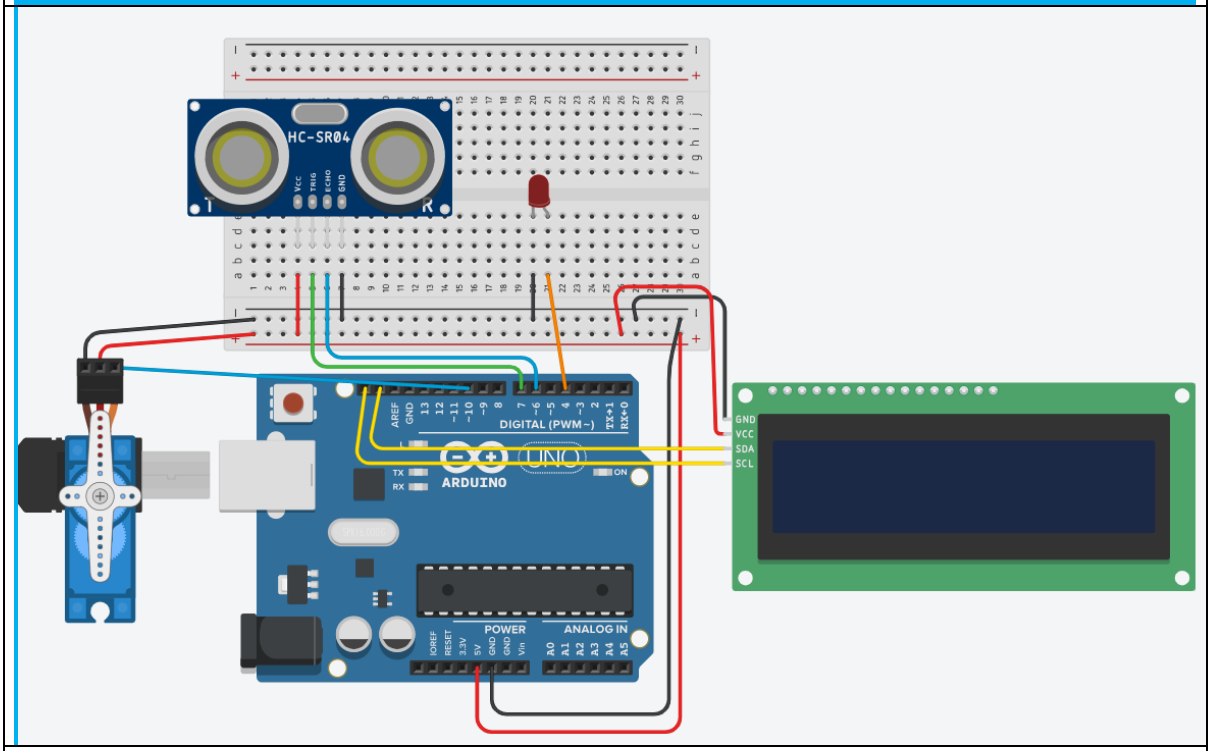
Code:

Ultrasonic sensor interfacing program
//function to find the duration
Long ultrasonicduration(int triggpin, int echopin)
{
pinMode(echopin, INPUT);
pinMode(triggpin, OUTPUT);
//Generating the pulse on triggpin
digitalWrite(triggpin, LOW);
delayMicroseconds(10);
digitalWrite(triggpin, HIGH);
delayMicroseconds(10);
digitalWrite(triggpin, LOW);
delayMicroseconds(10);
// Reads the echo pin, and returns the sound wave travel time
return pulsein(echopin, HIGH);
}
Void setup()

```
{
Serial.begin(9600);
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
}


Void loop()
{
Int cm=0.0172*ultrasonicduration(7,6);
Int inches=cm/2.54;
Serial.print(cm);
Serial.println(“ cm”);
Serial.print(inches);
Serial.println(“ inches”);
Int threshold=350;

If (cm > threshold)
{
digitalWrite(2,LOW);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
}

If(cm<=threshold && cm> threshold - 100)
{
digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
}

If(cm<=threshold-100 && cm> threshold - 200)
{
digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,LOW);
}

If(cm<=threshold-200 && cm> 0)
```

```
{
digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
}

Delay(100);
}
```

