# DHT11 Sensor and Its Working

Humidity is the measure of water vapour present in the air. The level of humidity in air affects various physical, chemical and biological processes.  In industrial applications, humidity can affect the business cost of the products, health and safety of the employees. So, in [semiconductor](https://www.elprocus.com/why-do-we-use-semiconductors-instead-of-conductors-in-electronic-circuit-design/) industries and control system industries measurement of humidity is very important.  Humidity measurement determines the amount of moisture present in the gas that can be a mixture of water vapour, nitrogen, argon or pure gas etc…  Humidity sensors are of two types based on their measurement units.  They are a relative humidity sensor and Absolute humidity sensor. DHT11 is a digital temperature and humidity sensor.

## ****What is a DHT11 Sensor?****

DHT11 is a low-cost digital sensor for sensing temperature and humidity.  This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc… to measure humidity and temperature instantaneously.

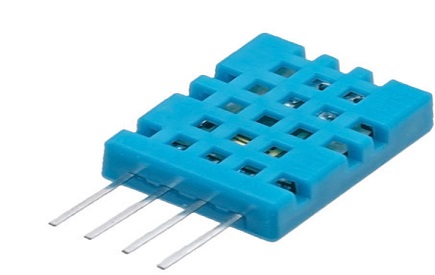
DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor.  To measure the surrounding air this sensor uses a [thermistor](https://www.elprocus.com/introduction-to-thermistor-types-with-its-workings-and-applications/" \t "_blank) and a capacitive humidity sensor.

### ****Working Principle of DHT11 Sensor****

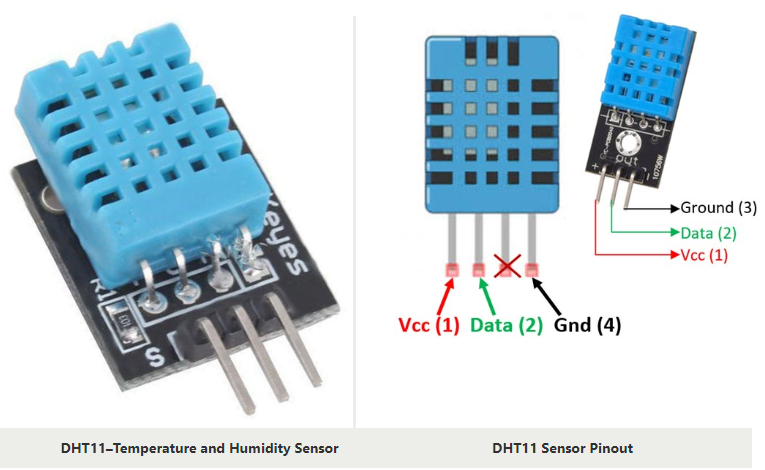
DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature.  The humidity sensing [capacitor](https://www.elprocus.com/construction-of-capacitor-with-working/) has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second.  DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

DHT11 Sensor

DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

****

**CODE:**

#include <dht.h>

#define dht\_apin A0 // Analog Pin sensor is connected to

dht DHT;

void setup(){

Serial.begin(9600);

delay(500);//Delay to let system boot

Serial.println("DHT11 Humidity & temperature Sensor\n\n");

delay(1000);//Wait before accessing Sensor

}//end "setup()"

void loop(){

//Start of Program

DHT.read11(dht\_apin);

Serial.print("Current humidity = ");

Serial.print(DHT.humidity);

Serial.print("% ");

Serial.print("temperature = ");

Serial.print(DHT.temperature);

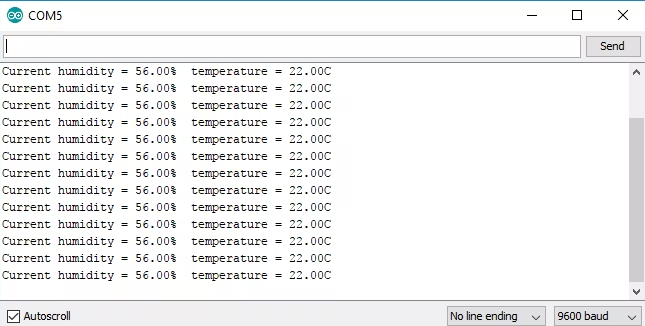
Serial.println("C ");

delay(5000);//Wait 5 seconds before accessing sensor again.

