WHAT ARE SENSORS AND WHY DO WE NEED THEM?

Definition could be like this:

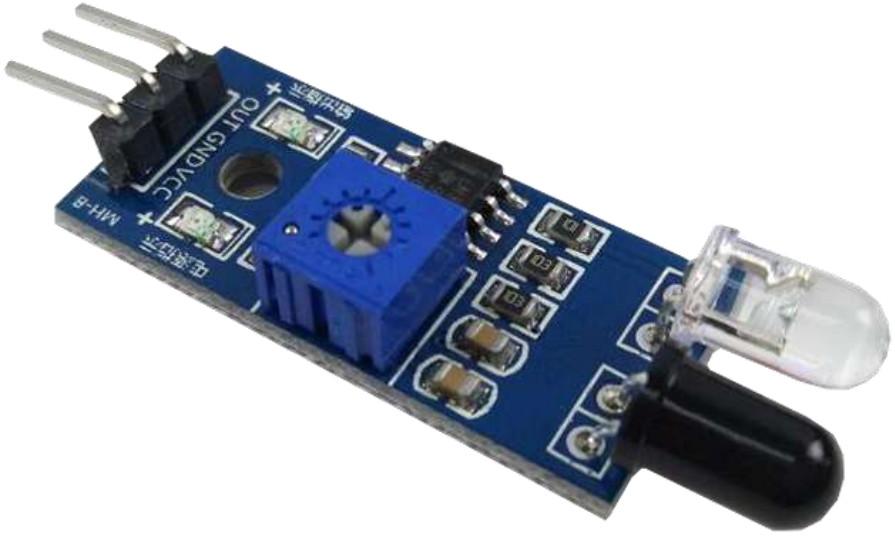
- A sensor is a device that detects and responds to some type of input from the physical environment.

Let us understand the working of few sensors in coming slides and their Usage..

IR sensor

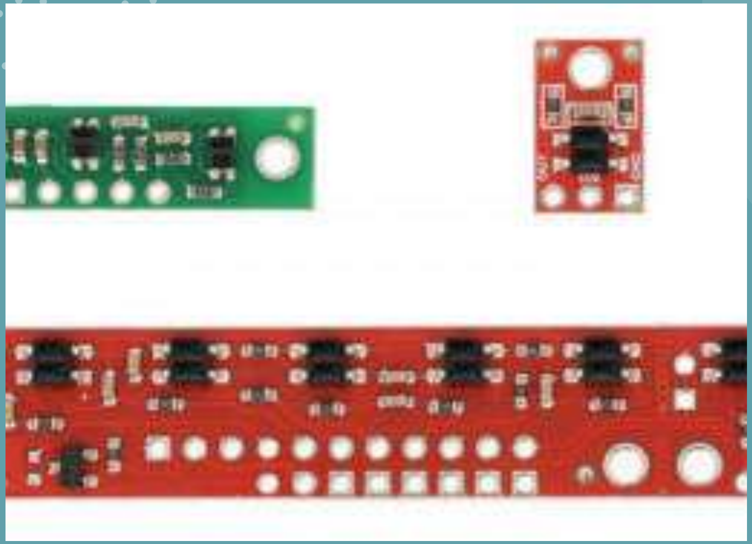
An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment.

- There are two types of IR sensors:
- Passive (popularly now as PIR)
- Active(IR sensor)



Active infrared sensor

- Active infrared sensors both emit and detect infrared radiation.
- When an object comes close to the sensor, the infrared light from the LED reflects off of the object and is detected by the receiver.
- Anything that emits heat (everything that has a temperature above around five degrees Kelvin) gives off infrared radiation.



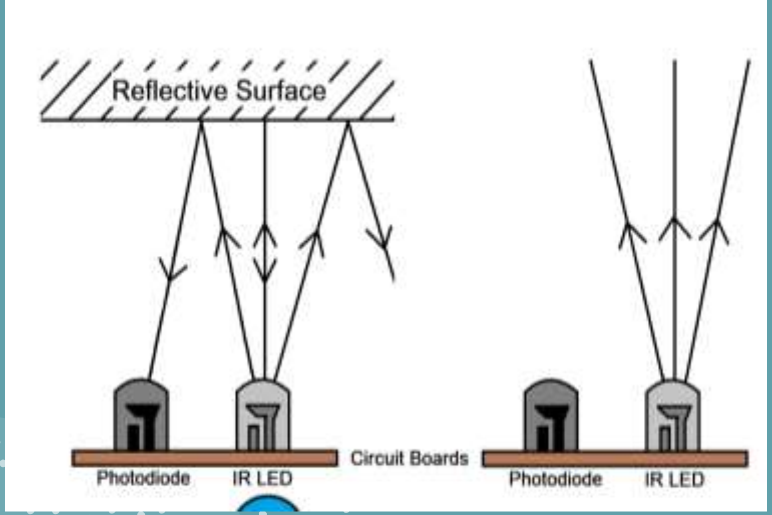
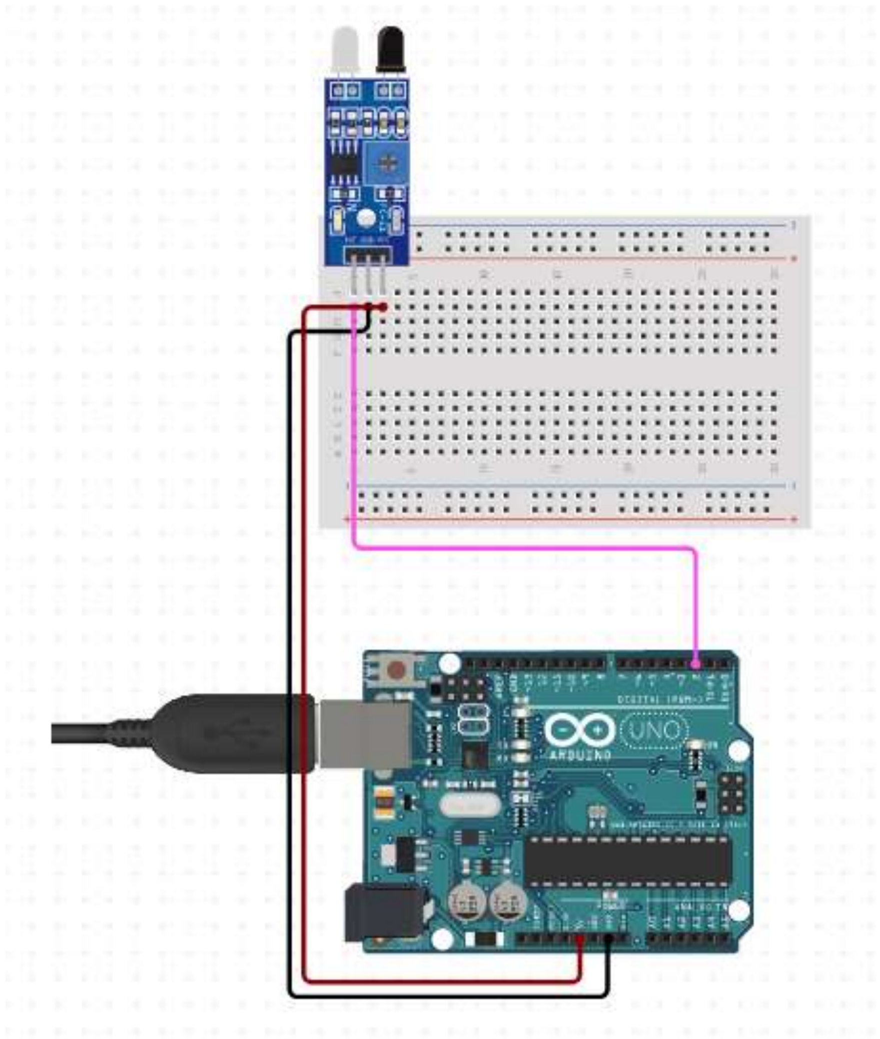