Ultrasonic sensor with I2C LCD and Servomotor




```
#include <Servo.h>
#include <Adafruit_LiquidCrystal.h>
Adafruit_LiquidCrystal lcd_1(0);
Servo myservo;

int echopin=6;
int trigpin=7;
int led=4;

void setup()
{
Serial.begin(9600);
myservo.attach(10);
pinMode(trigpin, OUTPUT);
```

```
pinMode(echopin, INPUT);
pinMode(led, OUTPUT);
lcd_1.begin(16, 2);
lcd_1.print("Distance in CM");
}

void loop()
{
int pos=0;
long duration;
long distancecm; //generating pulse on trigger pin
digitalWrite(trigpin, LOW);
delayMicroseconds(1);
digitalWrite(trigpin, HIGH);
delayMicroseconds(1);
digitalWrite(trigpin, LOW);
delayMicroseconds(1);
duration=pulseIn(echopin,HIGH);
distancecm=0.0172 *duration;
lcd_1.setCursor(7, 1);
lcd_1.print(distancecm);
lcd_1.print(" cm");
lcd_1.setBacklight(1);
Serial.print(distancecm);
Serial.println("cm");
//to glow the LED if diatnec is smaller than 10 cm
if (distancecm > 30)
{
digitalWrite(4, LOW);
for (pos = 0; pos <= 180; pos += 1)
{
  myservo.write(pos);
  delay(15);
}
for (pos = 180; pos >= 0; pos -= 1)
{
  myservo.write(pos);
  delay(15);
}
}
```

```
else
{
digitalWrite(4, HIGH);
myservo.write(0);
delay(10);
}
}
```

Experiment 10
(DC Motor)

Aim: Two DC motor interfacing with arduino using L293D motor driver.

- L293D is 16 pin IC.
- It is used to connect two DC Motors with arduino.
- Left side pins 1 to 8 are used for motor 1.
- Right side pins 9 to 16 are used for motor 2.

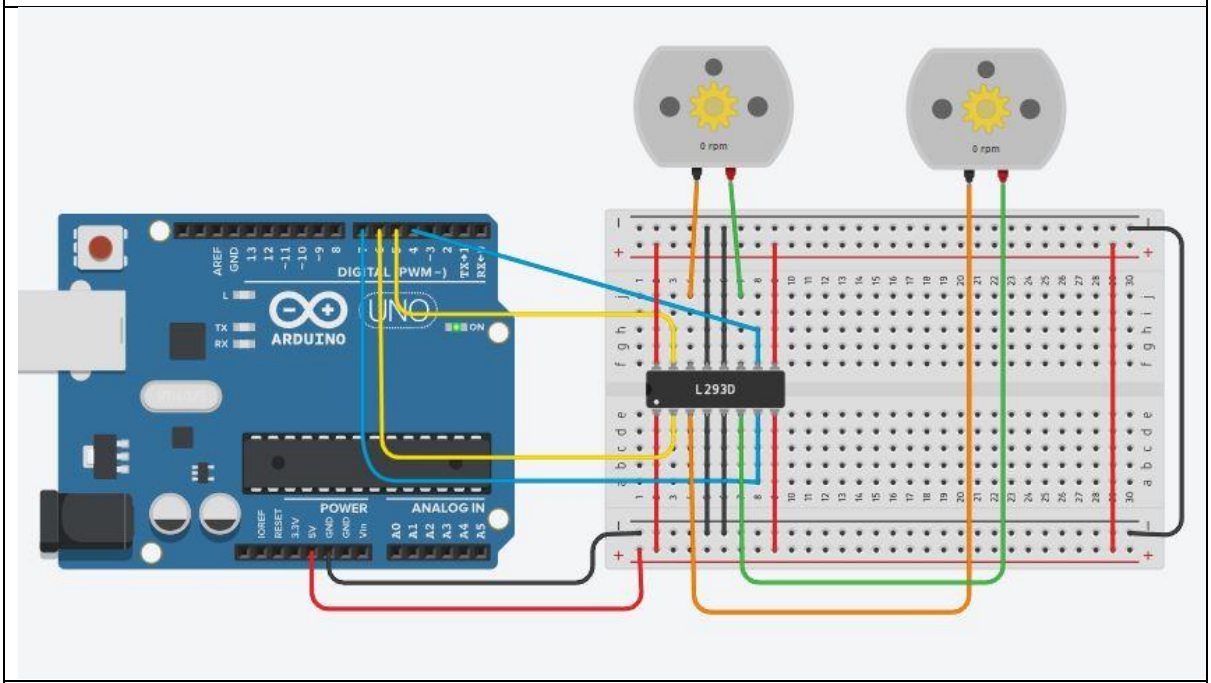
Pin Number	Pin Name	Pin Description
PIN 1	Enable 1,2	• Pin 1 is known as the enable pin. • If HIGH voltage I given to Pin 1 then it enables motor 1. • When enable pin is high then all input and output pins (2,3,6,7) are ready to work.
PIN 2	Input 1	• It is a pin which reads the input from arduino or any other controller.
PIN 3	Output 1	• It is a Pin which writes the data or output to DC motor input • It must be connected with DC Motor.
PIN 4	Ground	• The ground pin will be attached to the ground of the circuit or arduino.
PIN 5	Ground	• The ground pin will be attached to the ground of the circuit or arduino.
PIN 6	Output 2	• It is a Pin which writes the data or output to DC motor input • It must be connected with DC Motor.

