Potentiometer

Potentiometer is a simple knob that provides a variable resistance by changing the turning the shaft of the potentiometer we change the resistance of either side of the wiper which is connected to the center pin of the potentiometer.

Figure4-6:potentiometer

In this experiment we will connect a potentiometer to Arduino to have different analog input when we turn the shaft.

IN this experiment you'll need:
1-Arduino Uno(or any other arduino board)

2-Potentiometer(with any value)

We connect three wires to the Arduino the first goes to ground from one of outer pins of potentiometer the second goes from 5v to the other outer

pin of the potentiometer the third goes from analog input 0 to the middle of the potentiometer

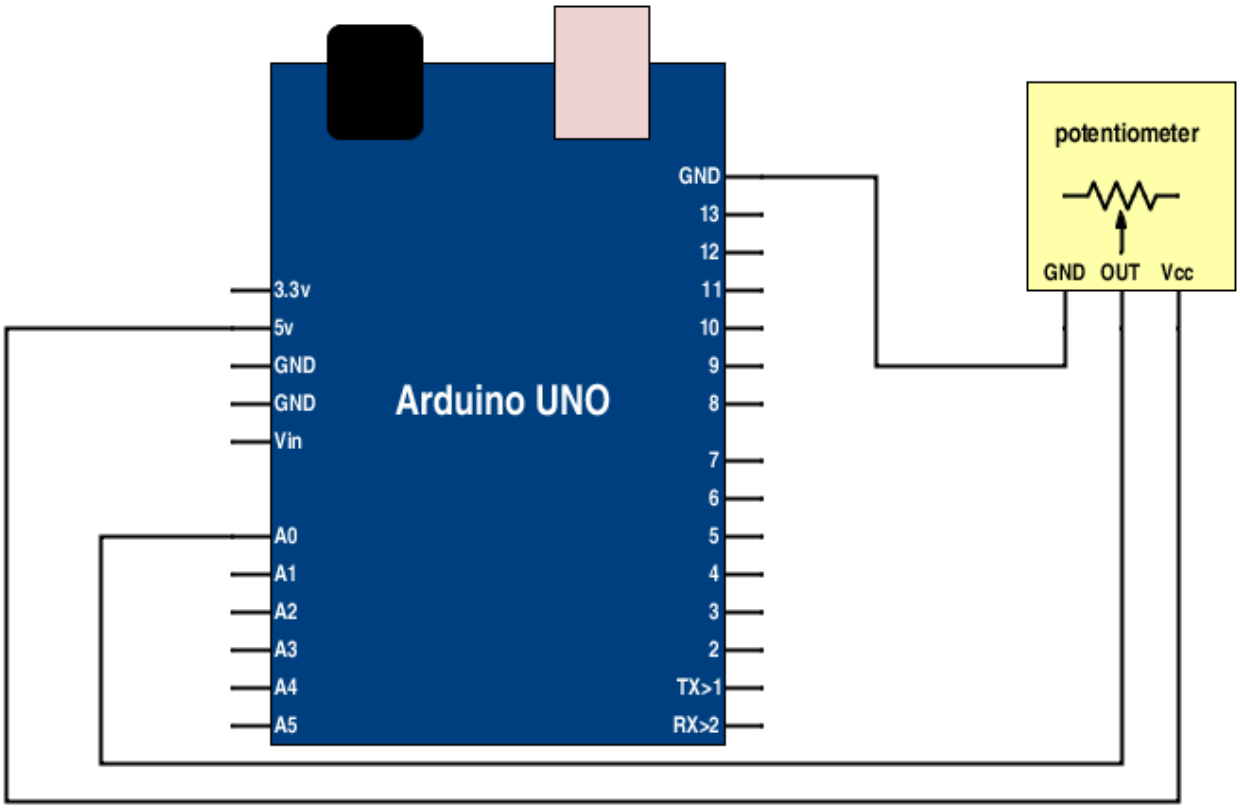


Figure 4-7:Potentiometer connection diagram

And here's the code :

```
Int potpin=A0;

Int Val=0;

Void setup(){

Serial.begin(9600);

}

Void loop(){

Val=analogRead(potpin);

Serial.println(Val);

Delay(500);

}
```

IR Sensor

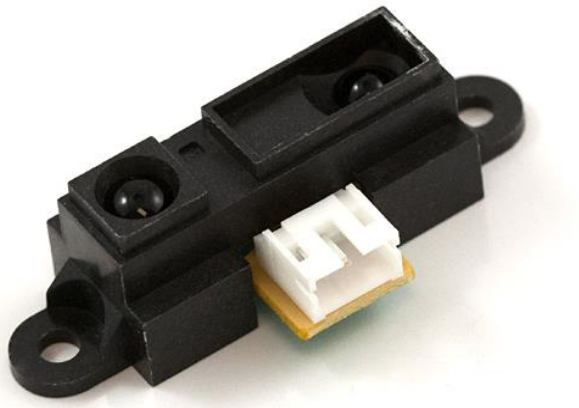


Figure 4-1: IR Sharp GP2D12

The IR sensor is used mainly for measuring distance in this experiment we will use the following components:

1-arduino Uno

2-IR Sharp GP2D12

Following are what the three different wires are:

Red (V_{CC}) – Connects to the positive terminal of the sensor

Black (GND) – Connects to the ground terminal of the sensor

Yellow (V_O) – Connects to the analogue output from the sensor

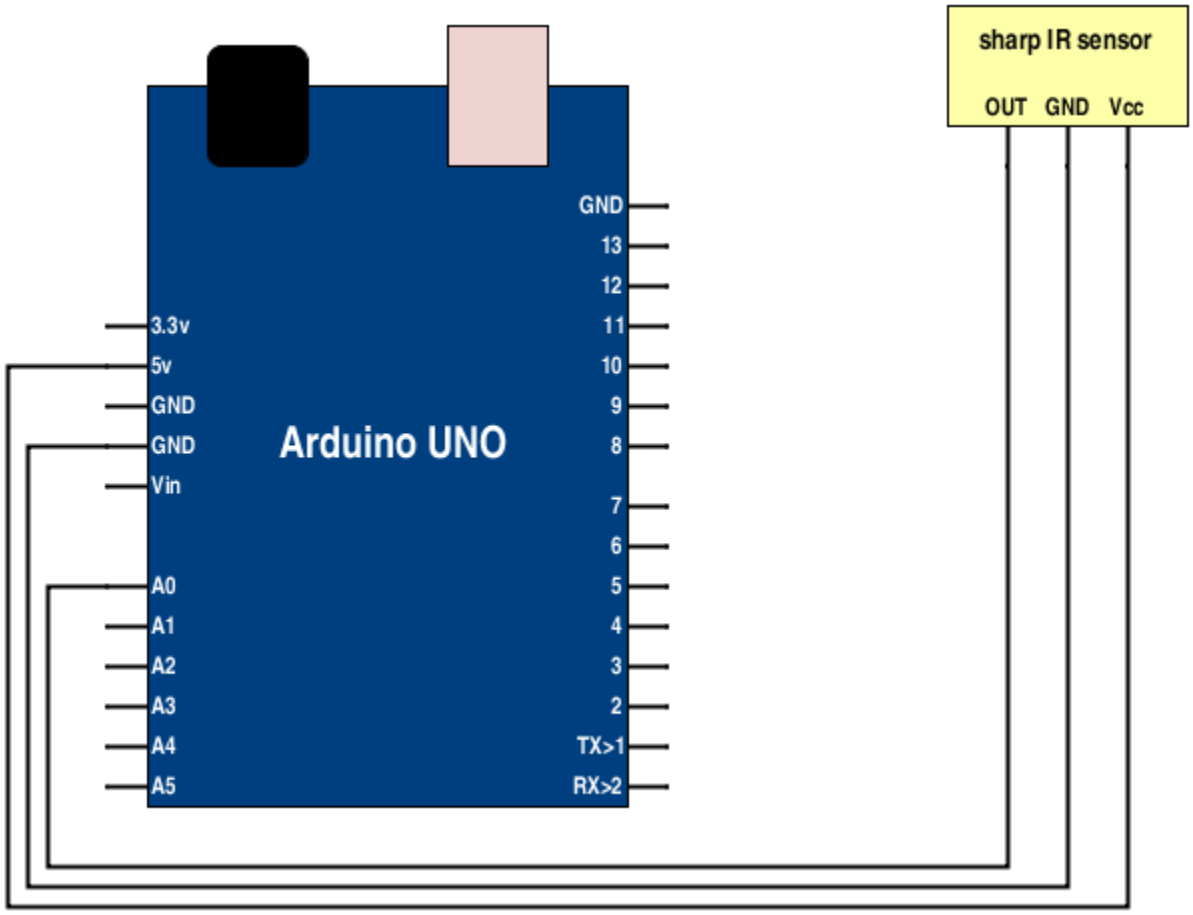


Figure 4-2:Sharp IR connection diagram

Ensure that your Arduino is powered off. Now insert the red wire into the +5V power pin and the black wire into the ground power pin of the Arduino. Insert the yellow wire into the A0 analogue pin. Plugging in the hardware is now complete. Let's move on to the programming bit.

Now let`s move to the code :

```
int sensorpin = 0;              // analog pin used to connect the sharp sensor.

int val = 0;                    // variable to store the values from sensor(initially
                                   zero).
void setup()
{
  Serial.begin(9600);          // starts the serial monitor.
}

void loop()
{
 val = analogRead(sensorpin);       // reads the value of the sharp sensor

 Serial.println(val);             // prints the value of the sensor to the serial
                                  monitor.

  delay(100);                     // wait for this much time before printing next value
}
```

Force Sensor

The force sensor will vary its resistance depending on how much pressure is being applied to the sensing area ,to measure the force using one of the arduino's analog inputs we need a fixed resistor that we can use for that comparison (we are using 10k resistor) this is called voltage divider and divides the 5v between the force sensor and the resistor.