Fig:Array of ir (used for high accuracies ,generally used to make line followers)

Simple example:

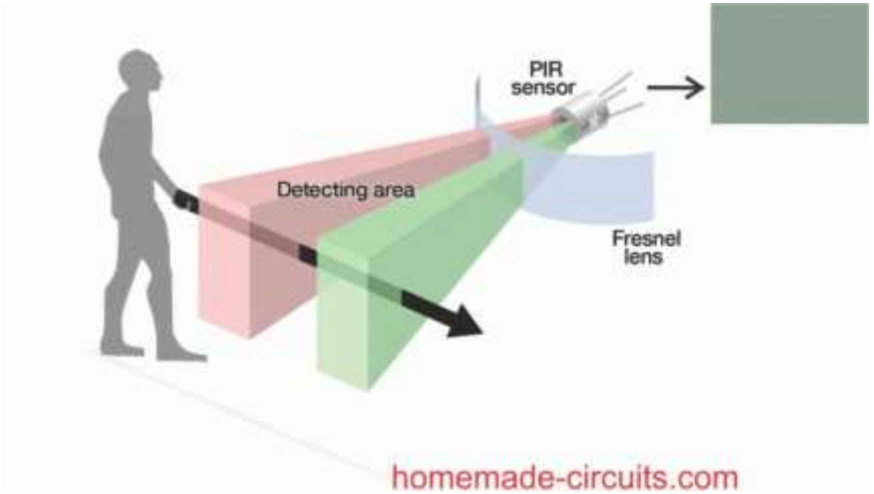
```
int IRSensor = 2; // connect ir sensor to arduino pin 2
int LED = 13; // conect Led to arduino pin 13

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
  }
}
```



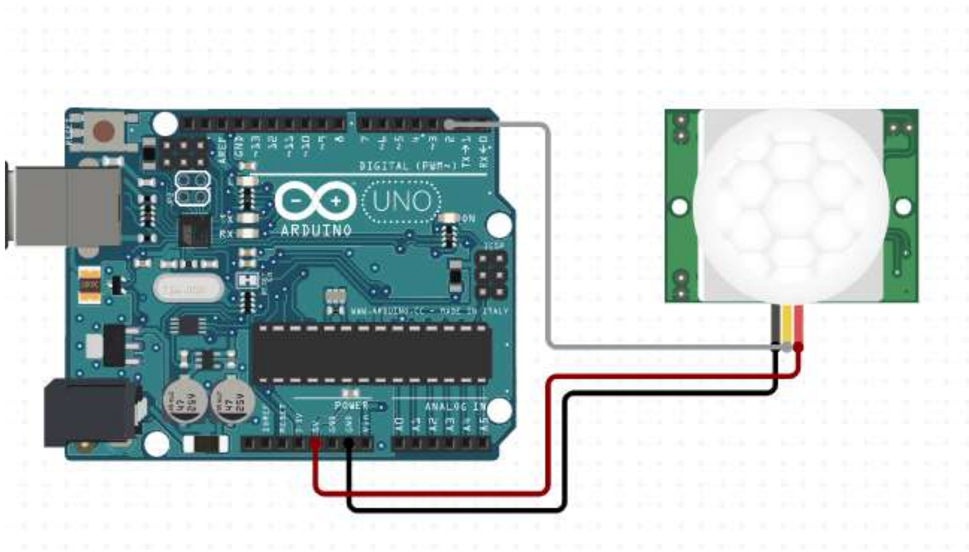
Passive Infrared Sensor

- Passive infrared (PIR) sensors only detect infrared radiation and do not emit it from an LED.
- PIR sensors are most commonly used in motion-based detection, such as in-home security systems.