PIN 7	Input 2	• It is a pin which reads the input from arduino or any other controller.
PIN 8	Vcc	• It must be connected with +5v to +16 v
PIN 9	Enable 3,4	• Pin 9 is known as the enable pin. • If HIGH voltage I given to Pin 9 then it enables motor 2. • When enable pin is high then all input and output pins (10,11,14,15) are ready to work.
PIN 10	Input 3	• It is a pin which reads the input from arduino or any other controller.
PIN 11	Output 3	• It is a Pin which writes the data or output to DC motor input • It must be connected with DC Motor.
PIN 12	Ground	• The ground pin will be attached to the ground of the circuit or arduino.
PIN 13	Ground	• The ground pin will be attached to the ground of the circuit or arduino.
PIN 14	Output 4	• It is a Pin which writes the data or output to DC motor input • It must be connected with DC Motor.
PIN 15	Input 4	• It is a pin which reads the input from arduino or any other controller.
PIN 16	Vss	• It must be connected with +5v to +16 v

Program to rotate the two motors anticlockwise and clockwise one by one.

```
void setup()
{
 pinMode(4, OUTPUT);→input from arduino pin 4 to motor 1 driver as output
 pinMode(5, OUTPUT); →input from arduino pin 5 to motor 1 driver as output
 pinMode(6, OUTPUT); →input from arduino pin 6 to motor 2 driver as output
 pinMode(7, OUTPUT); →input from arduino pin 7 to motor 2 driver as output
}

void loop()
{ //clockwise M1
 digitalWrite(4, HIGH);digitalWrite(5, LOW);
 delay(3000);
```

```
  //anticlockwise M1
  digitalWrite(4, LOW);digitalWrite(5, HIGH);
  delay(3000);
  //stop M1 for 3 second
  digitalWrite(4, LOW);digitalWrite(5, LOW);
  delay(3000);

  //clockwise M2
  digitalWrite(7, HIGH);digitalWrite(6, LOW);
  delay(3000);
  //anticlockwise M1
  digitalWrite(7, LOW);digitalWrite(6, HIGH);
  delay(3000);
  //stop M1 for 2 second
  digitalWrite(7, LOW);digitalWrite(6, LOW);
  delay(3000);
}
```



Updated Program

```
void setup()
{
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
```

```
}

void loop()
{
  //anticlockwise M2
 digitalWrite(4, LOW);digitalWrite(5, HIGH);
 digitalWrite(7, HIGH);digitalWrite(6, LOW);
 delay(10000);

 //stop M2 for 2 second
 digitalWrite(7, LOW);digitalWrite(6, LOW);
 digitalWrite(4, LOW);digitalWrite(5, LOW);
 delay(5000);

 digitalWrite(7, LOW);digitalWrite(6, HIGH);
 digitalWrite(4, HIGH);digitalWrite(5, LOW);
 delay(10000);

 //stop M2 for 2 second
 digitalWrite(7, LOW);digitalWrite(6, LOW);
 digitalWrite(4, LOW);digitalWrite(5, LOW);
 delay(5000);
}
```