Figure 4-3: Force sensor (flex sensor)

In this experiment You'll need:

1-Arduino Uno

2-Force sensor

3-10k Ohm resistance

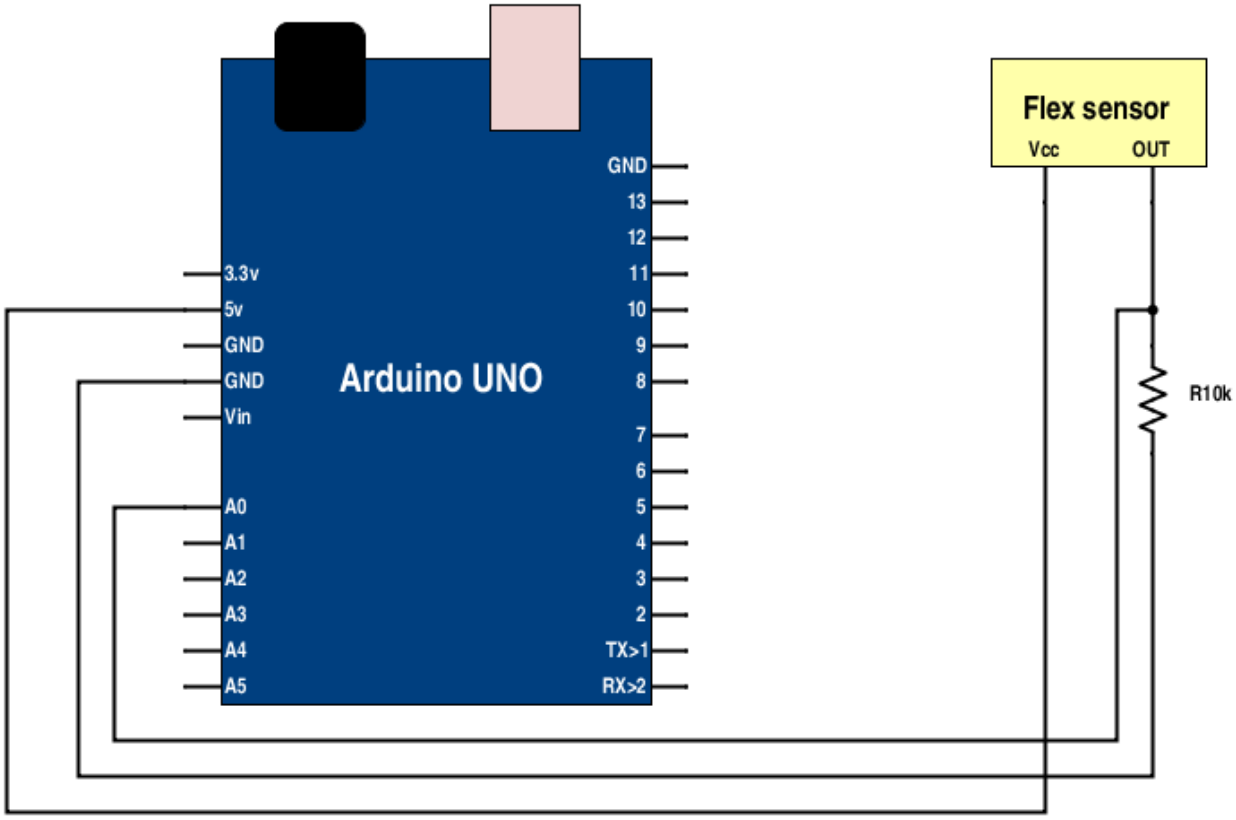


Figure 4-4:Force sensor connection diagram

Now lets move to the code:

```
int fsrReading;

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

void loop(void) {
  fsrReading = analogRead(A0);
  Serial.print("Analog reading = ");
  Serial.println(fsrReading);

  delay(100);
}
```

Motion Sensor

PIR motion sensor is used to detect any motion from about 19 feet away

This sensor is a digital sensor so the output is either high or low.

Figure 4-5:PIR motion sensor

In this experiment you'll need :

1-PIR motion sensor

2-ARduino uno

To connect the motion sensor to arduino you have to connect vcc pin to 5v pin in arduino GND pin to GND and the out pin to any digital pin from 2-13

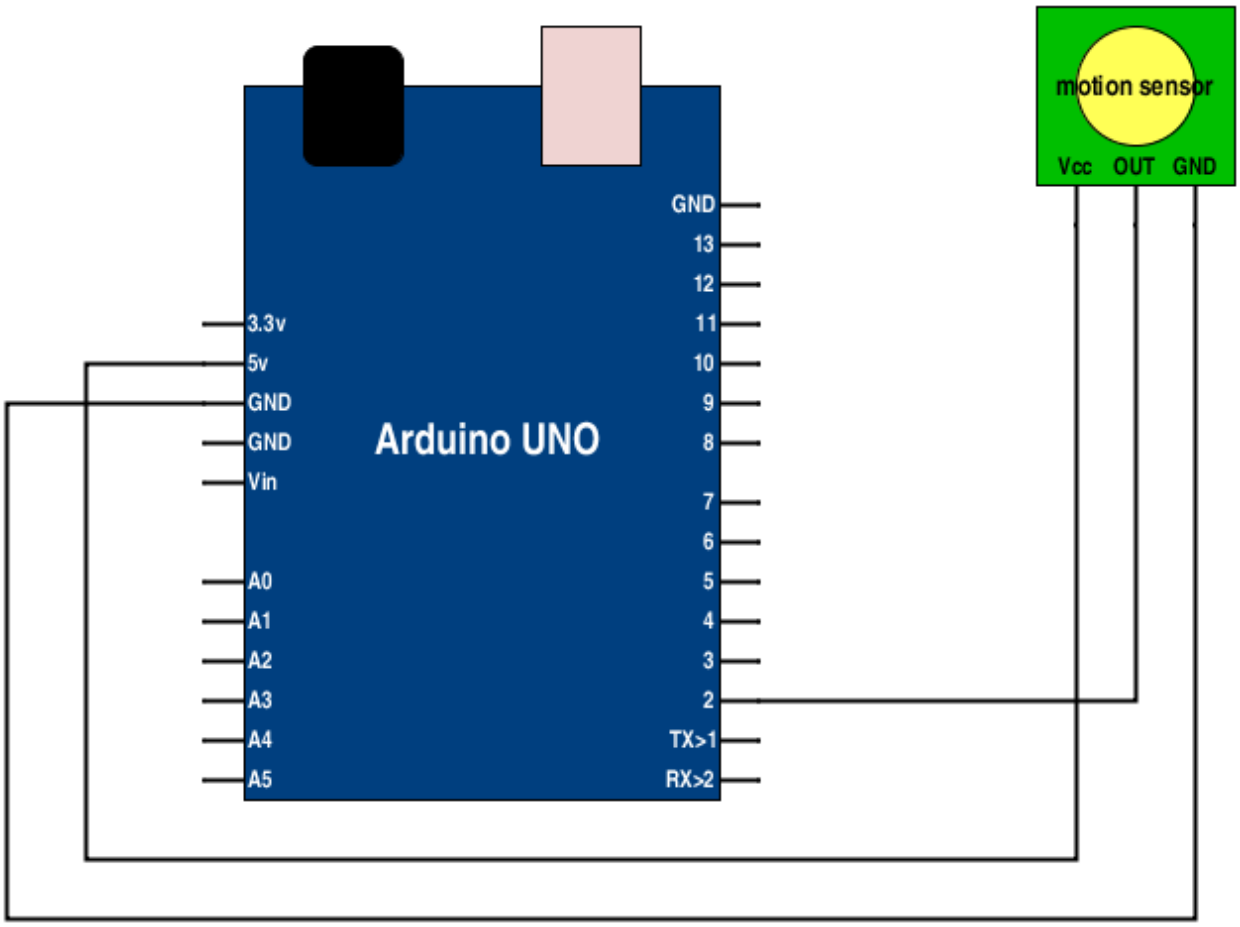


Figure 4-6: PIR motion sensor connection diagram

Now let's move to the code:

```
Int inpin=2;

Int  value;

Void setup(){

Serial.begin(9600);

pinMode(2,INPUT);

}

Void loop(){
```

```
Serial.println(Value);

If (Value==1){

Serial.println(“Motion Detected”);

}

}
```

Ultrasonic sensor

The HC-SR04 ultrasonic sensor uses ultrasonic sound to measure distance, This popular ultrasonic distance sensor provides stable and accurate distance measurements from 2cm to 450cm.

Figure 4-7: Ultrasonic sensor

In this experiment you'll need :

1- Ultrasonic sensor.

2-ARduino uno .

To connect the Ultrasonic sensor to arduino you have to connect vcc pin to 5v pin in arduino, GND pin to GND pin in arduino, Trigger pin (Trig) to to a digital output pin in arduino , Echo pin to a digital input in arduino .

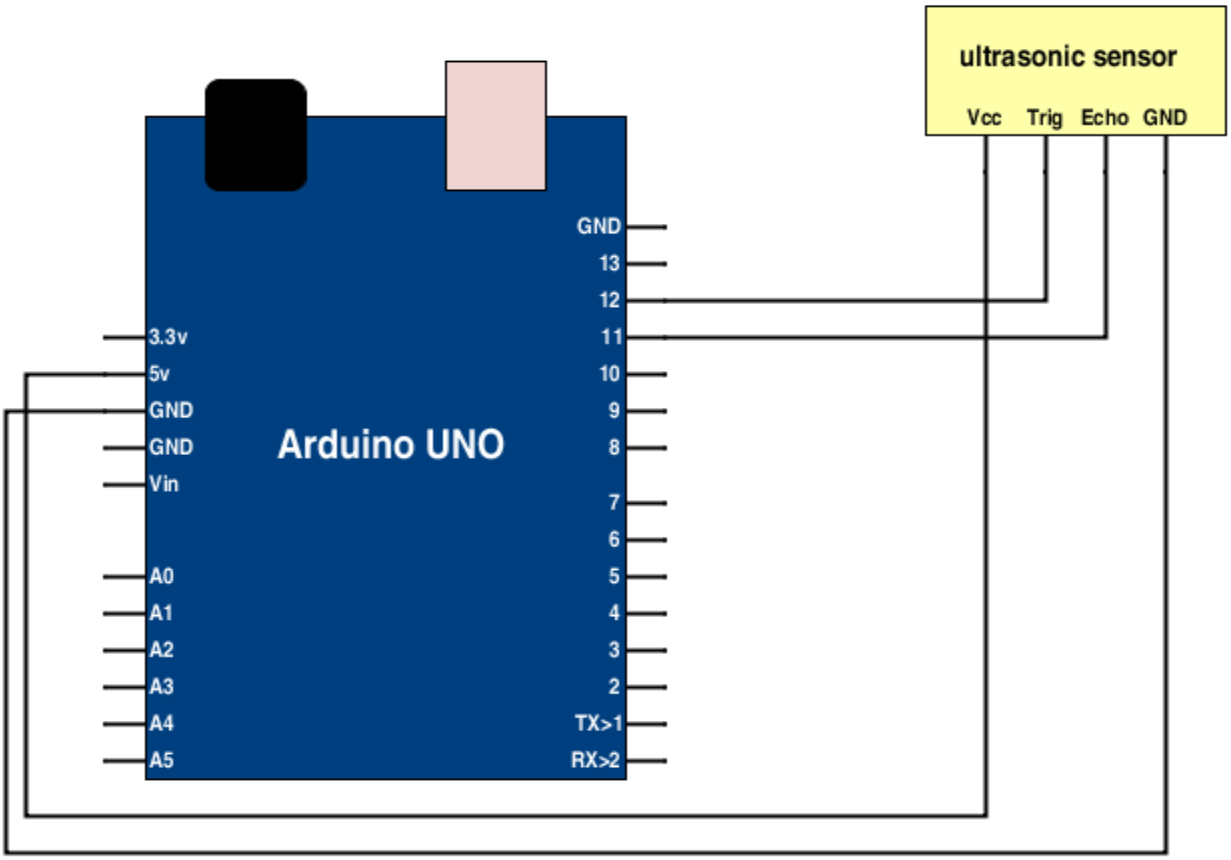


Figure 4-8: Ultrasonic sensor connection diagram

Now let's move to the code:

```
#include <Ultrasonic.h>

Ultrasonic ultrasonic(12,11);      // (Trig PIN,Echo PIN)

void setup() {

 Serial.begin(9600);

}

void loop()

{

 Serial.print(ultrasonic.Ranging(CM));      // CM or INC
```

Gas sensor

The smoke sensor we will use is the MQ-2. This is a sensor that is not only sensitive to smoke, but also to flammable gas.