- **Contains parts such as:**
  - strips of pyroelectric material (a pyroelectric sensor)
  - An infrared filter (that blocks out all other wavelengths of light)
  - A fresnel lens (which collects light from many angles into a single point)



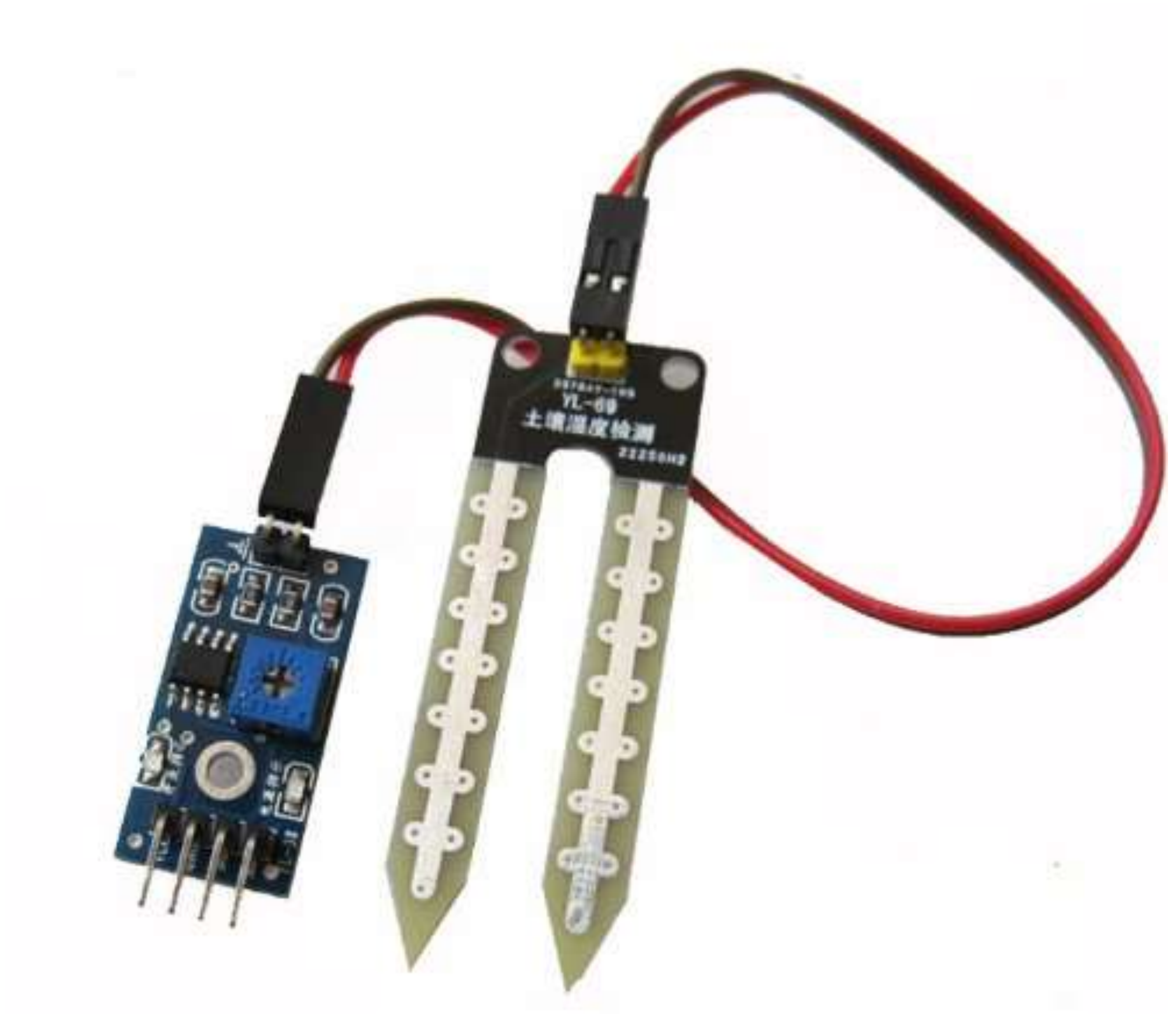
Simple example:



```
int ledPin = 13;   // choose the pin for the LED
int inputPin = 2;   // choose the input pin (for PIR sensor)
int pirState = LOW;   // we start, assuming no motion detected
int val = 0;   // variable for reading the pin status

void setup() {
pinMode(ledPin, OUTPUT);   // declare LED as output
pinMode(inputPin, INPUT);   // declare sensor as input

Serial.begin(9600);
}
```

contd

```
void loop(){
val = digitalRead(inputPin);              // read input value
if (val == HIGH) {                       // check if the input is HIGH
digitalWrite(ledPin, HIGH);             // turn LED ON
if (pirState == LOW) {
   // we have just turned on
Serial.println("Motion detected!");
   // We only want to print on the output change, not state
pirState = HIGH;
}
} else {
digitalWrite(ledPin, LOW);             // turn LED OFF
if (pirState == HIGH){
 // we have just turned off
Serial.println("Motion ended!");
// We only want to print on the output change, not state
pirState = LOW;
}
}
}
```



