



# **Embedded Systems** **(EPM)**

---

**Lecture (5) Summary**

## **Sensor Types in this Lec. :**

- **Tilt Sensors**
- **Light Sensors**
- **Motion Detection Sensors**
- **Vibration Detection Sensors**
- **Audio Sensors**
- **Temperature Sensors**
- **Identification Sensors**

## **Tilt Sensors:**

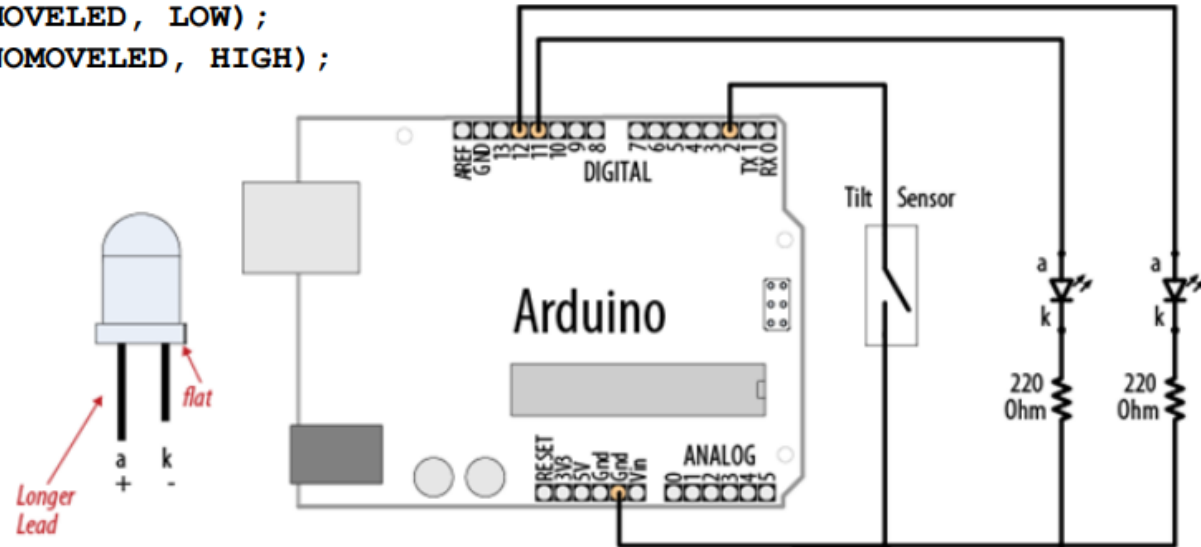
**Tilt sensor detect inclines using conducting liquid like mercury or rolling ball (بيحس ب أي ميل يحصل).**

**It has 2 Models:**

- Rolling Ball Mode: 107-2001-EV**
- Mercury Model: CM1320-0**

# Tilt Sensors

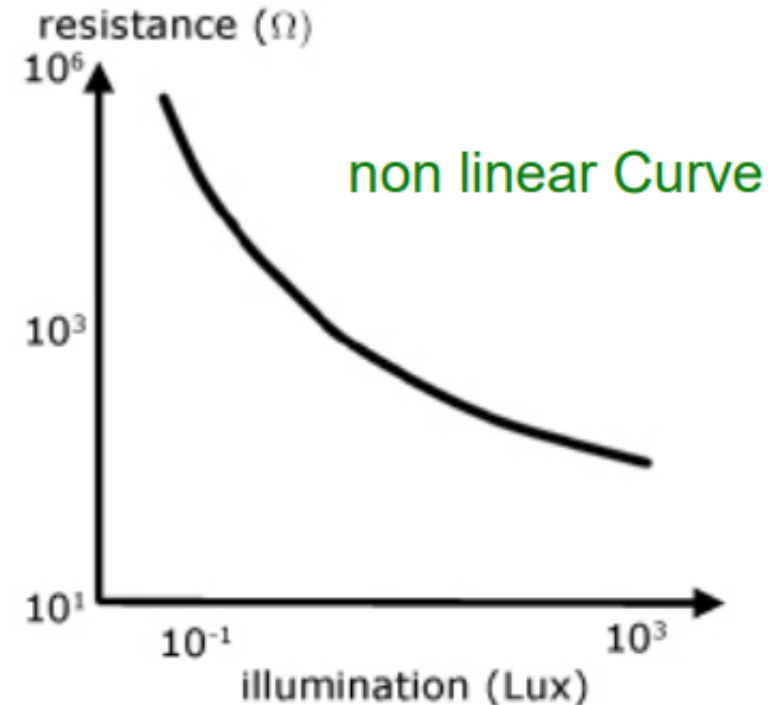
```
#define TILT 2
#define MOVELED 11
#define NOMEVELED 12
void setup()
{
    pinMode (TILT, INPUT);
    digitalWrite (TILT, HIGH); initially turned ON
    pinMode (MOVELED, OUTPUT);
    pinMode (NOMEVELED, OUTPUT);
}
void loop()
{
    if (digitalRead(TILT)){
        digitalWrite(MOVELED, HIGH);
        digitalWrite(NOMEVELED, LOW);
    }
    else{
        digitalWrite(MOVELED, LOW);
        digitalWrite(NOMEVELED, HIGH);
    }
}
```



## Light Sensors:

- Variable Resistance Sensitive to Light Intensity
- Light intensity is measured using Lux (بيحس ب شدة الاضاءة)