//Fastest should be once every two seconds.

}// end loop(

Output:



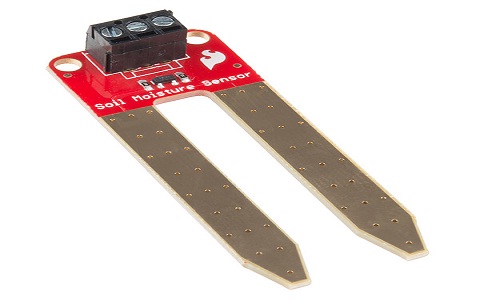
# Soil Moisture Sensor

The moisture of the soil plays an essential role in [the irrigation field](https://www.elprocus.com/microcontroller-based-automatic-irrigation-system/) as well as in gardens for plants. As nutrients in the soil provide the food to the plants for their growth. Supplying water to the plants is also essential to change the temperature of the plants. The temperature of the plant can be changed with water using the method like transpiration. And plant root systems are also developed better when rising within moist soil. Extreme soil moisture levels can guide to anaerobic situations that can encourage the plant’s growth as well as soil pathogens. This article discusses an overview of the soil moisture sensor, working and it’s applications.

## What is a Soil Moisture Sensor?

The soil moisture sensor is one [kind of sensor](https://www.elprocus.com/accelerometer-sensor-working-and-applications/) used to gauge the volumetric content of water within the soil. As the straight gravimetric dimension of soil moisture needs eliminating, drying, as well as sample weighting. These sensors measure the volumetric water content not directly with the help of some other rules of soil like dielectric constant, electrical resistance, otherwise interaction with neutrons, and replacement of the moisture content.

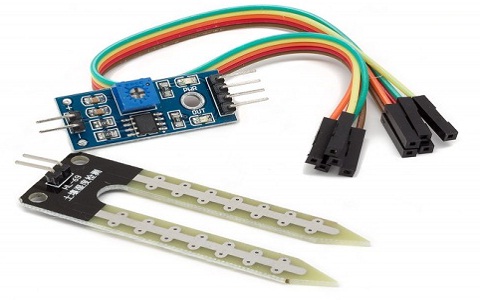
The relation among the calculated property as well as moisture of soil should be adjusted & may change based on ecological factors like temperature, type of soil, otherwise electric conductivity. The microwave emission which is reflected can be influenced by the moisture of soil as well as mainly used in agriculture and remote sensing within hydrology.

soil-moisture-sensor-device

These [sensors](https://www.elprocus.com/types-of-sensors-with-circuits/) normally used to check volumetric water content, and another group of sensors calculates a new property of moisture within soils named water potential. Generally, these sensors are named as soil water potential sensors which include gypsum blocks and tensiometer.

### Soil Moisture Sensor Pin Configuration

The FC-28 soil moisture sensor includes 4-pins

soil-moisture-sensor

* VCC pin is used for power
* A0 pin is an analog output
* D0 pin is a digital output
* GND pin is a Ground

This module also includes a potentiometer that will fix the threshold value, & the value can be evaluated by the [comparator-LM393](https://www.elprocus.com/lm393-ic-pin-configuration-circuit-diagram-and-its-working/). The [LED](https://www.elprocus.com/bipolar-led-driver-circuit-working-application/) will turn on/off based on the threshold value.

How the Sensor Works

The Soil Moisture Sensor uses capacitance to measure dielectric permittivity of the surrounding medium. In soil, dielectric permittivity is a function of the water content. The sensor creates a voltage proportional to the dielectric permittivity, and therefore the water content of the soil.