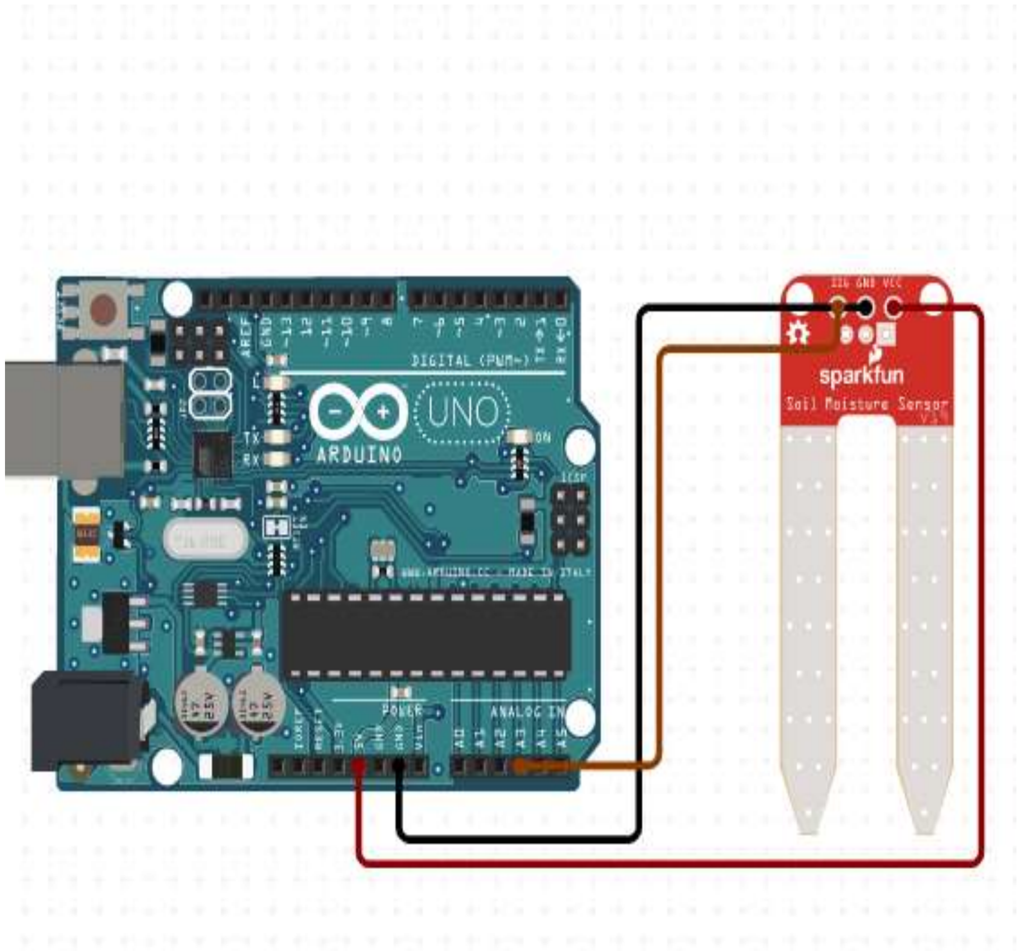
Moisture sensor

Soil moisture is basically the content of water present in the soil. This can be measured using a soil moisture sensor which consists of two conducting probes that act as a probe. It can measure the moisture content in the soil based on the change in resistance between the two conducting plates.

Pin Configuration:
VCC- Power(3.3-5v)
Gnd - 0V
Do- Digital Output
Ao -Analog Output

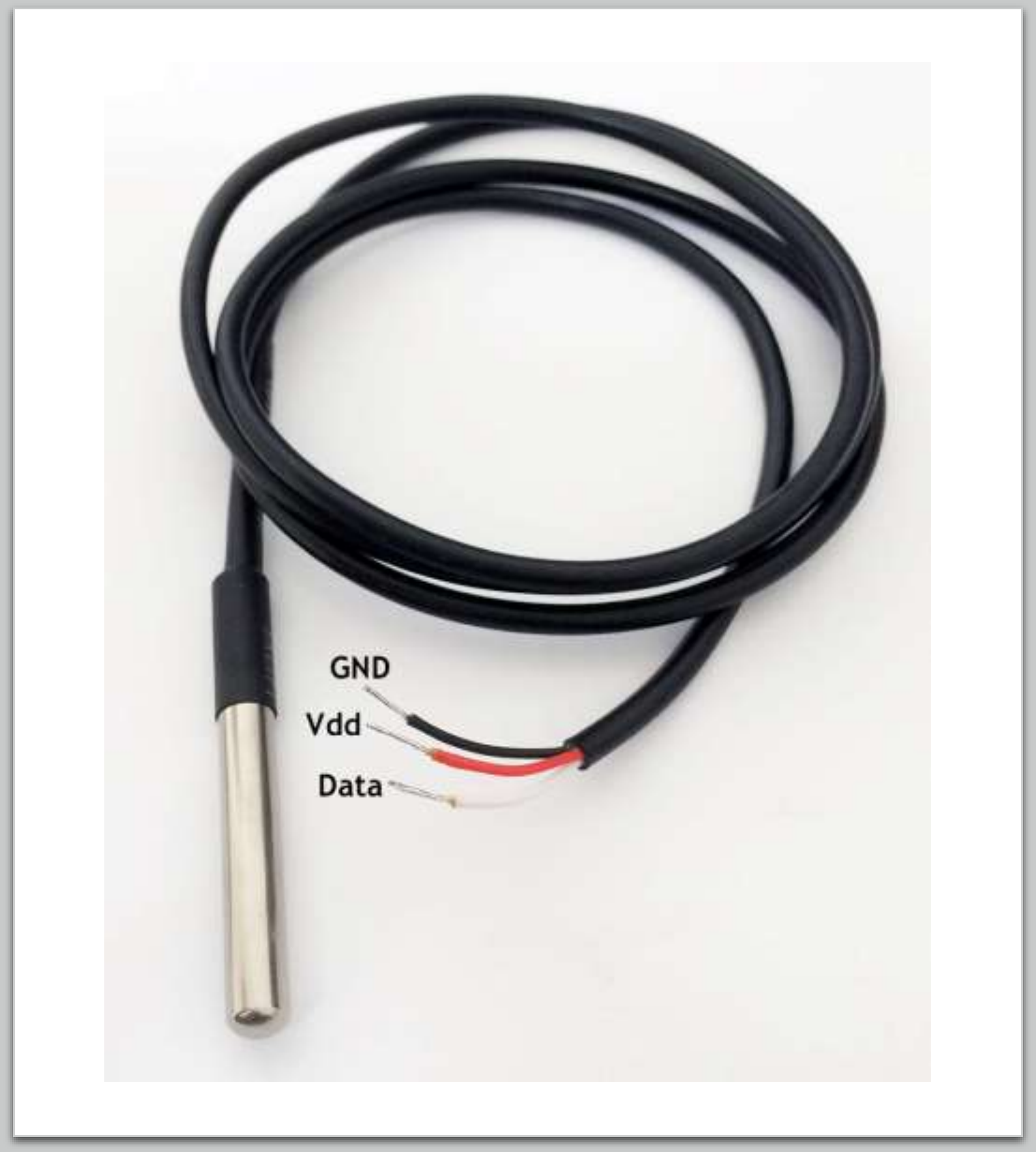
Simple example:

```
const int sensor_pin = A3; // Soil moisture   sensor O/P pin

  void setup()
{
 Serial.begin(9600);   // Define baud rate for serial communication
 }

 void loop()
 {
float moisture_percentage;
int sensor_analog;
sensor_analog = analogRead(sensor_pin); // Read analog value
moisture_percentage = ( 100 - ( (sensor_analog/1023.00) * 100 ) );
//else you can use map();function
 Serial.print("Moisture Percentage = ");
Serial.print(moisture_percentage);
Serial.print("%\n\n");
 delay(1000);
 }
```