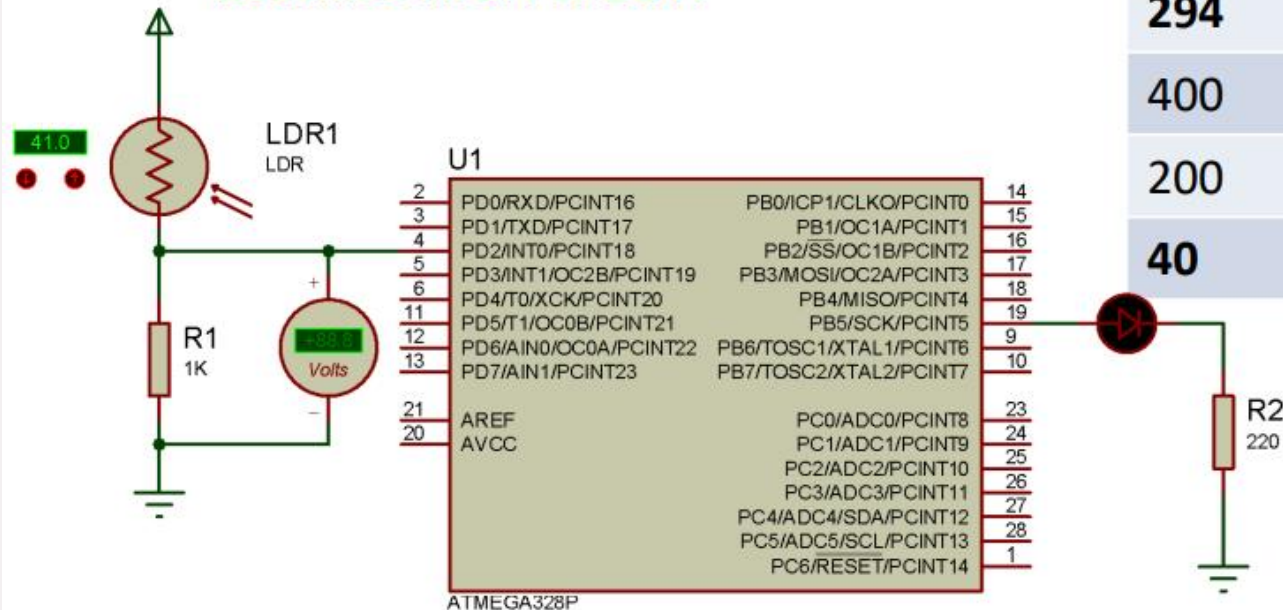
- Moonless clear night : 0.002 Lux
- Full Moon clear night : 0.1 Lux
- Home light : 50 Lux
- Cloudy day : 100 Lux
- Office light : 320 → 500 Lux
- Light while sunrise or sunset : 400 Lux
- TV studio light : 1,000 Lux
- Sunny Day (Indirect) : 10,000 → 25,000 Lux
- Sunny Day (Direct) : 32,000 → 130,000 Lux



EX(1):

```
#define LED 13
#define LDR 2
void setup() {
    pinMode(LED, OUTPUT);
    pinMode(LDR, INPUT);
}
void loop() {
    int value = digitalRead(LDR);
    digitalWrite(LED, value);
}
```

Value of Resistor (R1) decides that Led becomes HIGH or LOW



# Light Sensors : LDR

Intensity (Lux)	OP (Volt)
1	0.04 (OFF)
10	0.27 (OFF)
50	0.92 (OFF)
100	1.45 (OFF)
200	2.13 (OFF)
294	2.50 (ON)
400	2.78 (ON)
200	2.13 (ON)
40	0.8 (OFF)



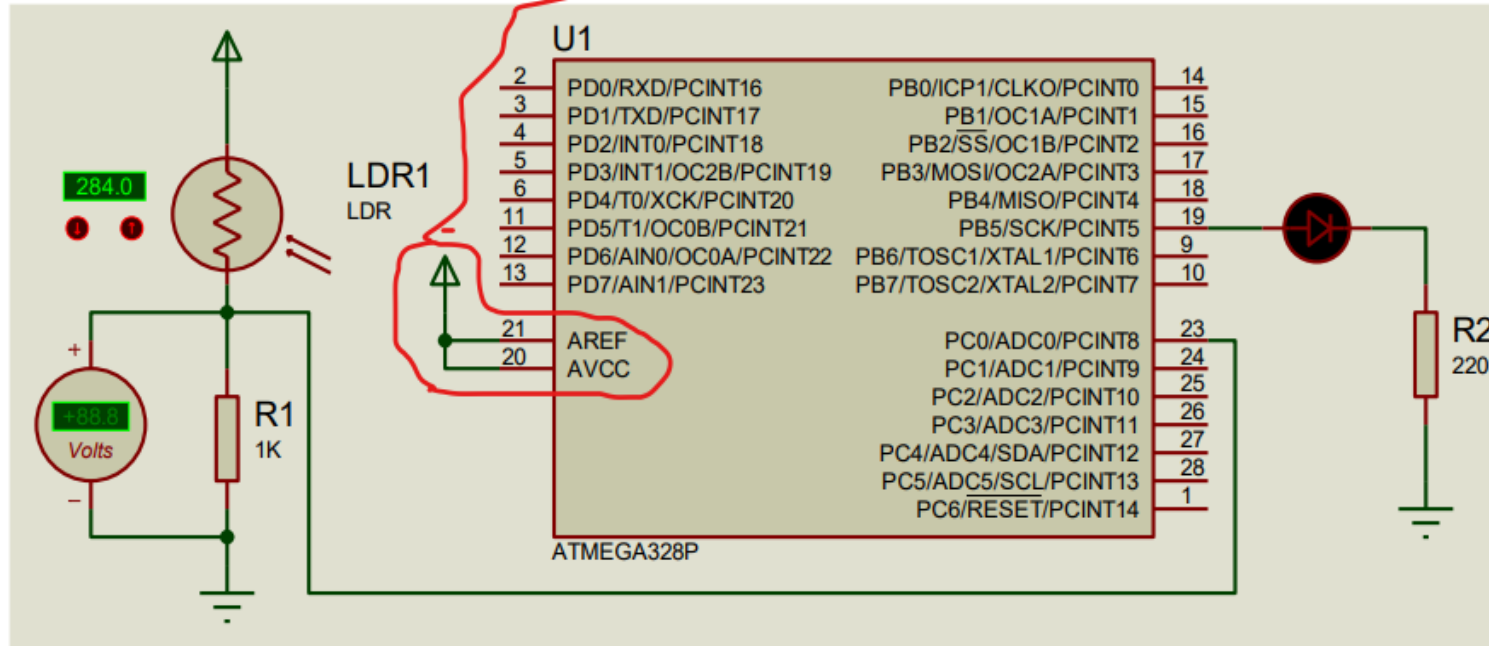
EX(2):

```
#define LED 13
void setup() {
    pinMode(LED, OUTPUT);
    analogReference(EXTERNAL);
}
int value;
void loop(){
    value = analogRead(0);
    digitalWrite(LED, (value>512)?HIGH:LOW);
}
```

we put an external reference Voltage

we read the intensity as a value and compare it internally with the Vref.