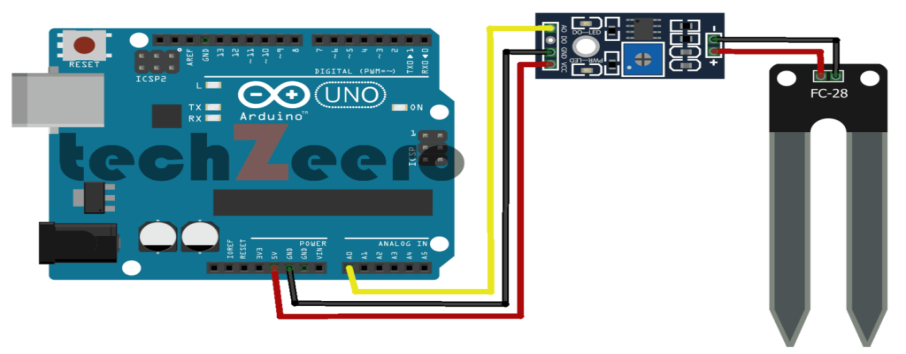
The sensor averages the water content over the entire length of the sensor. There is a 2 cm zone of influence with respect to the flat surface of the sensor, but it has little or no sensitivity at the extreme edges. The figure above shows the electromagnetic field lines along a cross-section of the sensor, illustrating the 2 cm zone of influence.

## Working Principle

This sensor mainly utilizes capacitance to gauge the water content of the soil (dielectric permittivity). The working of this sensor can be done by inserting this sensor into the earth and the status of the water content in the soil can be reported in the form of a percent.

This sensor makes it perfect to execute experiments within science courses like environmental science, agricultural science, biology, soil science, botany, and horticulture.

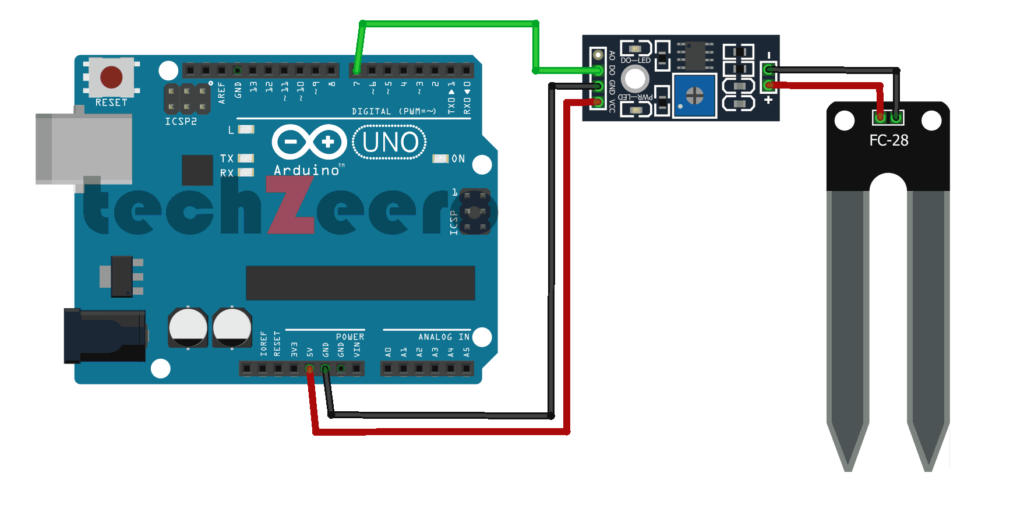
**Circuit Diagram**

[[](https://techzeero.com/wp-content/uploads/2019/12/circuit-soil-moisture-with-arduino-analog-1024x501.png)](https://techzeero.com/wp-content/uploads/2019/12/circuit-soil-moisture-with-arduino-analog-1024x501.png)

**Code – Analog Output**

|  |
| --- |
| Soil Moisture with Arduino - Analog Output |
| For more details, visit: https://techzeero.com/arduino-tutorials/soil-moisture-sensor-arduino/ |
| \*/ |
|  |
| int sensorPin = A0; |
| int outputValue ; |
|  |
| void setup() |
| { |
| Serial.begin(9600); |
| Serial.println("Reading Data From the Sensor ..."); |
| delay(2000); |
| } |
|  |
| void loop() |
| { |
| outputValue= analogRead(sensorPin); |
| outputValue = map(outputValue,550,0,0,100); |
|  |
| Serial.print("Moisture Value : "); |
| Serial.print(outputValue); |
| Serial.println("%"); |
| delay(1000); |
| } |