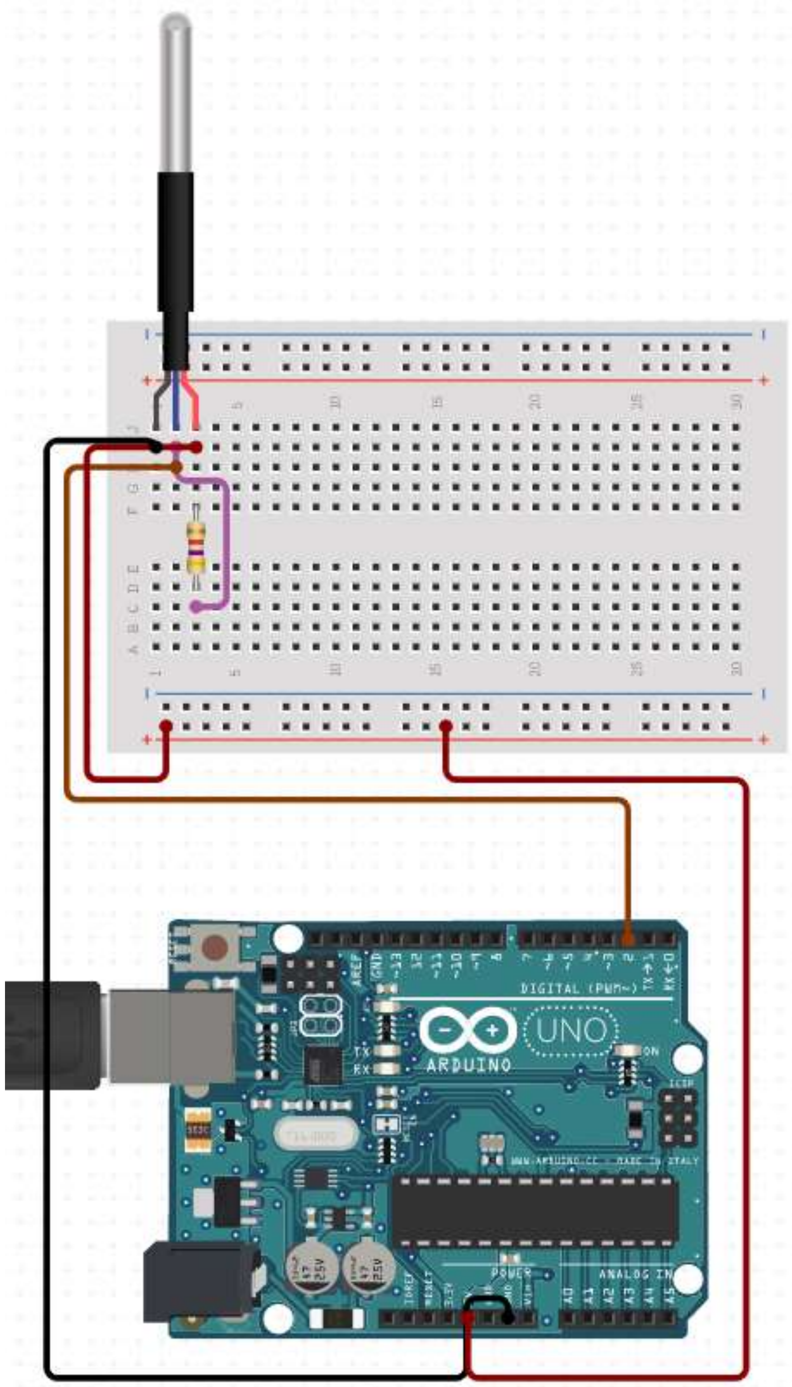


Temperature sensor(DS18B20)