Intensity (Lux)	OP (Volt)
200	2.13 (OFF)
<b>294</b>	<b>2.50 (ON)</b>
400	2.78 (ON)
<b>293</b>	<b>2.49 (OFF)</b>
200	2.13 (OFF)



# **Motion Detection Sensors:**(بتحس بالحركة)

**PIR: Passive Infra-Red**

**PING**

**XL MaxSonar EZ1**

**IR: InfraRed**



## PIR:

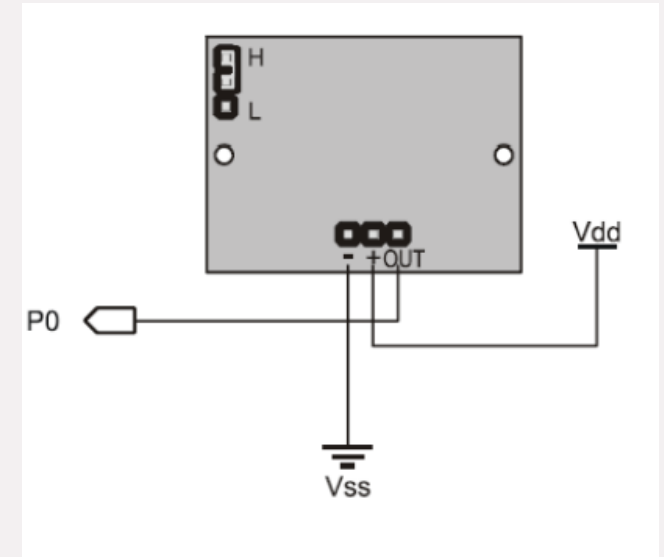
لما بيحصل حواليه حركة بيدي ON

### Model

- PIR#: 555-28027
- 1ELB106C5M

### Work Idea

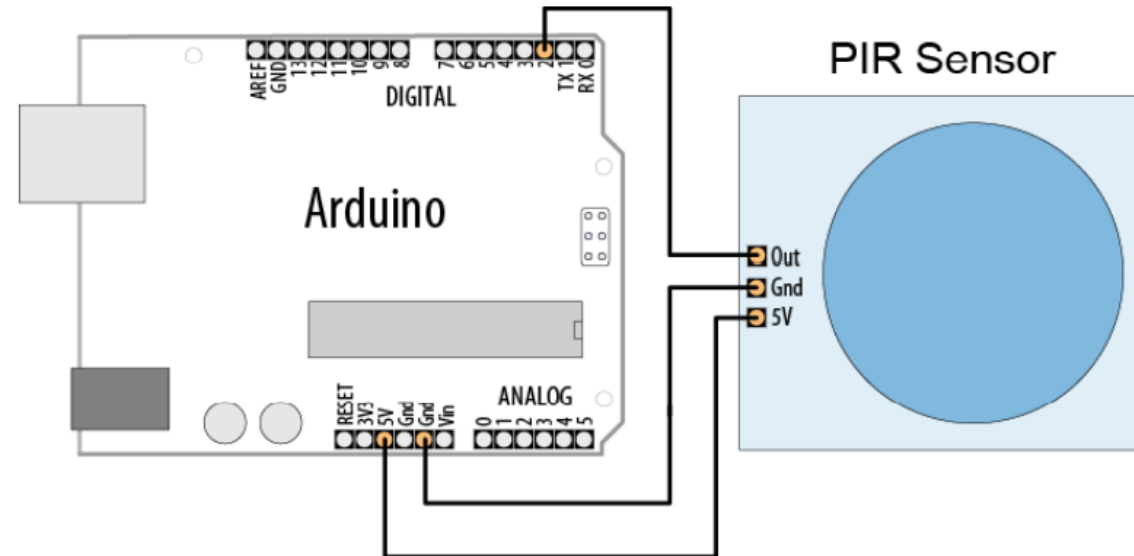
- After powering it detect any motion 7m (20ft) around sensor
- If any motion detected 5V is supplied to the OUT PIN



Ex:

```
#define LED 13
#define PIR 2
void setup() {
    pinMode(LED, OUTPUT);
    pinMode(PIR, INPUT);
}
void loop() {
    int value = digitalRead(PIR);
    if (value == HIGH)
    {
        digitalWrite(LED, HIGH);
        delay(50);
        digitalWrite(LED, LOW);
        delay(50);
    }
}
```

# Motion Detection Sensor : PIR



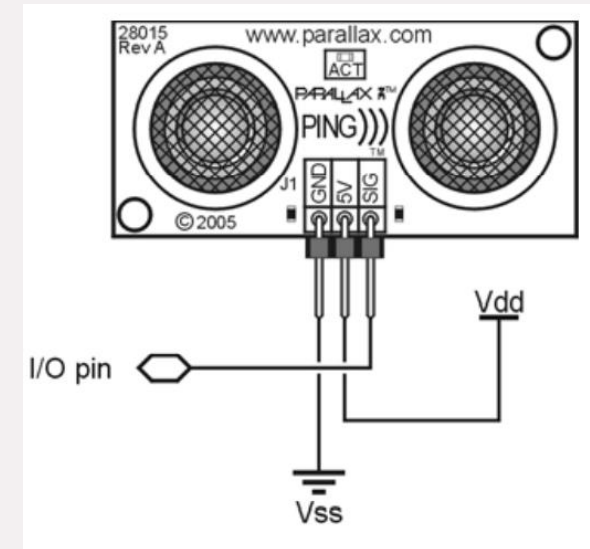
## PING:

Measure distance of moving object up to 3m , It uses ultrasound

بيعتد علي الultrasound بيبت اشارة والاشارة دي تخبط فحاجة وترجعه وبيقيس الوقت في الفترة دي ويترجمه لمسافة

Send clear (2ms→5ms) pulse to the SIG port  
(distance=delay/58) to calculate the distance in cm  
where delay in us

The speed of sound in air is ~ 29 cm/us



Ex:

```
const int pingPin = 5;
const int ledPin = 13;
long microsecondsToCentimeters(long microseconds) {
    return microseconds / 29 / 2;
}

int ping(int pingPin) {
    long duration;
    //Generate Pulse to trigger the PING device
    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);

    //Wait for the back pulse from PING device
    pinMode(pingPin, INPUT);
    duration = pulseIn(pingPin, HIGH);

    return microsecondsToCentimeters(duration);
}

void setup() {
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
}

void loop() {
    int cm = ping(pingPin);
    Serial.println(cm);
    digitalWrite(ledPin, HIGH);
    delay(cm * 10);
    digitalWrite(ledPin, LOW);
    delay(cm * 10);
}
```