**Circuit Diagram**

[[](https://techzeero.com/wp-content/uploads/2019/12/circuit-soil-moisture-with-arduino-digital-1024x512.png)](https://techzeero.com/wp-content/uploads/2019/12/circuit-soil-moisture-with-arduino-digital-1024x512.png)

**Code – Digital Output**

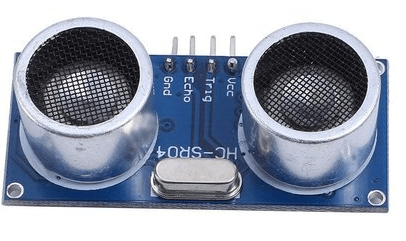
Upload the code to the Arduino board. When the value of the sensor is high, the inbuild led will on.

|  |
| --- |
| /\* |
| Soil Moisture with Arduino - Digital Output |
| For more details, visit: https://techzeero.com/arduino-tutorials/soil-moisture-sensor-arduino/ |
| \*/ |
|  |
| int sensorPin = 7; |
| int ledPin = 13; |
|  |
| void setup() |
| { |
| pinMode(ledPin, OUTPUT); |
| pinMode(sensorPin, INPUT); |
| Serial.begin(9600); |
| Serial.println("Reading Data From the Sensor ..."); |
| delay(2000); |
| } |
|  |
| void loop() |
| { |
| if(digitalRead(sensorPin) == HIGH) |
| { |
| digitalWrite(ledPin, HIGH); |
| } |
| else |
| { |
| digitalWrite(ledPin, LOW); |
| delay(1000); |
| } |
| } |

**Ultrasonic Sensor**

This section explains the interfacing of the ultrasonic sensor with an Arduino by considering HC-SR-04 where it explains the ultrasonic sensor pinout, its specifications, wiring diagram, and how the sensor with Arduino connection.

The ultrasonic sensor pin diagram is:



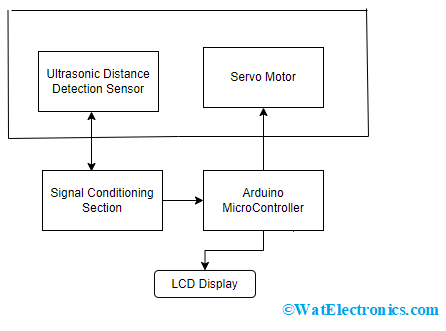
**Vcc** – This pin has to be connected to a power supply +5V.

**TRIG** – This pin is used to receive controlling signals from the Arduino board. This is the triggering input pin of the sensor

**ECHO** – This pin is used for sending signals to the Arduino board where the Arduino calculates the pulse duration to know the distance. This pin is the ECHO output of the sensor.

**GND** – This pin has to be connected to the ground.

The below picture shows the **ultrasonic sensor block diagram** for distance measurement.



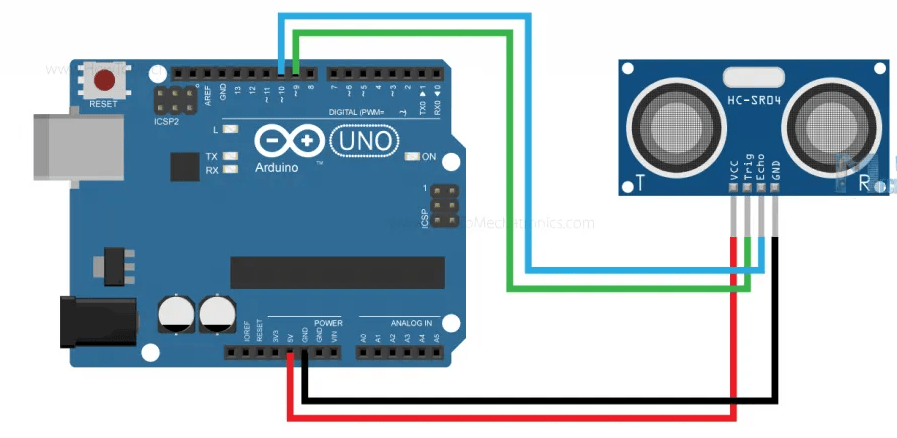
**Ultrasonic Sensor Block Diagram**

The target’s distance is calculated using an ultrasonic distance sensor and the output from the sensor is provided to the signal conditioning section and then is processed using an Arduino [microcontroller](https://www.watelectronics.com/attiny85-microcontroller/). The results from the microcontroller are fed to the LCD display and then moved to a personal computer.