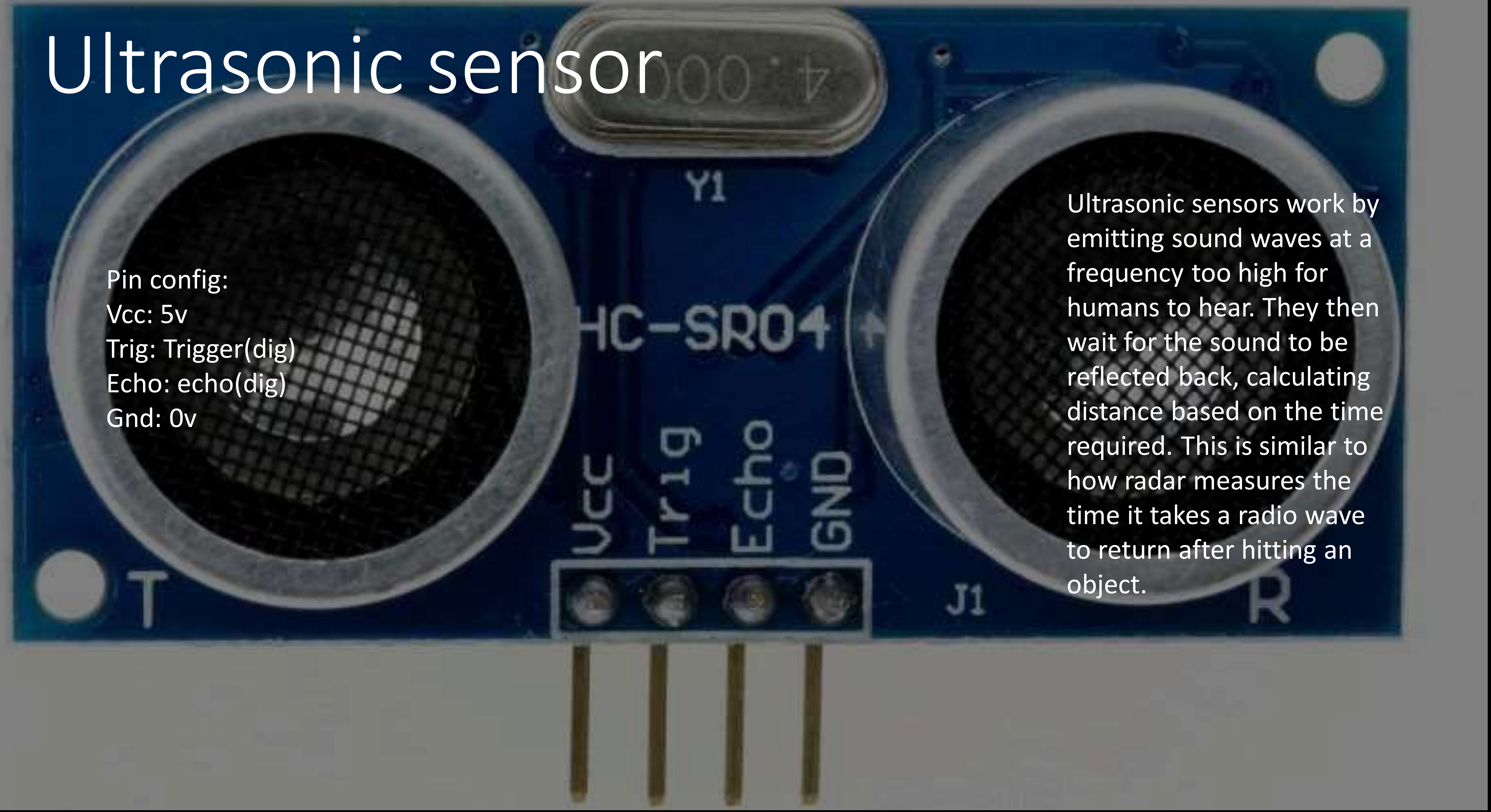
- These 1-wire digital temperature sensors are fairly precise (±0.5°C over much of the range) and can give up to 12 bits of precision from the onboard digital-to-analog converter.
- They work great with any microcontroller using a single digital pin, and you can even connect multiple ones to the same pin, each one has a unique 64-bit ID burned in at the factory to differentiate them. Usable with 3.0-5.0V systems.



```
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

 float Celcius=0;
 float Fahrenheit=0;
void setup(void)
{

  Serial.begin(9600);
  sensors.begin();
}
void loop(void)
{
  sensors.requestTemperatures();
  Celcius=sensors.getTempCByIndex(0);
  Fahrenheit=sensors.toFahrenheit(Celcius);
  Serial.print(" C  ");
  Serial.print(Celcius);
  Serial.print(" F  ");
  Serial.println(Fahrenheit);
  delay(1000);
}
```



Ultrasonic sensor

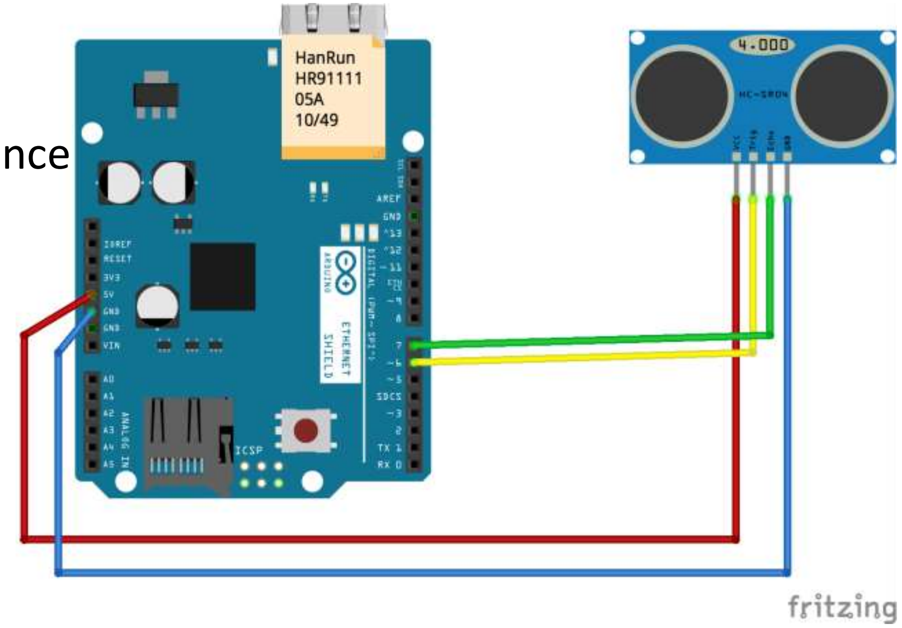


Pin config:
Vcc: 5v
Trig: Trigger(dig)
Echo: echo(dig)
Gnd: 0v

Ultrasonic sensors work by emitting sound waves at a frequency too high for humans to hear. They then wait for the sound to be reflected back, calculating distance based on the time required. This is similar to how radar measures the time it takes a radio wave to return after hitting an object.