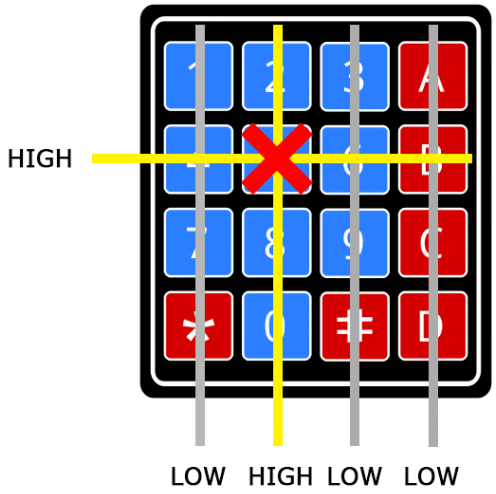
Experiment-11

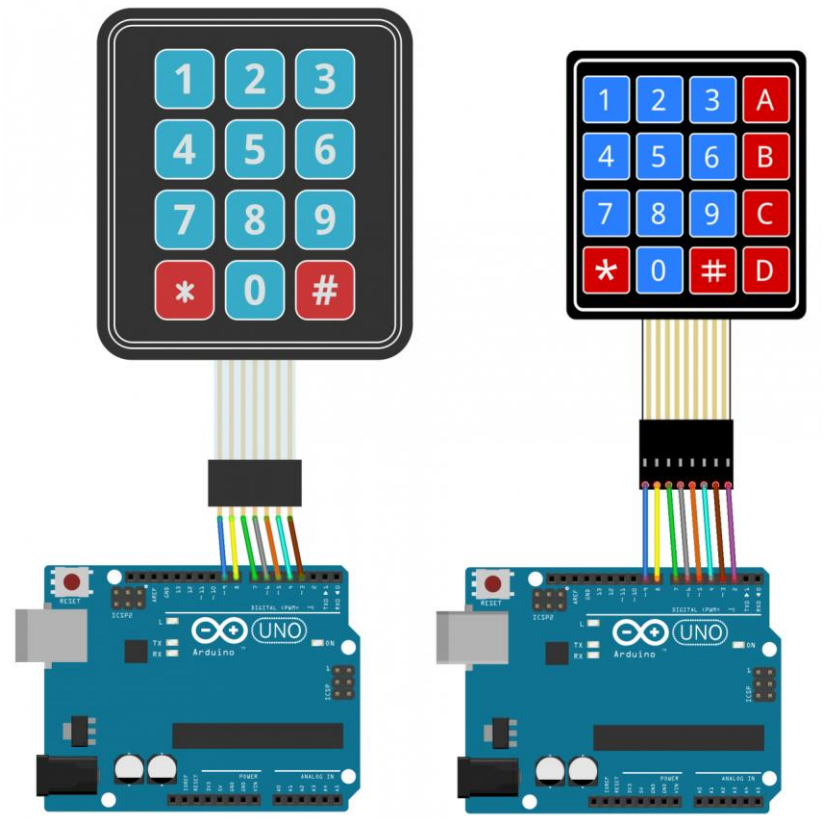
Aim: To interface keypad with arduino
Apparatus: Keypad, arduino, servomotor, LCD

Theory:

Keypads are a great way to let users interact with your project. You can use them

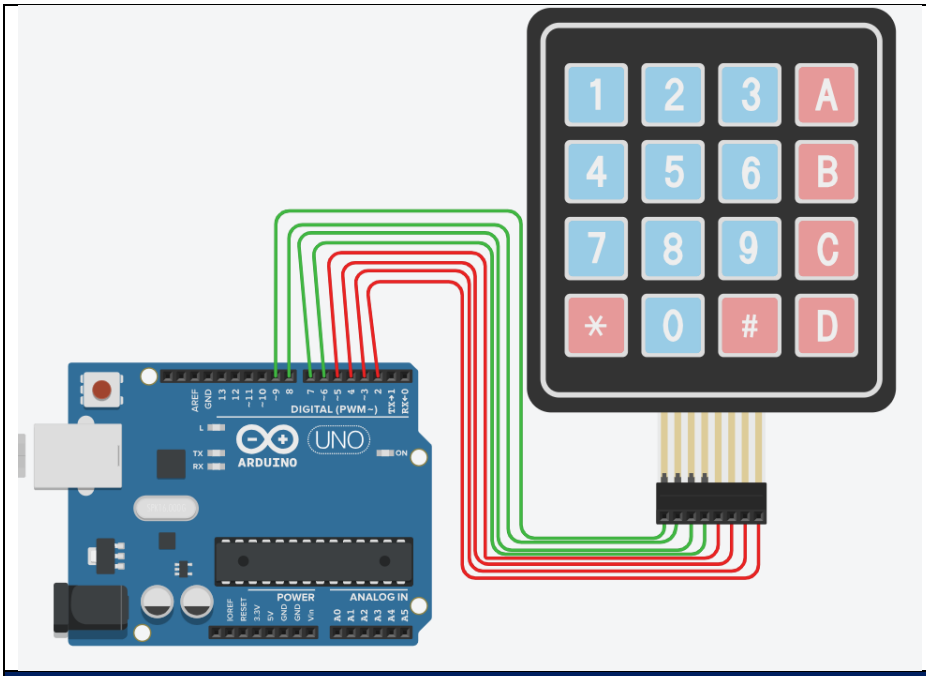
1. to navigate menus,
2. enter passwords, and
3. control games and robots.