when the distance decreases  
the led will turn on and off faster

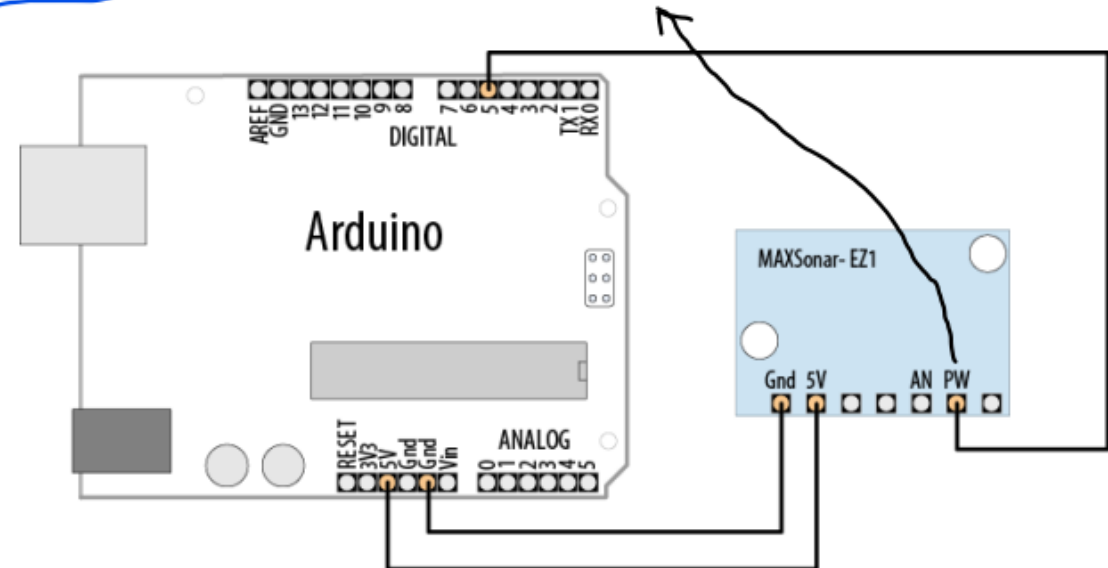
# Motion Detection Sensor : PING

this function take the time in us  
and return it as a distance in cm

send the pulse signal as a time and  
measure it in cm

function wait until it receives HIGH  
to return the duration

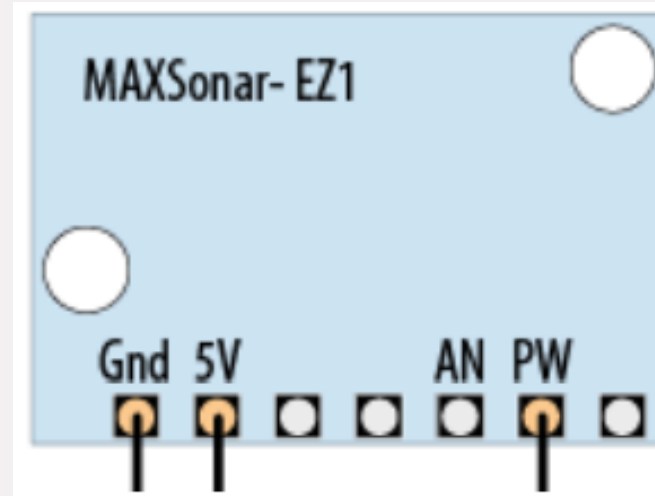
used to send trigger and take the answer



## XL MaxSonar EZ1

Measure distance of moving object up to (25 ft) 7.5m using the ultrasound

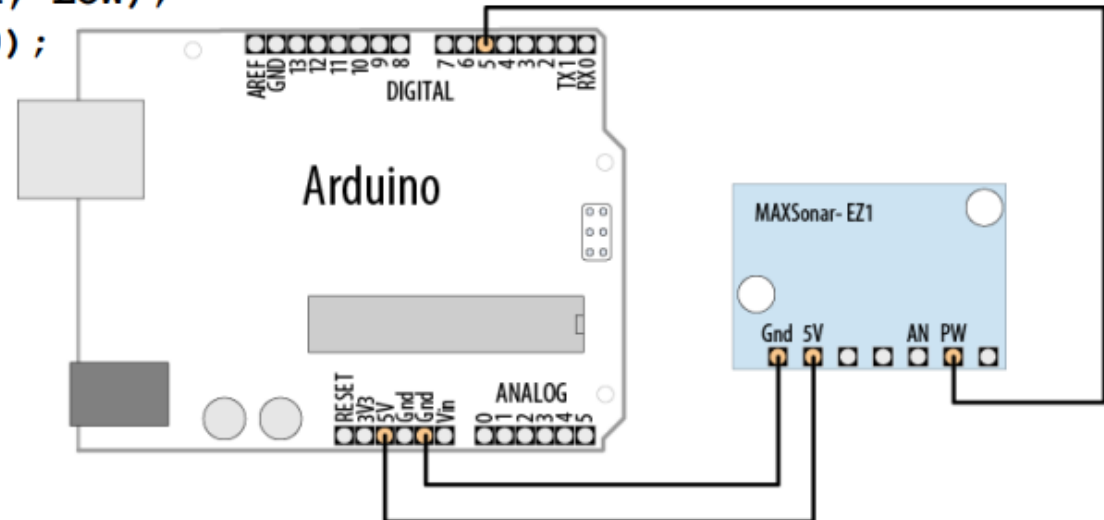
- This device is more simple since it can send continuous pulses without a trigger.
- The pulses is sent from PW PIN.
- Divide the pulse duration(us) by 58 to get the distance in cm



EX:

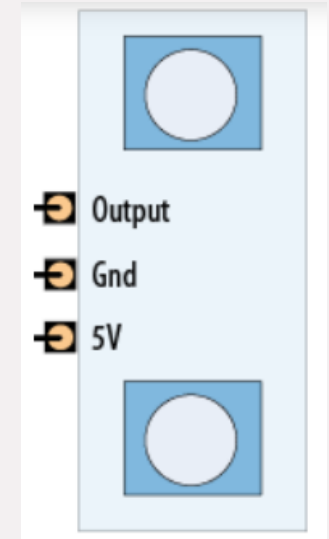
```
const int sensorPin = 5;  
const int ledPin = 13;  
void setup()  
{  
    Serial.begin(9600);  
    pinMode(ledPin, OUTPUT);  
}  
void loop()  
{  
    int duration = pulseIn(sensorPin, HIGH) ;  
    int distance = duration / 58;  
    Serial.println(distance);  
    digitalWrite(ledPin, HIGH);  
    delay(distance * 10);  
    digitalWrite(ledPin, LOW);  
    delay(distance * 10);  
    delay(20);  
}
```

## Other Motion Detection Sensor



## IR:

- Measure distance of moving object between 15 to 150cm.
- This device uses Infrared signal.