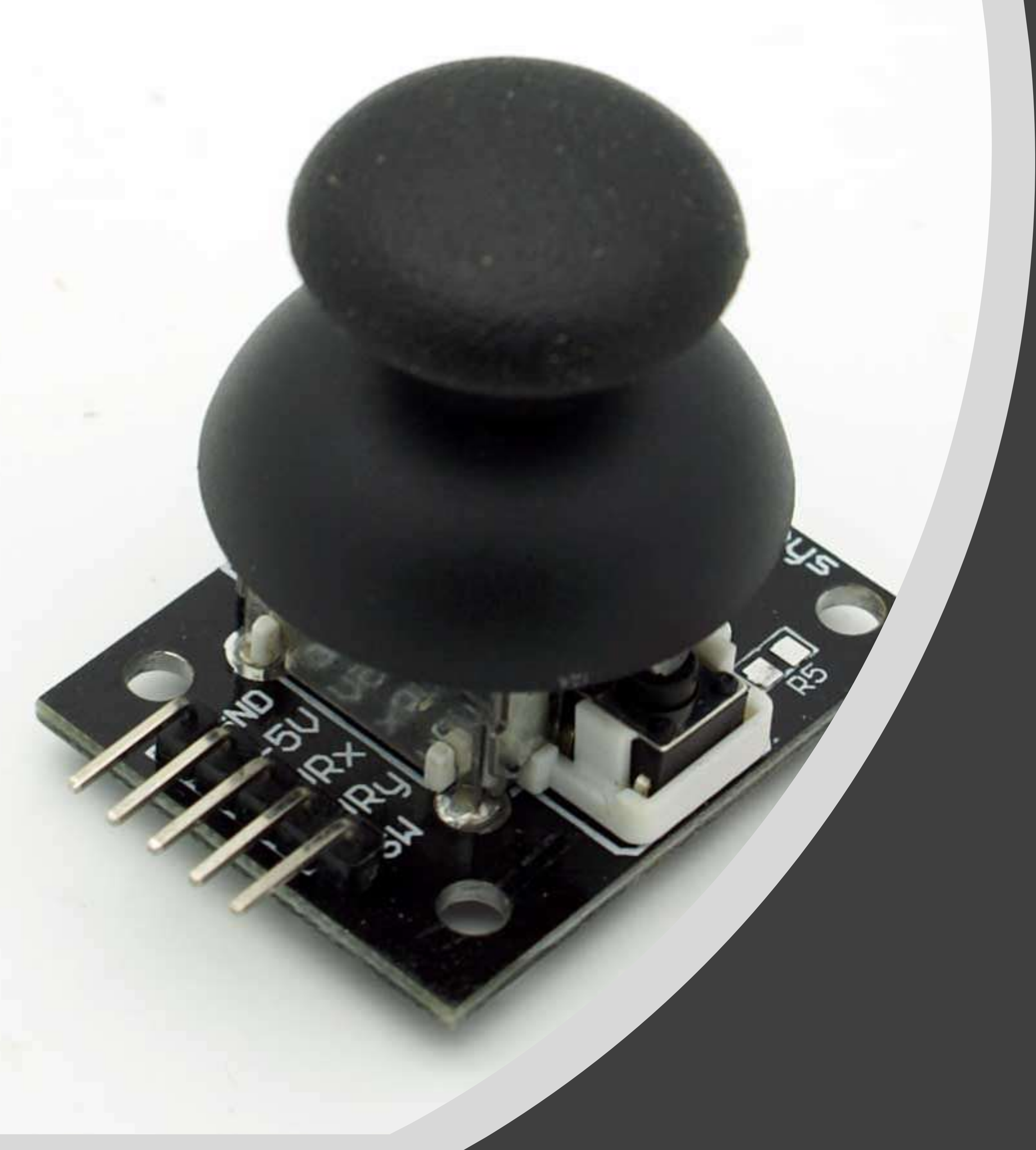
Simple example

```
int inches = 0;

int cm = 0;

long readUltrasonicDistance(int triggerPin, int echoPin) //function for finding distance
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}
```



cont.