The MQ-2 smoke sensor reports smoke by the voltage level that it outputs. The more smoke there is, the greater the voltage that it outputs. Conversely, the less smoke that it is exposed to, the less voltage it outputs.

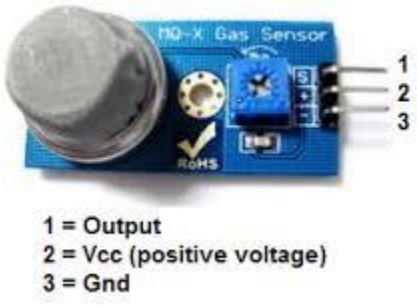


Figure 4-9: Gas sensor board

In this experiment you'll need :

1- Gas sensor board .
2- arduino uno .
3- Buzzer .

To connect the Gas sensor board to arduino you have to connect first pin to analog output in arduino , second pin to 5v pin in arduino , third pin to GND pin in arduino .

To connect the buzzer connect the black wire to GND pin in arduino and the red wire to digital output pin in arduino .

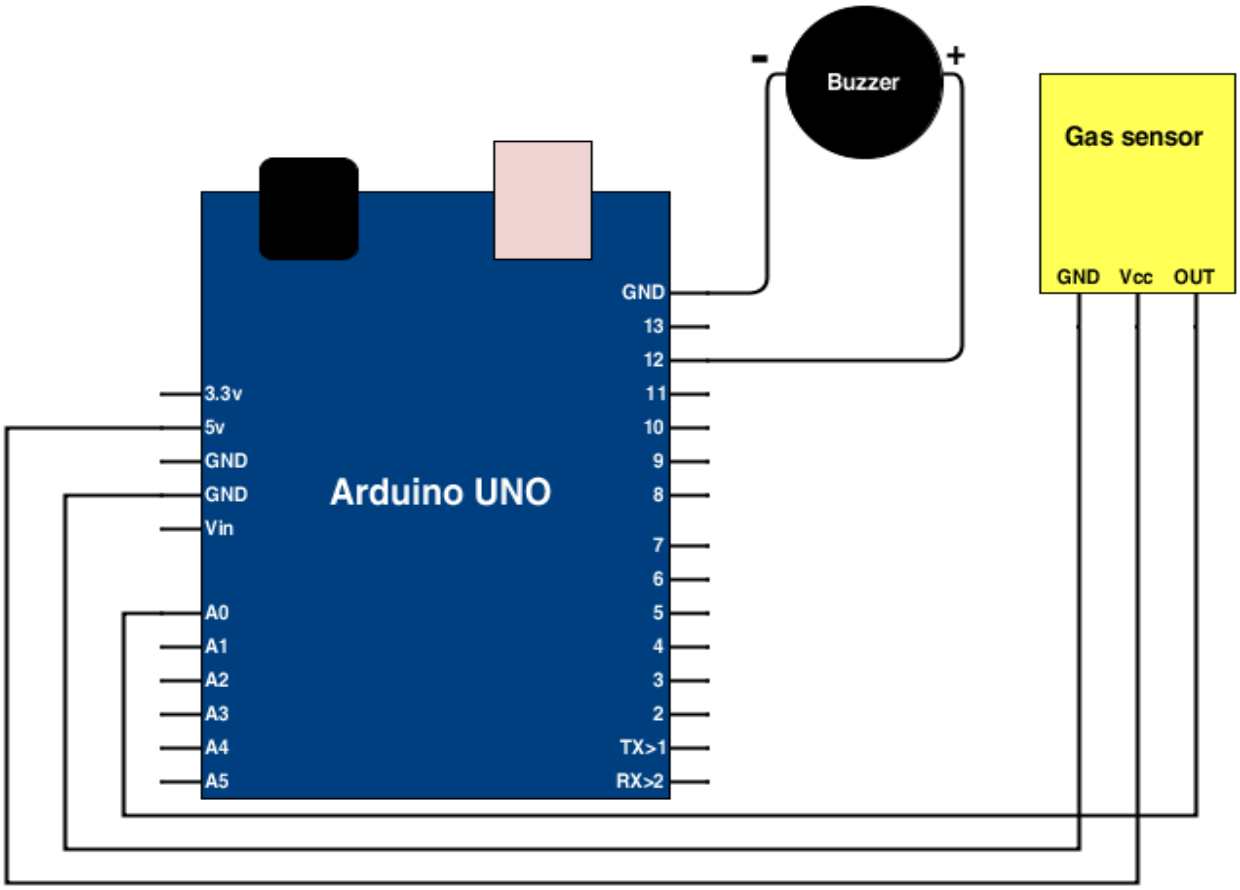


Figure 4-10: Gas sensor and buzzer with arduino.

Now let's move to the code:

```
Const                                                    intsensorPin=0;
constintbuzzerPin=12;
intsmoke_level;

voidsetup(){
Serial.begin(115200);                    //sets the baud rate for data transfer in bits/second
pinMode(sensorPin, INPUT);       //the smoke sensor will be an input to the arduino


pinMode(buzzerPin, OUTPUT);       //the buzzer serves an output in the circuit
}

void loop() {
```

```
if(smoke_level > 500){                  //if smoke level is greater than 500, the buzzer will go oN
digitalWrite(buzzerPin,HIGH);
}
else{
digitalWrite(buzzerPin,LOW);

}
```

Flame sensor

The flame sensor used to detect fire or other wavelength at 760 nm ~ 1100 nm light, and it's operating temperature is -25 degrees Celsius to 85 degrees Celsius

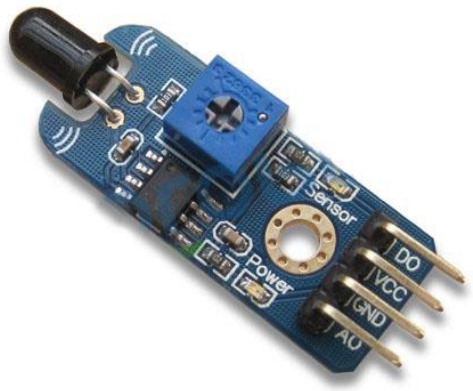


Figure 4-11: Flame sensor .

In this experiment you'll need :

1- Flame sensor .
2- arduino uno .
3- Buzzer .

To connect the Flame sensor to arduino you have to connect Vcc pin to 5v in arduino , GND pin to GND pin in arduino , analog output (A0)pin to analog output pin in arduino .

To connect the buzzer connect the first wire to GND pin in arduino and the second wire to digital output pin in arduino .

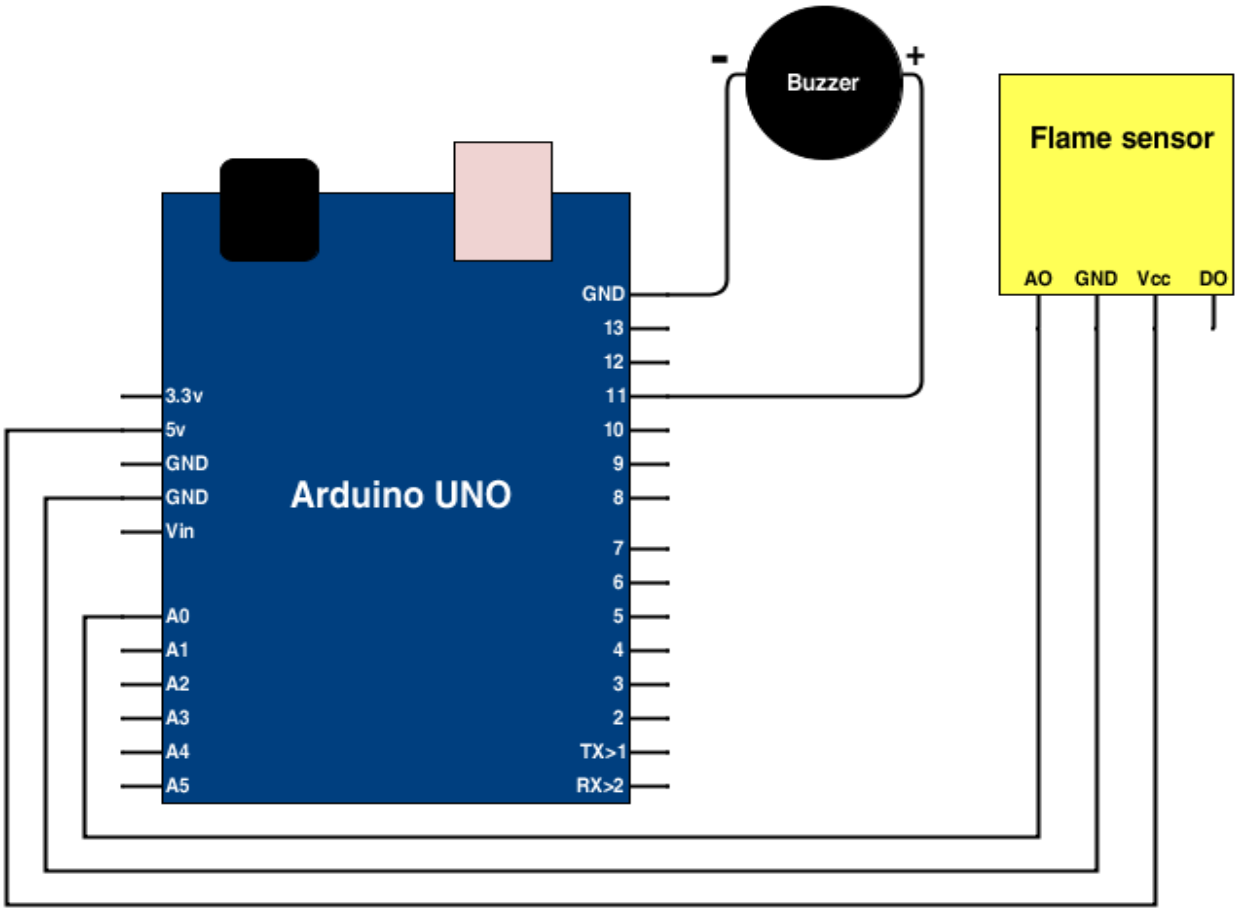


Figure 4-11: Flame sensor and buzzer with arduino .