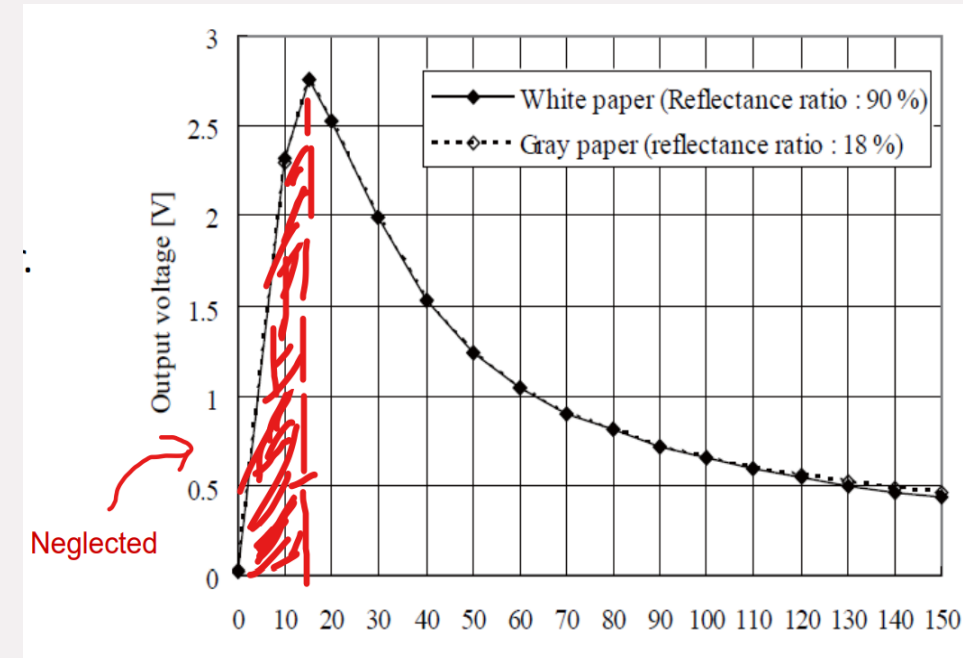


- The device provide analog signal in the Output PIN.

- The relation between the output voltage and the distance is nonlinear.

- Lookup table or graph can be used for conversion.

بهمل الجزء الي علي الشمال في الجراف عشان ميحصلش لغبطة





# EX:

```
const int ledPin = 13;  
const int sensorPin = 0; XValue = volt variable in the code XStart = defined as 250 in the code  
int lookup(int XValue, int XStart, int XStep, int YValues[], int nYValues){
```

```
    if(XValue > (XStart + XStep*(nYValues-1)))  
        return YValues[nYValues-1];  
    if(XValue < XStart)  
        return YValues[0];
```

XStep = the difference between every point and equals 250 in code

YValues[ ] = Array of distance values (dValues)

nYvalues = number of elements in the array

```
    int index = (XValue - XStart) / XStep;  
    float fraction = (XValue - index * XStep - XStart) / float(XStep);  
    return (int)(YValues[index] - fraction * (YValues[index] - YValues[index+1]));
```

If XValue exceeds the of value of max. value then return the Last value of the array  
and If XValue less than the start value then return the first value of the array

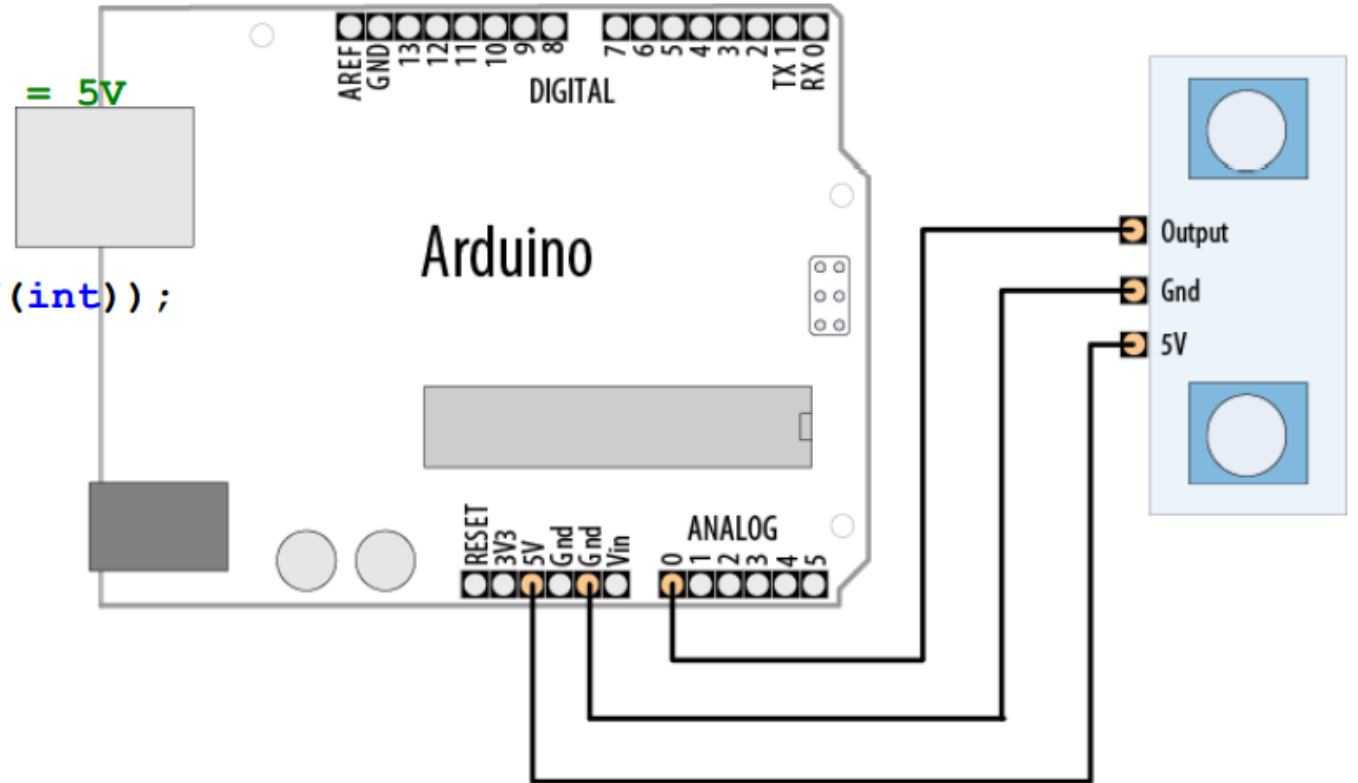
interpolation , we can use map function here.

```

void setup() {
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
}

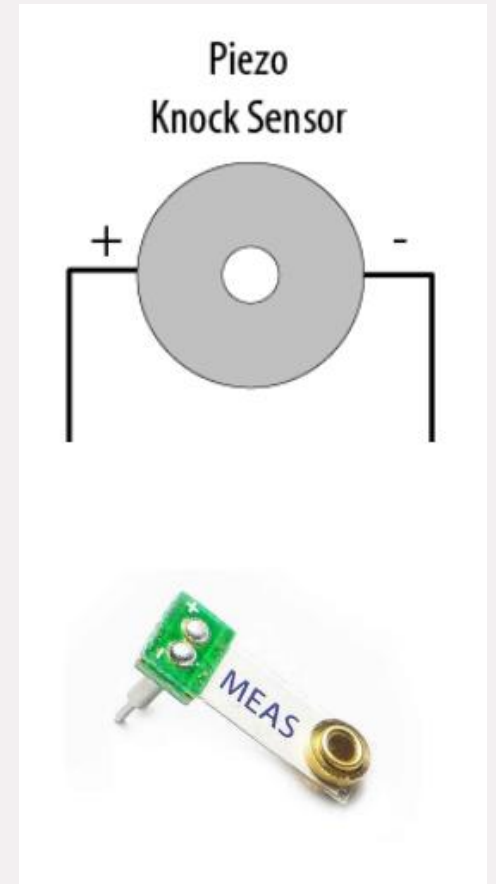
static int dValues[] = {150,140,130,100,60,50,40,35,30,25,20,15}; Array of distance values
const int start = 250, step = 250;
long value; int volt, distance;
void loop() {
    value = analogRead(sensorPin);
    volt = (value * 5000) / 1023; //5000 = 5V
    Serial.print(volt);
    Serial.print(",");
    distance = lookup(volt, start, step,
        dValues, sizeof(dValues)/sizeof(int));
    Serial.println(distance);
    digitalWrite(ledPin, HIGH);
    delay(distance * 1);
    digitalWrite(ledPin, LOW);
    delay(distance * 1);
    delay(100);
}