Keypad Interfacing

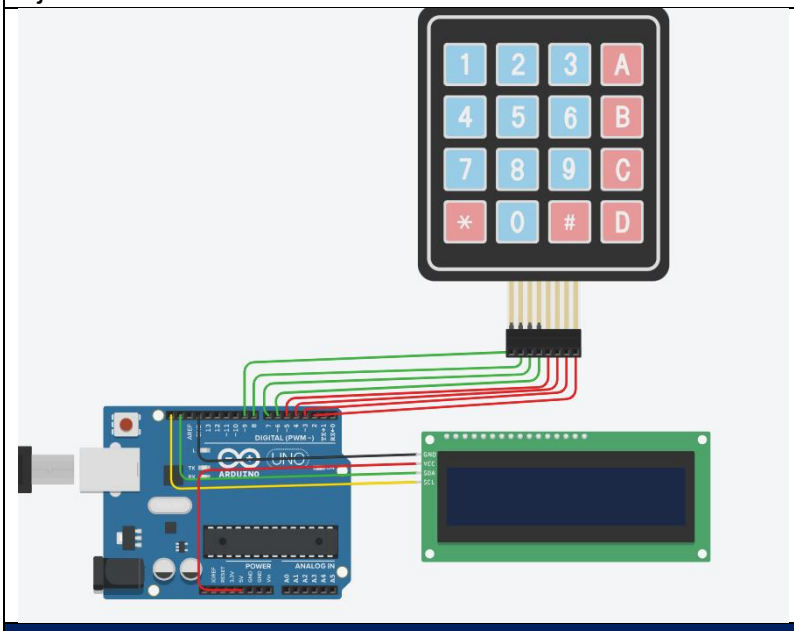
```
#include <Keypad.h>

const byte ROWS = 4;
const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins,
ROWS, COLS);

void setup(){
  Serial.begin(9600);

}

void loop(){
  char customKey = customKeypad.getKey();

  if (customKey){
    Serial.println(customKey);
  }
}
```



Keypad with I2C LCD

```
#include <Keypad.h>
#include <Adafruit_LiquidCrystal.h>

Adafruit_LiquidCrystal lcd_1(0);

const byte ROWS = 4;
const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins,
ROWS, COLS);

void setup()
{
  Serial.begin(9600);
  lcd_1.begin(16, 2);
```

```
  lcd_1.print("Enter Key Value");
}

void loop()
{
  char customKey = customKeypad.getKey();
  if (customKey){
    Serial.println(customKey);
    lcd_1.setCursor(7, 1);
    lcd_1.setBacklight(1);
          delay(500); // Wait for 500 millisecond(s)
          lcd_1.print(customKey);
    }
  }
```



Password based door locking system

```
#include <Keypad.h>
#include <Servo.h>

#define codeLength 5
Servo myservo;

char Code[codeLength];
char PassW[codeLength] = "1234";
byte keycount = 0;

const byte ROWS = 4;
const byte COLS = 4;
```

```
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {9,8,7,6};
byte colPins[COLS] = {5,4,3,2};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys),rowPins, colPins,ROWS, COLS);

void setup(){
  Serial.begin(9600);
  myservo.attach(10);
}

void loop()
{
  int pos;
  char customKey = customKeypad.getKey();
  if(customKey){
    //Serial.println(customKey);
    Code[keycount] = customKey;
    Serial.print(Code[keycount]);
    keycount++;
  }
  if(keycount==codeLength-1){
    Serial.println("");

    if(!strcmp(Code, PassW)){
      Serial.println("correct password");
      for(pos=90;pos<=180;pos++)
      {
      myservo.write(pos);
      delay(40);
      }
      delay(5000);
       for(pos=180;pos>=90;pos--)
      {
      myservo.write(pos);
      delay(40);
      }
    }
    else
      {
      Serial.println("Incorrect password");
      //myservo.write(10);
      delay(1000);
    }
    deletecount();
```

```
   }
}

void deletecount(){
  while(keycount !=0){
    Code[keycount--]=0;
   }
  return;
}
```