Now let's move to the code:

```
void setup ()
{
  Serial.begin(9600);
  pinMode(A0, INPUT);
  pinMode (11, OUTPUT);
}

void loop ()
{
  Serial.print ("Sensor Value: ");
  int x = analogRead (A0);
  Serial.println (x);
```

```
    digitalWrite (11, HIGH);
  }
  else
  {
    digitalWrite (11, LOW);
  }
  delay (1000);
}
```

Vibration sensor

The vibration sensor used to test electric level and if the environment is in vibration, alarm function.

Suitable for a variety of vibration triggering functions

.

Figure 4-12 : vibration sensor .

In this experiment you'll need :

1- vibration sensor .
2- arduino uno .

To connect the vibration sensor to arduino you have to connect Vcc pin to 5v in arduino , GND pin to GND pin in arduino , digital output (DO)pin to digital pin in arduino .

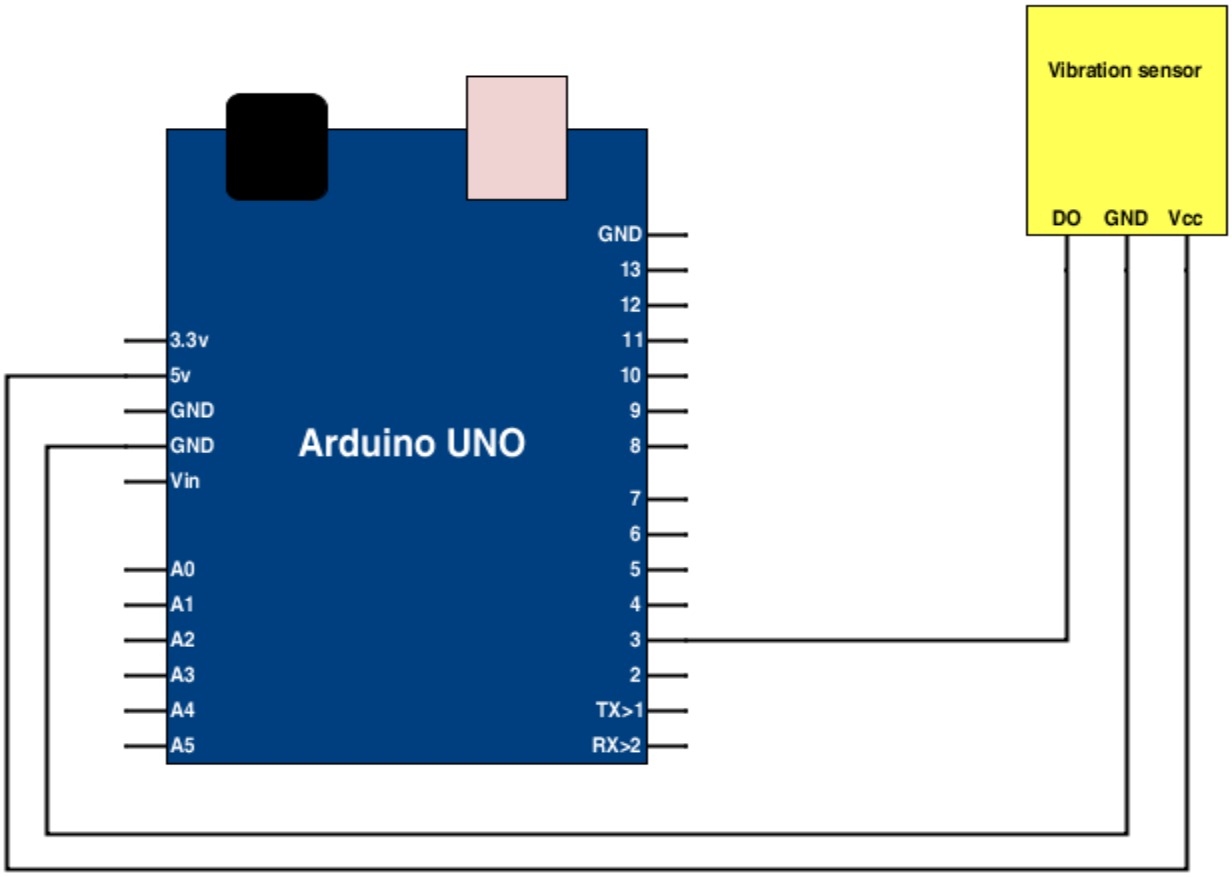


Figure 4-13: vibration sensor with arduino connection diagram .

Now let`s move to the code :

```
int ledPin = 13;
int vibration_sensor = 3;
 void setup()
{
 pinMode(ledPin, OUTPUT);      // Set digital pin 13 to output mode
 pinMode(vibration_sensor, INPUT);       // Set digital pin 3 to input
mode
}
void loop()
```

```
{

  if(digitalRead(vibration_sensor)==HIGH) //Read sensor value
    {
       digitalWrite(ledPin, HIGH);   // Turn on LED when the sensor is
tilted
       delay(300);
    }
  else
    {
       digitalWrite(ledPin, LOW);    // Turn off LED when the sensor is
not triggered
    }
}
```

Sound sensor

Sound sensor is used to detect sound .

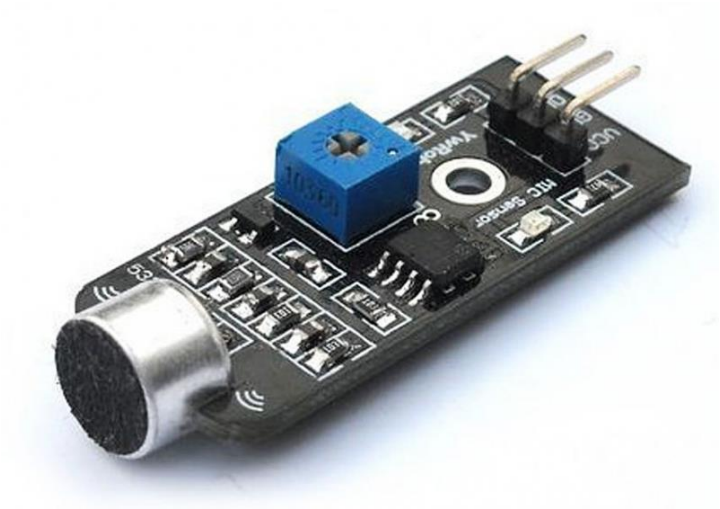
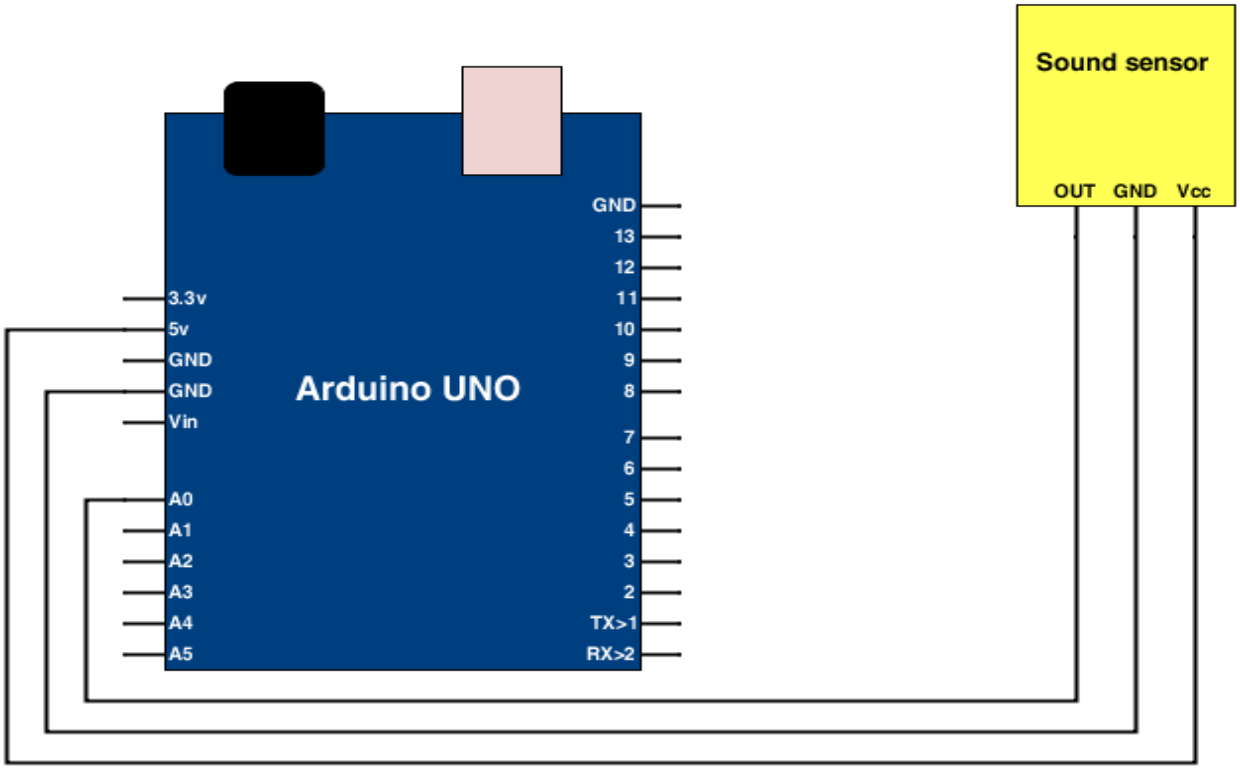


Figure 4-14 : sound sensor .