```



## Vibration Detection Sensors:

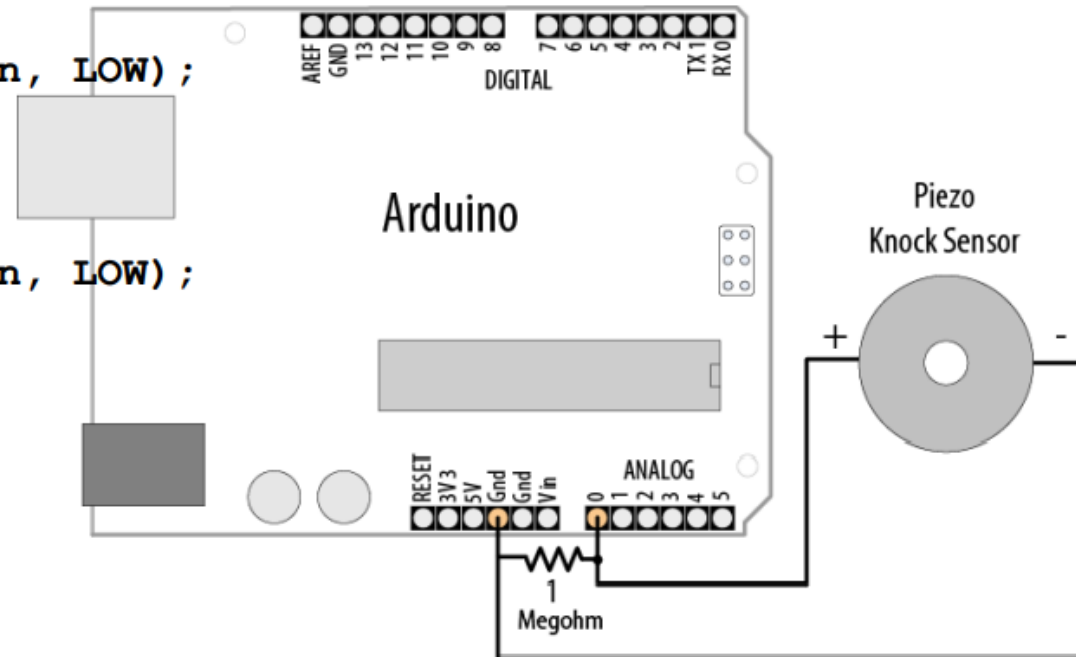
- it detect vibrations
- Uses Piezoelectric material. It produces voltage output while vibration proportional to vibration Strength



# EX:

```
const int sensorPin = 0;
const int ledPin = 13;
const int THRESHOLD = 100;
void setup()
{
    pinMode(ledPin, OUTPUT);
}
void loop()
{
    int val = analogRead(sensorPin);
    if (val >= THRESHOLD)
    {
        digitalWrite(ledPin, HIGH);
        delay(100);
        digitalWrite(ledPin, LOW);
        delay(100);
    }
    else
        digitalWrite(ledPin, LOW);
}
```