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

}

void loop()
{
  // measure the ping time in cm
  cm = 0.01723 * readUltrasonicDistance(7, 7);
  // convert to inches by dividing by 2.54
  inches = (cm / 2.54);
  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.println("cm");
  delay(100); // Wait for 100 millisecond(s)
}
```



Joystick (just the integration of two potentiometers)

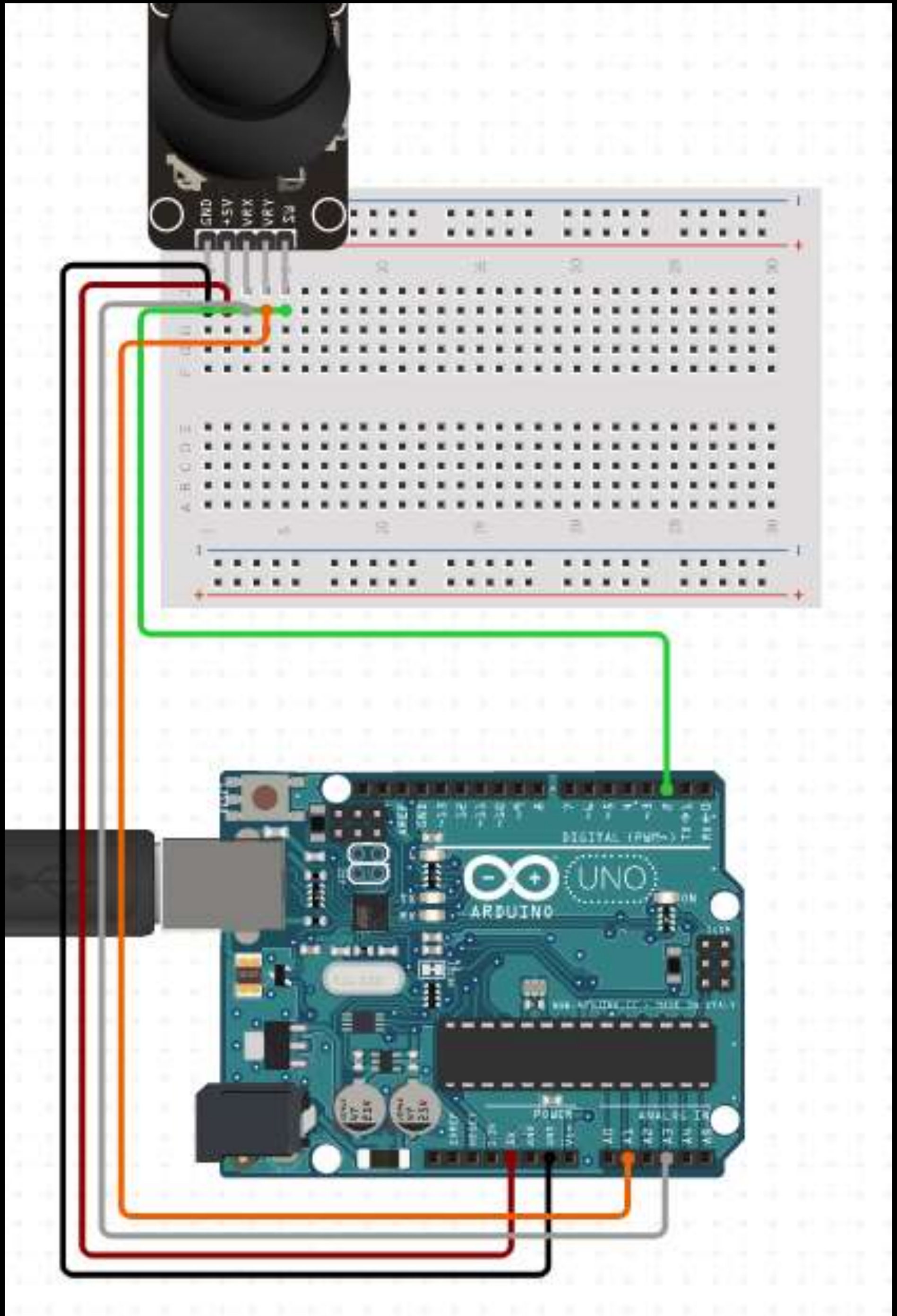
- The joystick is similar to two potentiometers connected together, one for the vertical movement (Y-axis) and other for the horizontal movement (X-axis).

Pin configuration:

- VCC - 5v
- GND - 0v
- VRx -> Variable resistance x
- VRY -> Variable resistance Y
- SW -> Switch / Button

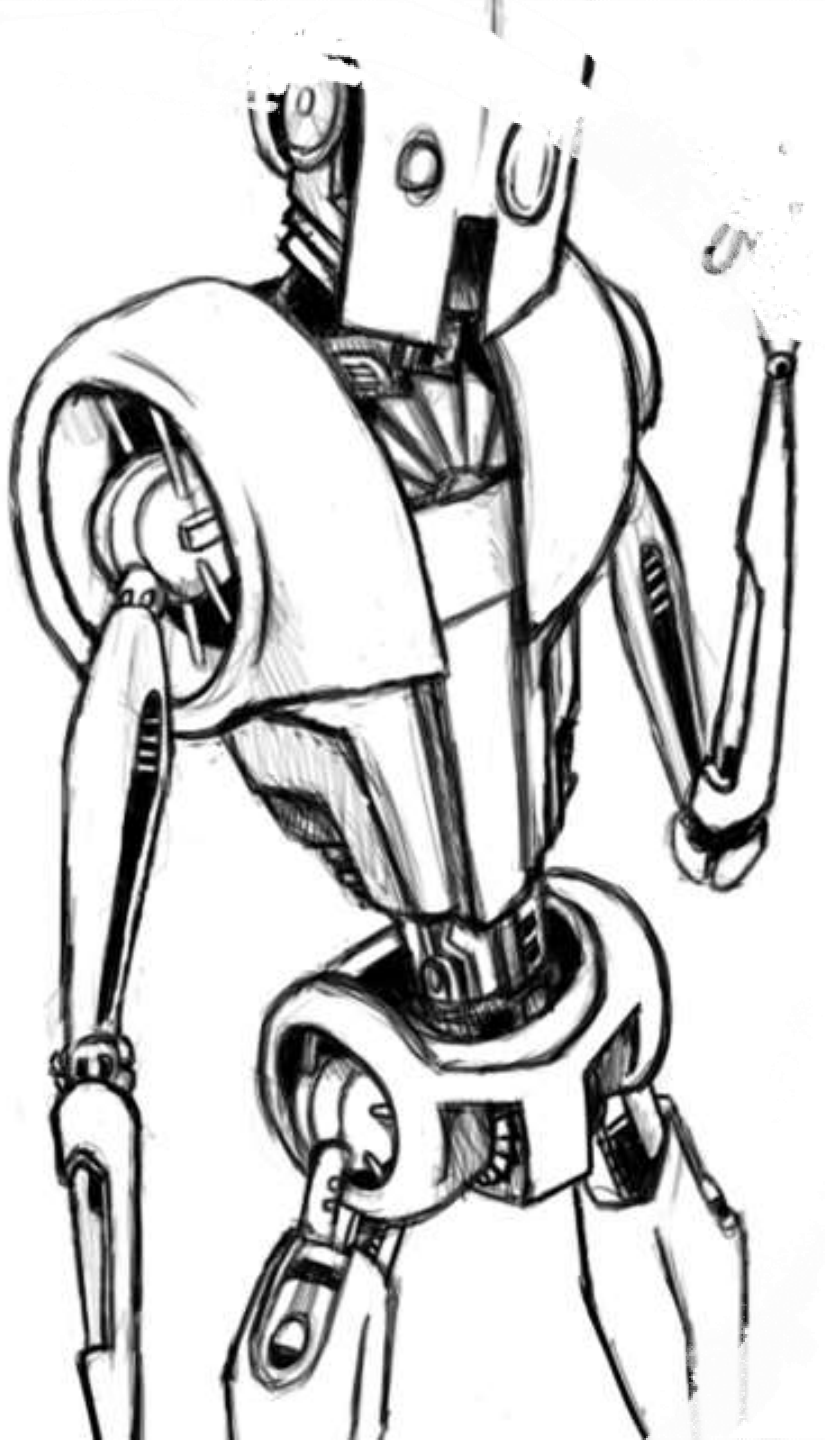
Simple example:

```
int SW = 2;//just to project the values of axis
int xPosition = 0;
int yPosition = 0;
int SW_state = 0;
int mapX = 0;
int mapY = 0;
void setup() {
  Serial.begin(9600);
  pinMode(A1, INPUT);
  pinMode(A3, INPUT);
  pinMode(SW, INPUT_PULLUP);
}
void loop() {
  xPosition = analogRead(A1);
  yPosition = analogRead(A3);
  SW_state = digitalRead(SW);
  mapX = map(xPosition, 0, 1023, -512, 512);
  mapY = map(yPosition, 0, 1023, -512, 512);

  Serial.print("X: ");
  Serial.print(mapX);
```



Some of other Sensors you could find:

- Accelerometer
- Pressure Sensor
- Light Sensor
- Smoke, Gas and Alcohol Sensor
- Touch Sensor
- Humidity Sensor
- Tilt Sensor
- Color sensor
- Flow and Level Sensor
- Torque sensor
- Variable reluctance sensor
- Hall effect sensor
- AFR sensor(Air fuel ratio)
- PH sensor
- Dissolved oxygen sensor
- Leaf sensor
- Impact sensor
- Flex sensor
- Shack–Hartmann wave front sensor



There are still many sensors and modules out there.. explore them!