In this experiment you'll need :

1- Sound sensor .
2- arduino uno .

To connect the Sound sensor to arduino you have to connect Vcc pin to 5v in arduino , GND pin to GND pin in arduino ,and output pin to analog pin in arduino .



Now let`s move to the code:

```
int sensorpin = A0;              // analog pin used to connect the sound sensor.

int val = 0;                    // variable to store the values from sensor(initially
                                   zero).
void setup()
{
  Serial.begin(9600);          // starts the serial monitor.
}

void loop()
{
 val = analogRead(sensorpin);        // reads the value of the sound sensor

 Serial.println(val);              // prints the value of the sensor to the serial
                                   monitor.

  delay(100);                      // wait for this much time before printing next value
}
```

Voltage sensor is based on principle of resistive voltage divider design. and it can make the input voltage of red terminal reduce 5 times of original voltage.

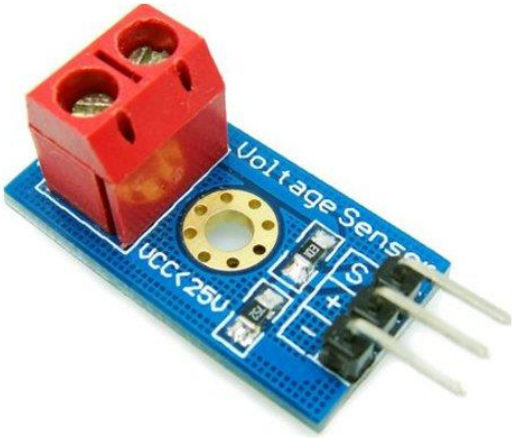


Figure 4-16 : voltage sensor .

In this experiment you'll need :

1- voltage sensor .
2- arduino uno .

To connect the voltage sensor to arduino you have to connect Vcc(+) pin to 5v in

Arduino , GND (-) pin to GND pin in arduino , signal (s) pin to analog input pin in arduino .

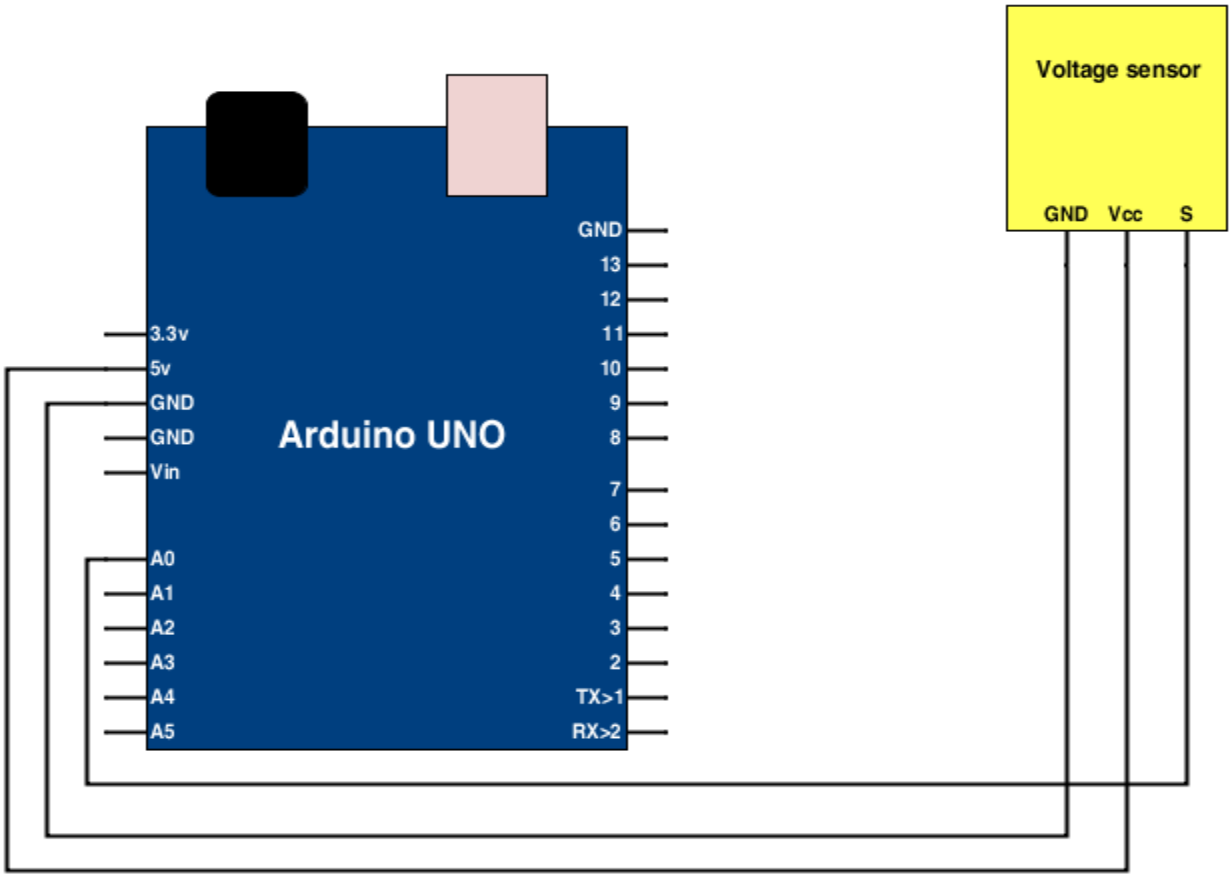


Figure 4-17 : voltage sensor with arduino connection diagram.

Now let`s move to the code :

```
# include <Wire.h>      //  we use a ready library from arduino  to deal with voltage sensor and to make the program easier to write .

int val11;
int val2;

void setup ()
{
  pinMode (LED1, OUTPUT);
  Serial.begin (9600);
  Serial.println ("Voltage:");
  Serial.print ("V");
}
void loop ()
{
  float temp;
  val11 = analogRead (1);
  temp = val11/4.092;
  val11 = (int) temp ;/ /
```

```
  val2 = ((val11% 100) / 10);
  Serial.println (val2);
  delay (1000);


}
```

LDR sensor

LDR sensor (photo resistor) is light sensitive device most often used to indicate the presence or absence of light, or to measure the light intensity. In the dark, their resistance is very high, sometimes up to 1MΩ, but when the LDR sensor is exposed to light, the resistance drops dramatically, even down to a few , depending on the light intensity.

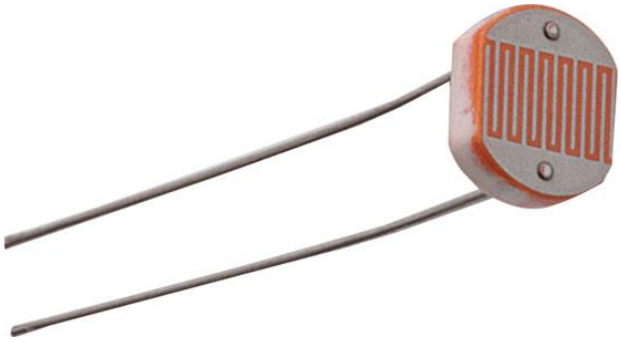


Figure 4-18 : LDR sensor.

In this experiment you'll need :

1- voltage sensor .
2- arduino uno .
3- two resistor (100Ω , 320Ω).
4- wires .
5- LED .

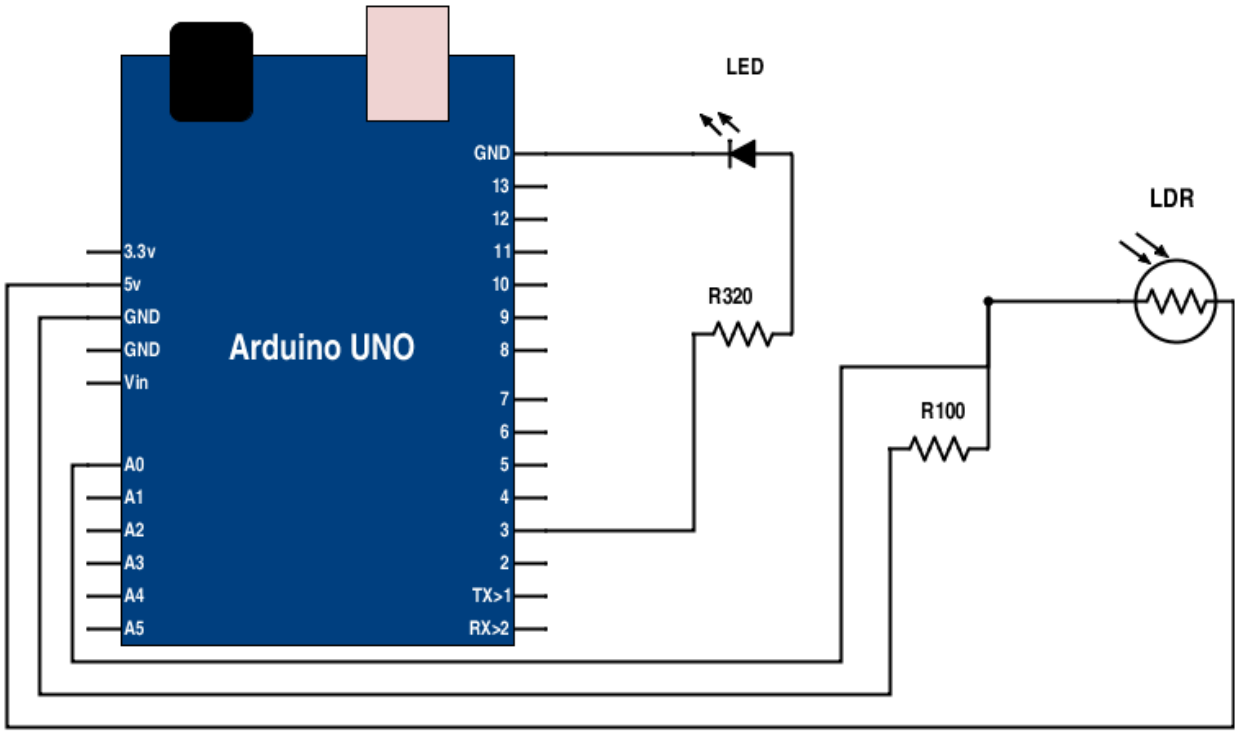


Figure 4-19 : LDR sensor with arduino connection diagram.