if the value that comes from  
`analogRead > THRESHOLD` it will give HIGH

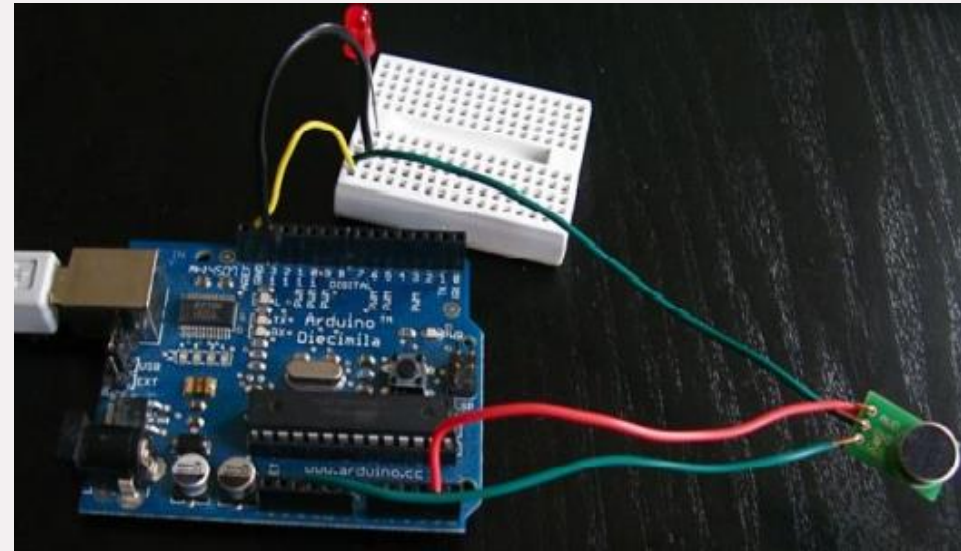


## Audio Sensors:

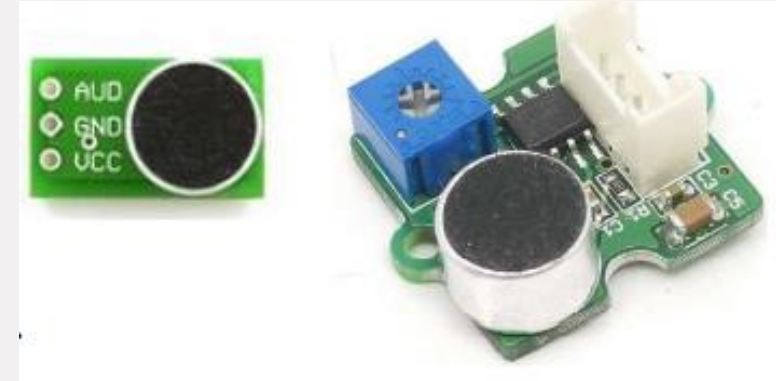
**MIC :Microphone (Used to record Waves).**

**Covert audio signal into a vibration which affect the internal resistance of the device.**

**- Connect the device as shown**



**-The output PIN generate an oscillatory signal (audio wave).  
As the audio signal go high as the average output increases.**



# EX:

```
const int ledPin = 13;
const int middleValue = 512;
int samples[128];
int nSamples = sizeof(samples)/sizeof(int);
int index = 0;
const int threshold = 400;
long average;

void setup() {
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);
    for(int i=0;i<nSamples;i++) samples[i] = 0;
}

void loop() {
    samples[index] = analogRead(0);
    index = (index+1)%nSamples;

    average = 0;
    for(int i=0;i<nSamples;i++) average += samples[i];
    average /= nSamples;

    if(average>512)
        digitalWrite(ledPin, HIGH);
    else
        digitalWrite(ledPin, LOW);

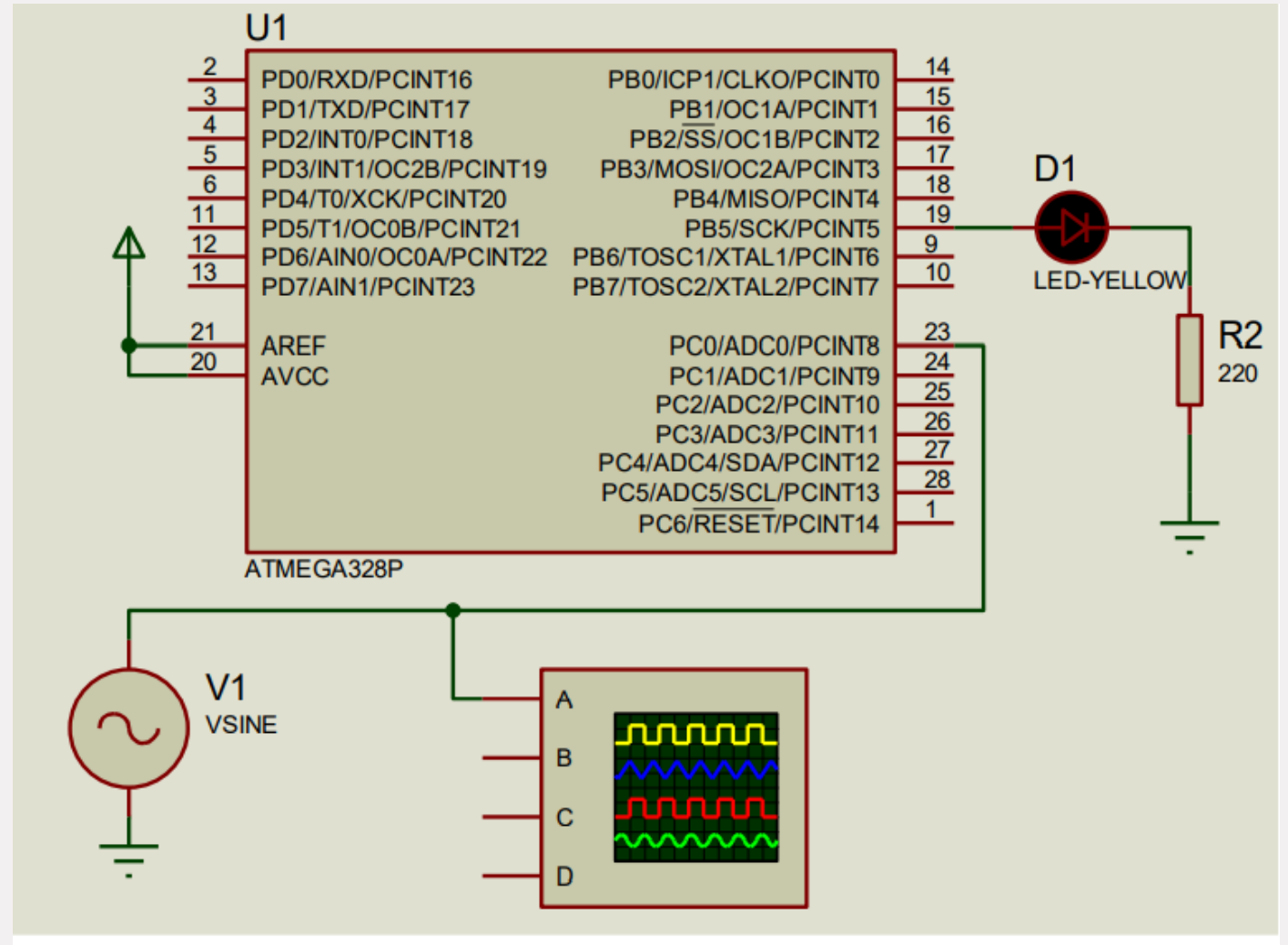
    Serial.println(average);
}
```

we make this step to not exceed 128 number in the array. so we guaranteed that the numbers of index will be from 0 to 127.