The ultrasonic sensor can be connected to the servo motor to know the polar distance of the sensor up to 1800 rotations approximately.

#### Working

In general, an ultrasonic sensor has two sections which are the transmitter and receiver. These sections are closely placed so that the sound travel in a straight line from the transmitter to the target and travels back to the receiver. Making sure to have minimal distance between transmitter and receiver section delivers minimal errors while calculations.



[Integration of Ultrasonic Transducer with Arduino](https://howtomechatronics.com/)

These devices are also termed ultrasonic transceivers because both the transmitter and receiver sections are combined in a single unit which considerably minimizes the PCB footprint.

Here, the sensor operates as a burst signal and it is transmitted for some period. Later the transmission, there exists a **silent period** and this period is termed **response time**. The response time indicates that it is waiting for the reflected waves.

The shape of the acoustic waves that leave the transmitter section resembles the same shape of the light emitted from a laser so beam angle and spread have to be measured. When the sound waves move away from the transmitter, the detection area increases vertically and sideways too. Because of the varying detection area, the coverage specification is considered either as beam angle/beamwidth other than the standard area of detection.

It is more recommended to observe the beam angle pattern for the sensor whether it is the complete angle of the beam or the angle of variation corresponding to the straight line that forms a [transducer](https://www.watelectronics.com/ultrasonic-transducer/). Mostly, a thin beam angle results in a higher detection range, and a broader beam angle corresponds to a lesser detection range.

The transmitted/acoustic signals might find a hindrance or not. When there is any hindrance, the acoustic wave bounces back from the hindrance. This bounced signal is termed ECHO. This echo travels to the receiver.

Then the received signal is either filtered or amplified and then transformed into a digital signal. With the time between transmission and reception of acoustic waves, the distance between the ultrasonic system and hindrance can be known.

#### Code to Measure Distance

//defining pin numbers

int trig = 9; // trigger pin connected to 9th pin in Arduino board

int echo = 8; // echo pin connected to 10th pin in Arduino board

// defining variables

long timetaken;

int distance;

void setup() {

pinMode (trig, OUTPUT); // sets the trigger pin as output mode

pinMode(echo, INPUT); // sets the echo pin as input mode

// initiating the serial communication

Serial.begin(9600);

}

Void loop () {

digitalWrite (trig, LOW); // clearing the trigger pin

delayMS (2);

digitalWrite (trig, HIGH); // sets the trigger pin to HIGH state for 10 µsec

delayMS (10);

digitalWrite (trig, LOW);

timetaken = pulseIN(echo, HIGH); // calculates the time taken by pulse from echo pin

distance = timetaken \* 0.034/2; // measures the distance

serial.print (“Timetaken:….”) // prints the value on LCD display

serial.println(timetaken);

}

**CODE:**

#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04

#define trigPin 3 //attach pin D3 Arduino to pin Trig of HC-SR04

// defines variables

long duration; // variable for the duration of sound wave travel

int distance; // variable for the distance measurement

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT

pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT

Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed

Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor

Serial.println("with Arduino UNO R3");

}

void loop() {

// Clears the trigPin condition

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin HIGH (ACTIVE) for 10 microseconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance = duration \* 0.034 / 2; // Speed of sound wave divided by 2 (go and back)

// Displays the distance on the Serial Monitor

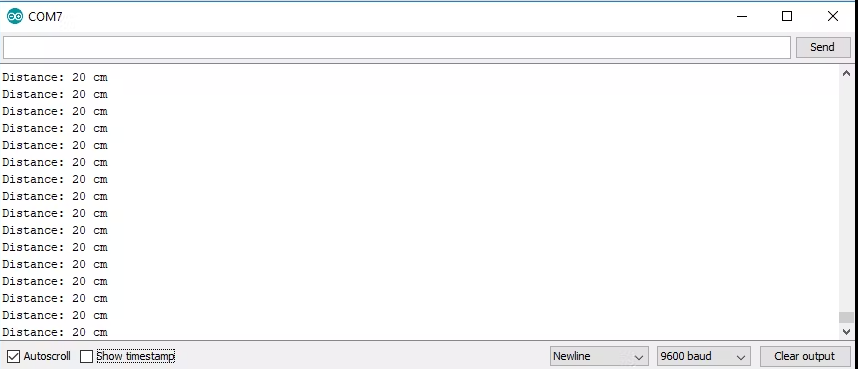
Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