Now let's move to the code :

```
int ldr ;
int ledpin=3;
void setup()
{
  pinMode(ledpin,OUTPUT);
}
void loop(){
ldr =analogRead(A0);
if(ldr>100){
```

```
  digitalWrite(ledpin,0);
 }
 if(ldr<100) {
  digitalWrite(ledpin,1);
 }
 }
```

Joystick sensor

Joystick sensor contain two potentiometers and button to control position ; Two of the input value of potentiometer are respectively to show the user offset in the X and Y axis, and the type of offset is analog ; the button is used to show whether the user push down button in the Z axis, its type is the digital quantity.

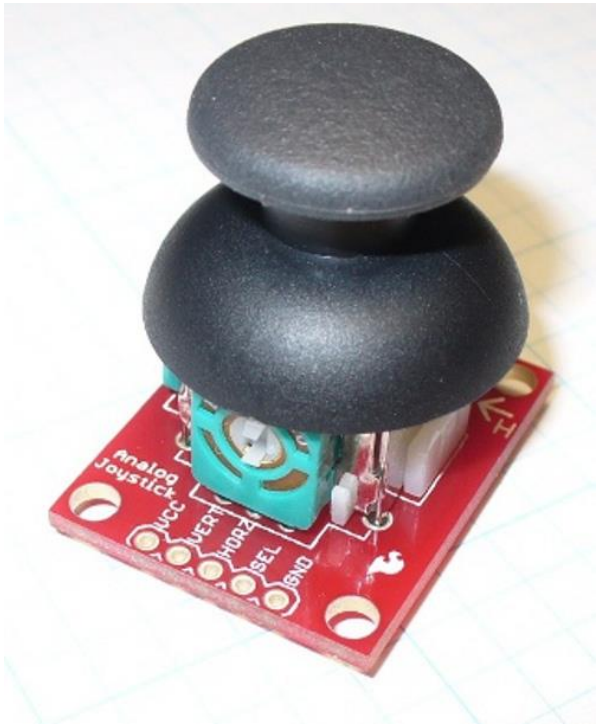


Figure 4-20 : Joystick sensor.

In this experiment you'll need :

1- Joystick sensor.
2- arduino uno .

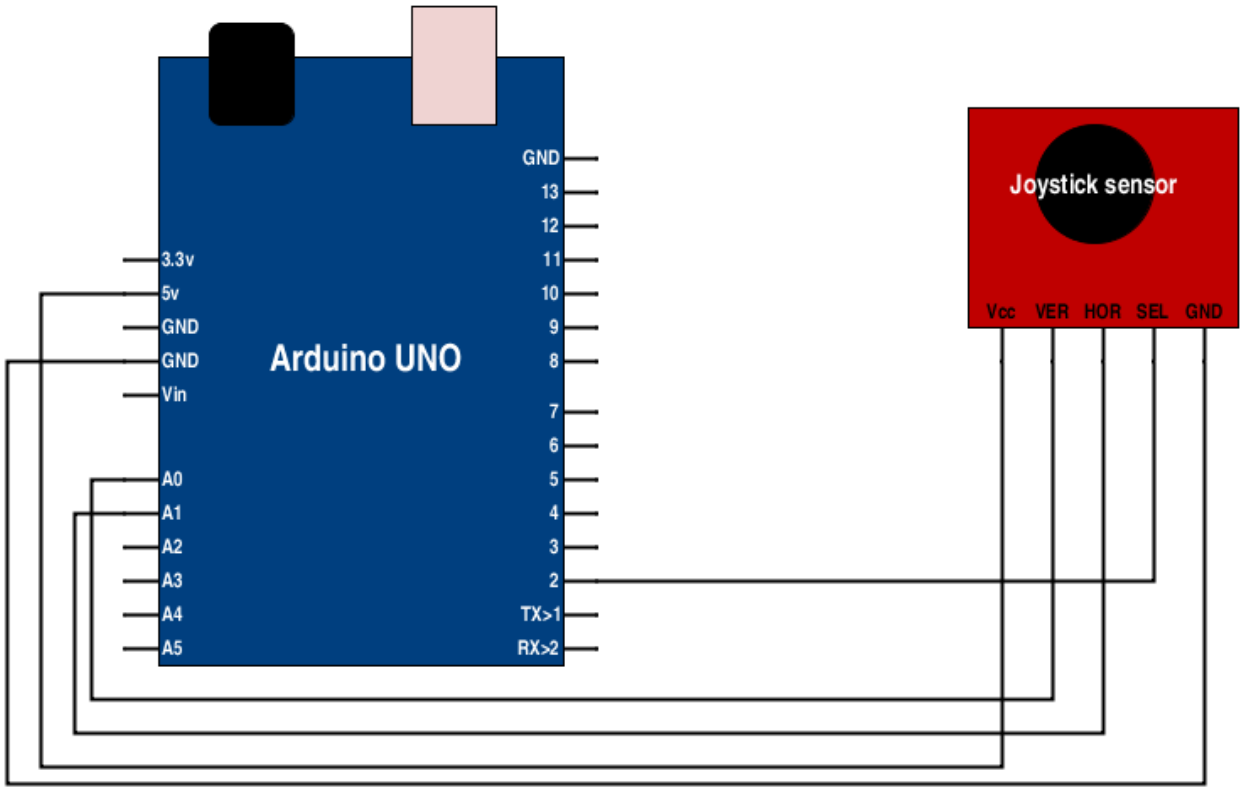


Figure 4-20 : Joystick sensor with arduino connection diagram.

Now let's move to the code :

```
int verPin = A0;
int horPin = A1;
int selPin = 2;

void setup() {
  pinMode(selPin, INPUT);
  digitalWrite(selPin, HIGH);
}

void loop() {
  int verPos = analogRead(verPin);
  int horPos = analogRead(horPin);
  boolean selBtn = digitalRead(selPin);
}
```

Temperature sensor

Temperature sensor (LM35) used to measure temperature with an electrical output proportional to the temperature (in ºC) .

Figure 4-21 : Temperature sensor(LM35).

In this experiment you'll need :

1- Temperature sensor(LM35).
2- Breadboard .
3- arduino uno .

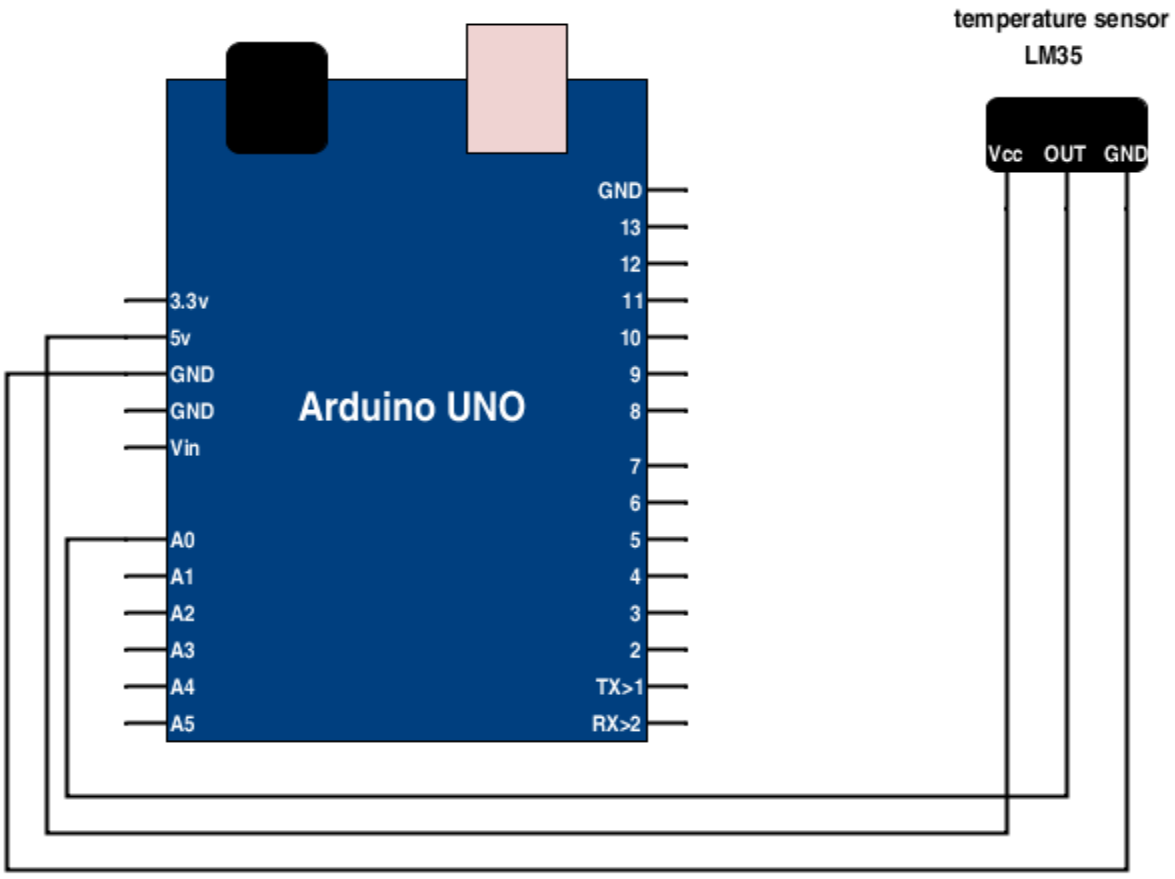


Figure 4-22 : Temperature sensor with arduino connection diagram.

Now let's move to the code :

```
int temp =0 ;
void setup(){
Serial.begin(9600);
}
void loop(){
  temp = analogRead(A0)/2 ;
  Serial.println(temp);
  deay(20);
}
```

Accelerometer sensor :

Accelerometer sensor is used to measure angle , acceleration and angular velocity in three dimensions (x,y,z) .

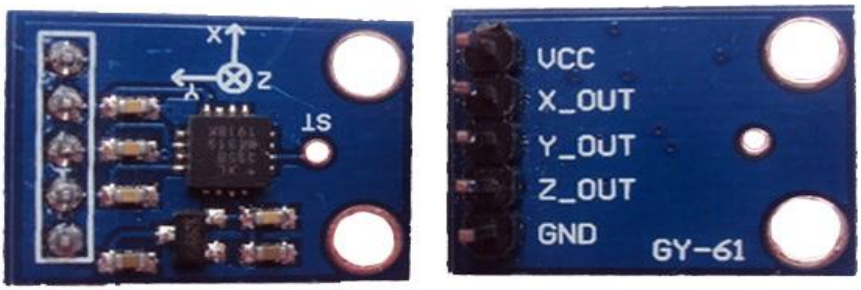


Figure : Accelerometer sensor.