sound is a wave that has many values so we calculate the average to determine if the sound is high or not

Additional: we should add here delay function to notice the Answer

## Circuit of the last Ex:

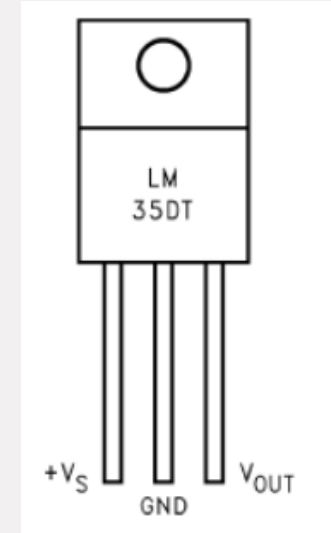




## Temperature Sensors:

# LM35

- Depend on semiconductor material
- The resistance of the material changes with temperature Produces linear output proportional to the temperature (Celsius).
- Temperature range (-55 to 150 Celsius)



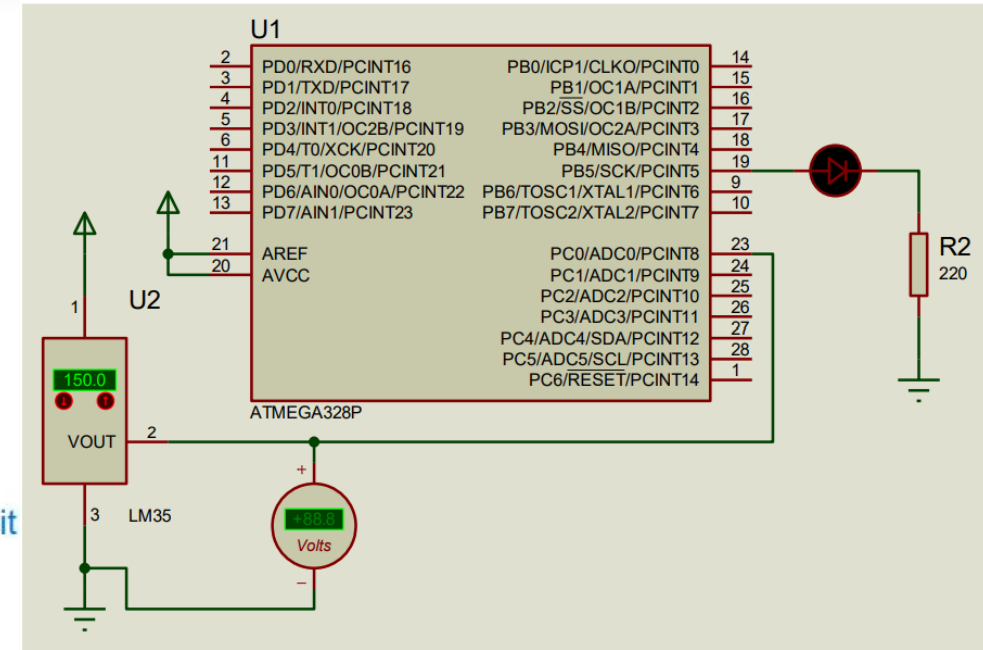
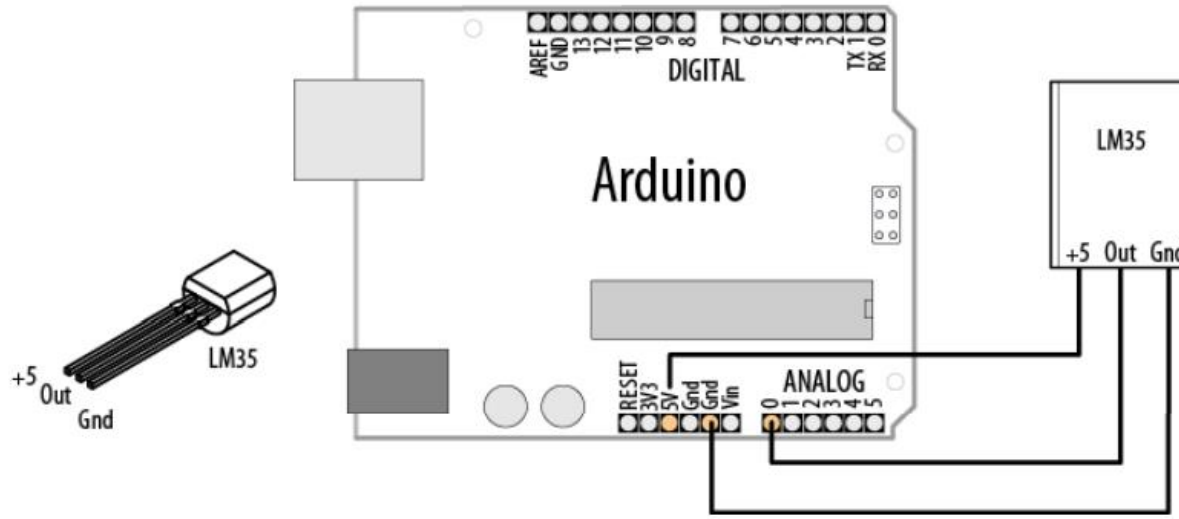
# Ex:

```
const int inPin = 0;
void setup()
{
    Serial.begin(9600);
}
void loop()
{
    int value = analogRead(inPin);
    Serial.print(value); Serial.print(" > ");
    float millivolts = (value / 1024.0) * 5000;
    float celsius = millivolts / 10;
    Serial.print(celsius);
    Serial.print(" degrees Celsius, ");
    Serial.print( (celsius * 9) / 5 + 32 );
    Serial.println(" degrees Fahrenheit");
    delay(1000);
}
```

Convert Value to mV

Convert millivolts to Celsius

Convert from Celsius to Fahrenheit



## **Identification Sensors:**

**RFID: Radio Frequency Identification (Smart Sensor).**

**Read RFID tags and produce serial signal containing the tag information(Such as Underground Metro Cards).**



## **Software Serial Library:**

**Library allows developer to communicate with serial devices not connected to the PINs(0,1) , a new Serial pins**

# EX:

```
#include <NewSoftSerial.h>

NewSoftSerial RFID(2, 3);

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

void loop() {
    String msg;
    if(RFID.available())
    {
        while(RFID.available()>0)
            msg += (char)RFID.read();
        Serial.println(msg);
    }
}
```