}

OUTPUT:



### IR Sensor

An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measure only infrared radiation, rather than emitting it that is called a [passive IR sensor](https://www.elprocus.com/passive-infrared-pir-sensor-with-applications/). Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation.

Infrared Sensor

These types of radiations are invisible to our eyes, which can be detected by an infrared sensor. The emitter is simply an IR LED ([Light Emitting Diode](https://www.elprocus.com/explain-different-types-leds-working-applications-engineering-students/)) and the detector is simply an IR photodiode that is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received.

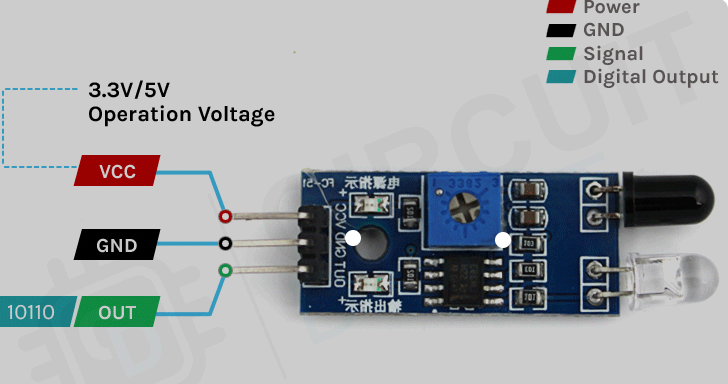
### Working Principle

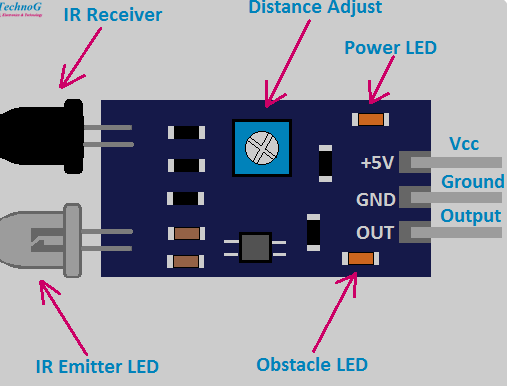
The working principle of an infrared sensor is similar to the object detection sensor. This sensor includes an IR LED & an IR Photodiode, so by combining these two can be formed as a photo-coupler otherwise optocoupler. The physics laws used in this sensor are planks radiation, Stephan Boltzmann & weins displacement.

IR LED is one kind of transmitter that emits IR radiations. This LED looks similar to a standard LED and the radiation which is generated by this is not visible to the human eye. Infrared receivers mainly detect the radiation using an infrared transmitter. These infrared receivers are available in photodiodes form. IR Photodiodes are dissimilar as compared with usual photodiodes because they detect simply IR radiation. Different kinds of infrared receivers mainly exist depending on the voltage, wavelength, package, etc.

Once it is used as the combination of an IR transmitter & receiver, then the receiver’s wavelength must equal the transmitter. Here, the transmitter is IR LED whereas the receiver is IR photodiode. The infrared photodiode is responsive to the infrared light that is generated through an infrared LED. The resistance of photo-diode & the change in output voltage is in proportion to the infrared light obtained. This is the IR sensor’s fundamental working principle.

Once the infrared transmitter generates emission, then it arrives at the object & some of the emission will reflect back toward the infrared receiver. The sensor output can be decided by the IR receiver depending on the intensity of the response.