In this experiment you'll need :

1- Arduino Uno .
2- Accelerometer sensor .
3- Wires .

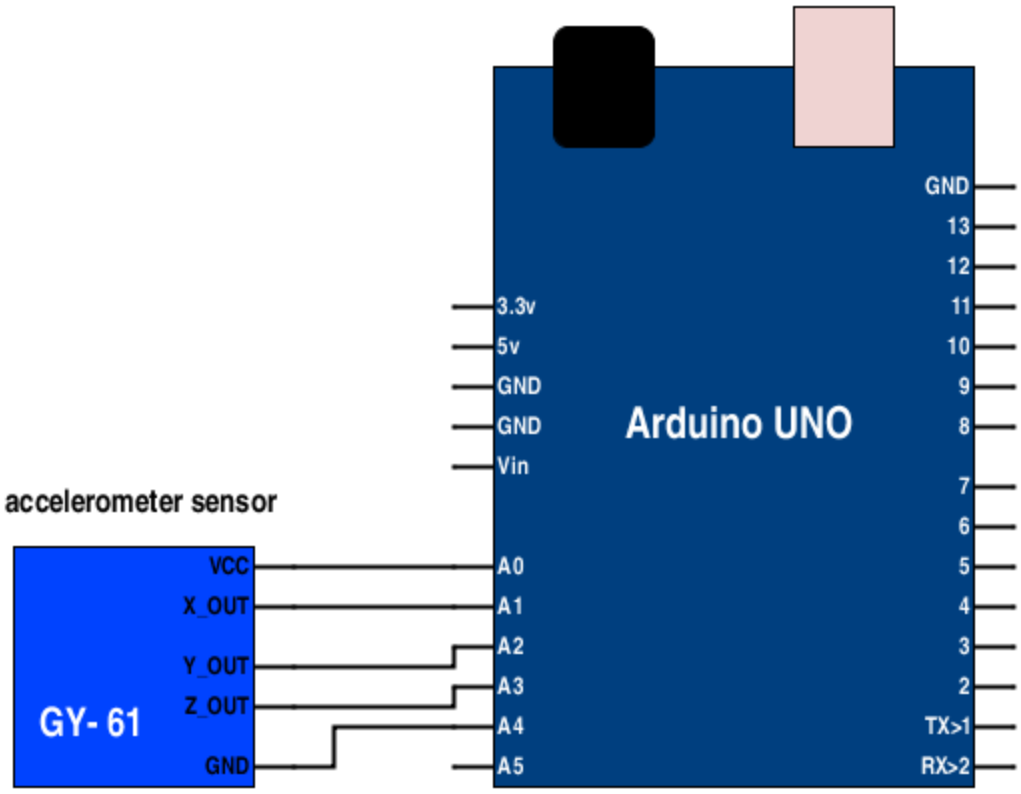


Figure : Accelerometer sensor with arduino connection diagram.

Now let`s move to the code :

```
int x,y,z;

int vcc_pin=0;

int gnd_pin=4;

int x_pin=1;

int y_pin=2;

int z_pin=3;


void setup(){

Serial.begin(9600);

pinMode(vcc_pin,OUTPUT);
```

```
pinMode(gnd_pin,OUTPUT);
digitalWrite(vcc_pin,1);
digitalWrite(gnd_pin,0);
}

void loop(){
  x=analogRead(x_pin);
  y=analogRead(y_pin);
  z=analogRead(z_pin);

  Serial.print("x=  ");
  Serial.print(x);
  Serial.print("       y=  ");
  Serial.print(y);
  Serial.print("       z=  ");
  Serial.println(z);
}
```

Keypad :

4x4 keypad provides a useful interface component for microcontroller projects such as : security systems , menu selection , data entry for embedded systems .

Figure : 4x4 Keypad.

In this experiment you'll need :

1- Arduino Uno .
2- 4x4 keypad .
3- Wires .

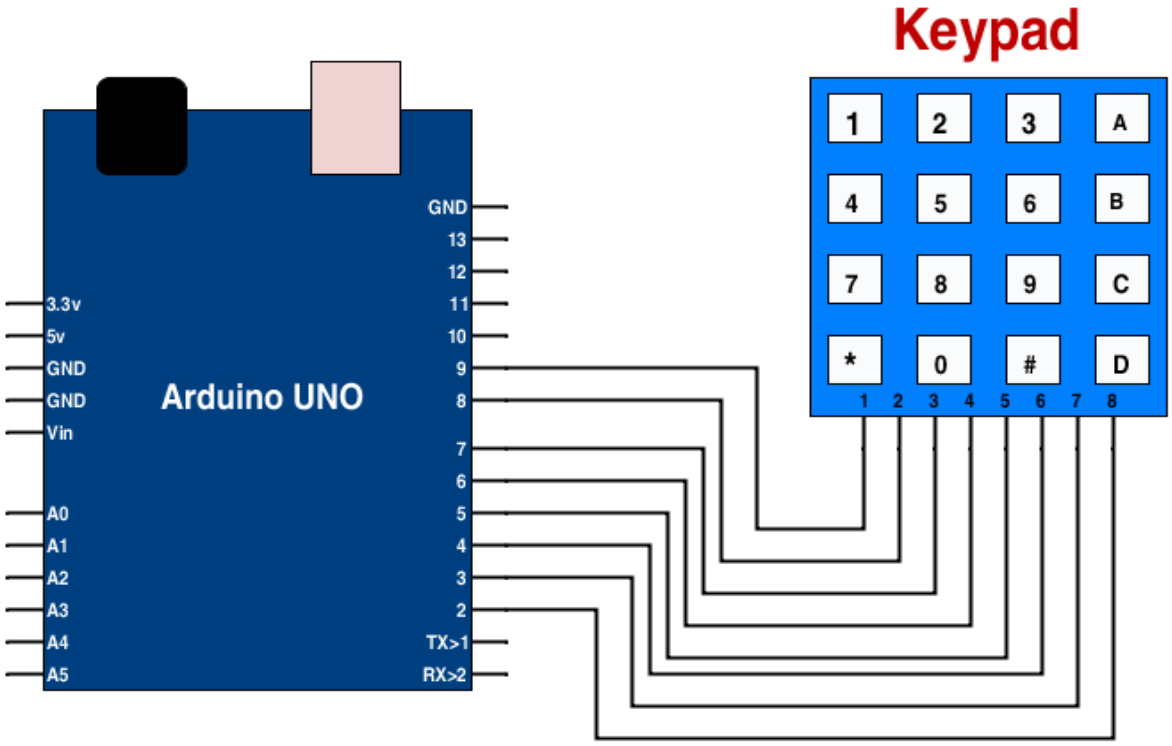


Figure : 4x4 keypad with arduino connection diagram.

Now let`s move to the code :

```
#include <Keypad.h>


const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

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

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

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

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

};
```

```
byte rowPins[ROWS] = {2,3,4,5}; //connect to row pinouts

byte colPins[COLS] = {6,7,8,9}; //connect to column pinouts



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



void setup(){

  Serial.begin(9600);

}



void loop(){

  char key = keypad.getKey();



  if (key != NO_KEY){

    Serial.println(key);

  }

}
```

Tiny RTC :

Real Time Clock (RTC)Module for arduino .

Figure : Tiny RTC module .

In this experiment you'll need :

1- Arduino Uno .
2- Tiny RTC module .
3- Wires .

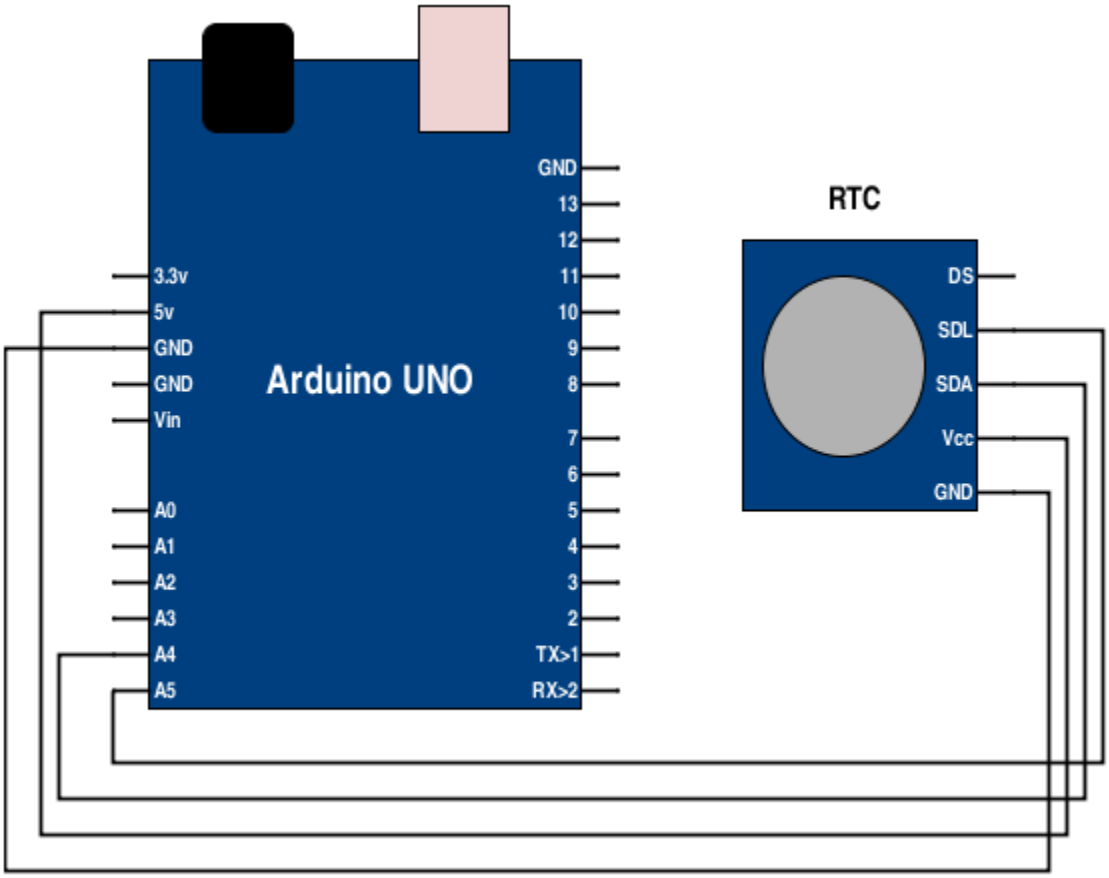


Figure : Tiny RTC module with arduino connection diagram .