**Code**

int IRSensor = 2; // connect ir sensor to arduino pin 2

int LED = 13; // conect Led to arduino pin 13

void setup()

{

pinMode (IRSensor, INPUT); // sensor pin INPUT

pinMode (LED, OUTPUT); // Led pin OUTPUT

}

void loop()

{

int statusSensor = digitalRead (IRSensor);

if (statusSensor == 1)

digitalWrite(LED, LOW); // LED LOW

}

else

{

digitalWrite(LED, HIGH); // LED High

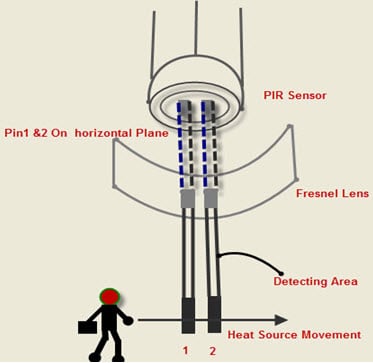
}

}

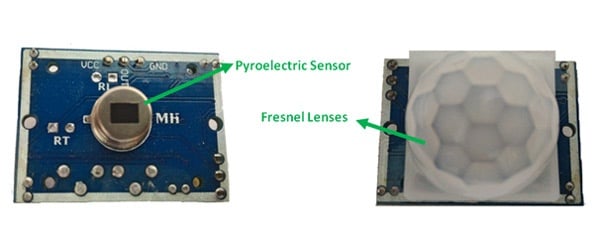
**PIR Sensor**

## PIR Sensor Working Principle

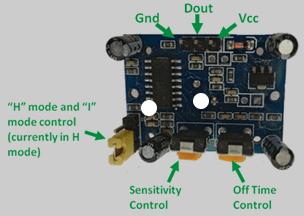
The passive infrared sensor does not radiate energy to space. It receives the  infrared radiation from the human body to make an alarm. Any object with temperature is constantly radiating infrared rays to the outside world. The surface temperature of the human body is between 36° C - 27 ° C and most of its radiant energy concentrated in the wavelength range of 8 um-12 um.

[](https://robu.in/wp-content/uploads/2020/05/9-10-2014-10-30-47-AM.jpg)

Passive infrared alarms classified into[**infrared detectors**](https://robu.in/product-category/sensor/ir-and-pir-sensor/) (infrared probes) and alarm control sections. The most widely used infrared detector is a pyroelectric detector. It uses as a sensor for converting human infrared radiation into electricity. If the human infrared radiation is directly irradiated on the detector, it will, of course, cause a temperature change to output a signal. But in doing all this, the detection distance will not be more. In order to lengthen the detection distance of the detector, an optical system  must be added to collect the infrared radiation. Usually, plastic optical reflection system or plastic **Fresnel lens** used as a focusing system for infrared radiation.

[](https://robu.in/wp-content/uploads/2020/05/main-qimg-80b474beb656c28373fbb8258a9968e9.jpg)

In the detection area, the lens of the detector receives the infrared radiation energy of the human body through the clothing and focused on the pyroelectric sensor. When the human body moves in this surveillance mode, it enters a certain field of view in sequence and then walks out of the field of view. The[**pyroelectric sensor**](https://www.sciencedirect.com/topics/engineering/pyroelectric-sensor) sees the moving human body for a while and then does not see it, so the infrared radiation of human body constantly changes the temperature of the pyroelectric material. So that it outputs a corresponding signal, which is the alarm signal.



**CODE:**

int led = 13; // the pin that the LED is atteched to

int sensor = 2; // the pin that the sensor is atteched to

int state = LOW; // by default, no motion detected

int val = 0; // variable to store the sensor status (value)

void setup() {

pinMode(led, OUTPUT); // initalize LED as an output

pinMode(sensor, INPUT); // initialize sensor as an input

Serial.begin(9600); // initialize serial

}

void loop(){

val = digitalRead(sensor); // read sensor value

if (val == HIGH) { // check if the sensor is HIGH

digitalWrite(led, HIGH); // turn LED ON

delay(500); // delay 100 milliseconds

if (state == LOW) {

Serial.println("Motion detected!");

state = HIGH; // update variable state to HIGH

}

}

else {

digitalWrite(led, LOW); // turn LED OFF

delay(500); // delay 200 milliseconds

if (state == HIGH){

Serial.println("Motion stopped!");

state = LOW; // update variable state to LOW

}

}

}