Now let`s move to the code :

```
#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 rtc;

void setup () {
 Serial.begin(57600);
 #ifdef AVR
  Wire.begin();
 #else
```

```
  Serial.print(now.day(), DEC);
    Serial.print(' ');
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    Serial.print(now.second(), DEC);
    Serial.println();


    delay(3000);
}
```

DHT-11 sensor :

DHT-11 is a sensor for measuring temperature and humidity .

Figure : DHT-11 sensor .

In this experiment you will need :

1- Arduino board .
2- DHT-11 sensor .
3- Wires .

DHT-11 sensor have four pins : VCC ,DATA ,NC and GND .(from left to write)

And connected to arduino like connection diagram below .

See connection diagram :

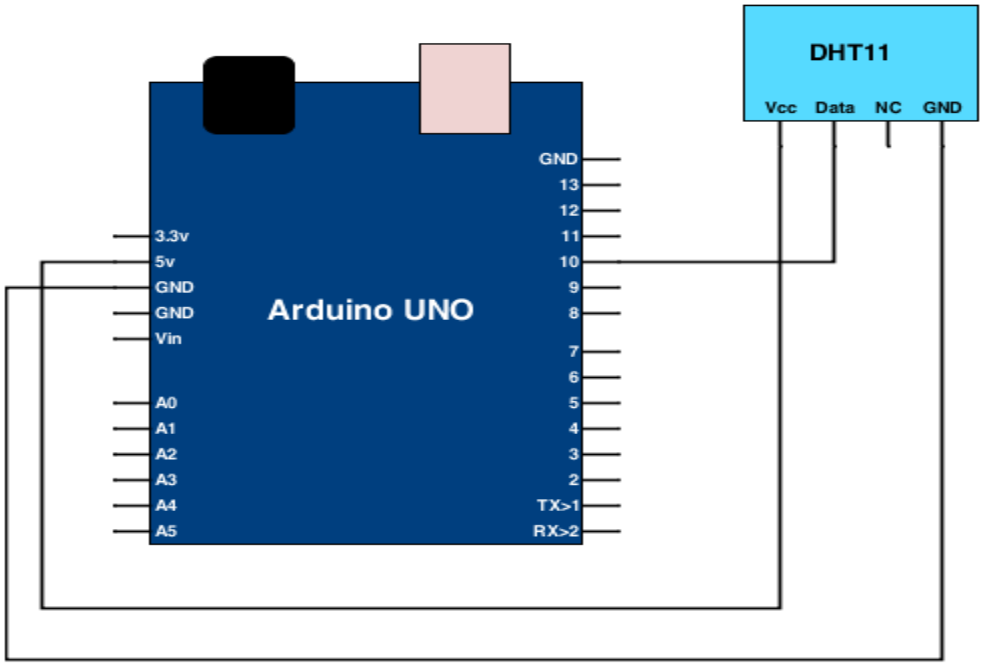


Figure : DHT-11 sensor with arduino connection diagram .

Now let`s move to the code :

```
#include "DHT.h"
#define DHTPIN 10    // what pin we're connected to
// Uncomment whatever type you're using!
#define DHTTYPE DHT11   // DHT 11
//#define DHTTYPE DHT22   // DHT 22  (AM2302)
//#define DHTTYPE DHT21   // DHT 21 (AM2301)

// Initialize DHT sensor for normal 16mhz Arduino
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");
 dht.begin();
 }
```

```
void loop() {
  // Wait a few seconds between measurements.
  delay(2000);
 // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit
  float f = dht.readTemperature(true);
  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // Compute heat index
  // Must send in temp in Fahrenheit!
  float hi = dht.computeHeatIndex(f, h);
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.print(" *F\t");
  Serial.print("Heat index: ");
  Serial.print(hi);
  Serial.println(" *F");
}
```

IR Transmitter + IR Receiver :

IR led transmitter send undetectable light (inferred signal) and received it by IR Receiver ; you can use this communication in many application like counting , send special code , make an IR remote …. etc

Figure : led IR Transmitter .

Figure : IR Receiver .

In this experiment you will need :

1- Two arduino board.

2- IR LED transmitter .

3- IR LED receiver.

4- push button .

5- 10K ohm resistor .

6- wires .

See connection diagram :

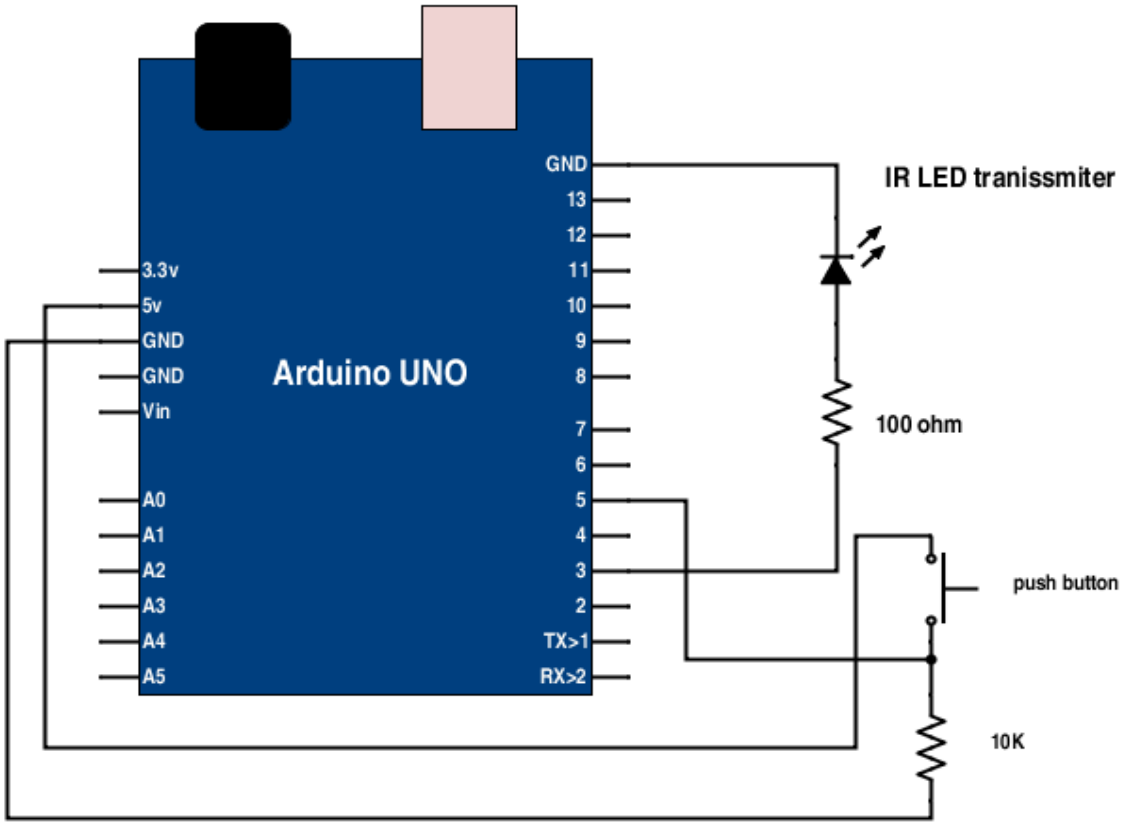


Figure : IR LED Transmitter with arduino connection diagram .

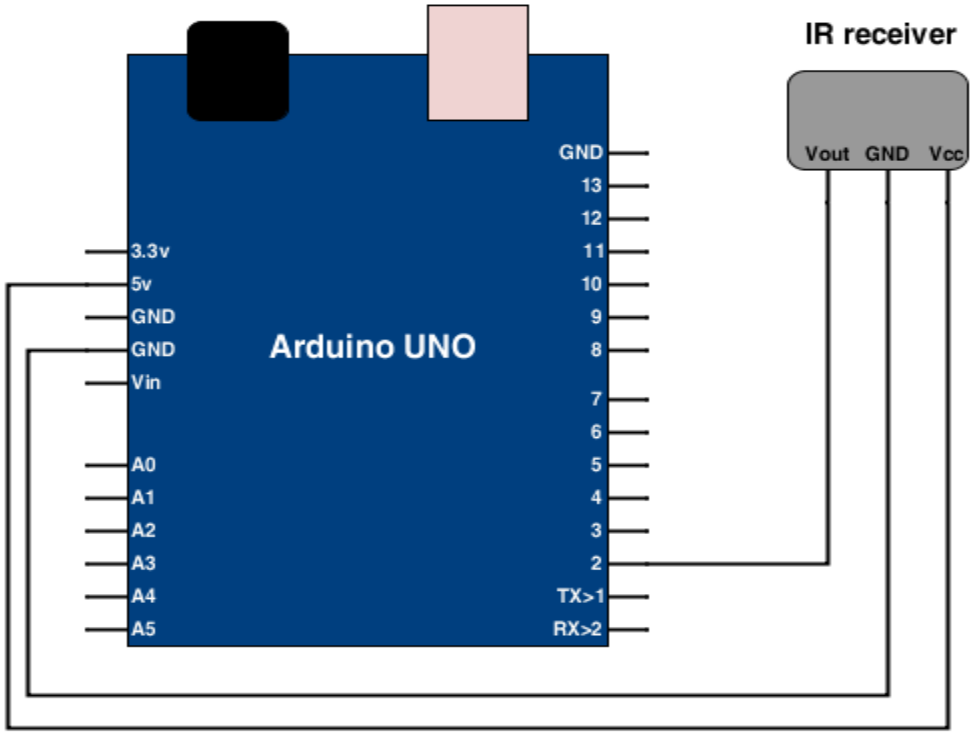


Figure : IR Receiver connection diagram .

Now let`s move to the code :

Transmitter code :

```
#include <IRremote.h>

IRsend irsend;

int pb = 5 ;

int pb_state ;

void setup()

{

  Serial.begin(9600);

  pinMode (pb,INPUT);

}

void loop() {

  if (Serial.read() != 1) {

    for (int i = 0; i < 3; i++) {

    pb_state = digitalRead(pb);

      if (pb_state==1){



      irsend.sendSony(0xa50, 12);

      delay(40);

      }

    }

  }

}
```

Receiver code :

```
#include <IRremote.h>

int RECV_PIN = 2;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()

{

  Serial.begin(9600);

  irrecv.enableIRIn(); // Start the receiver

}

void loop() {

  if (irrecv.decode(&results)) {

    Serial.println(results.value, HEX);

    irrecv.resume(); // Receive the next value

  }

}
```