SRM UNIVERSITY
DELHI-NCR, SONEPAT

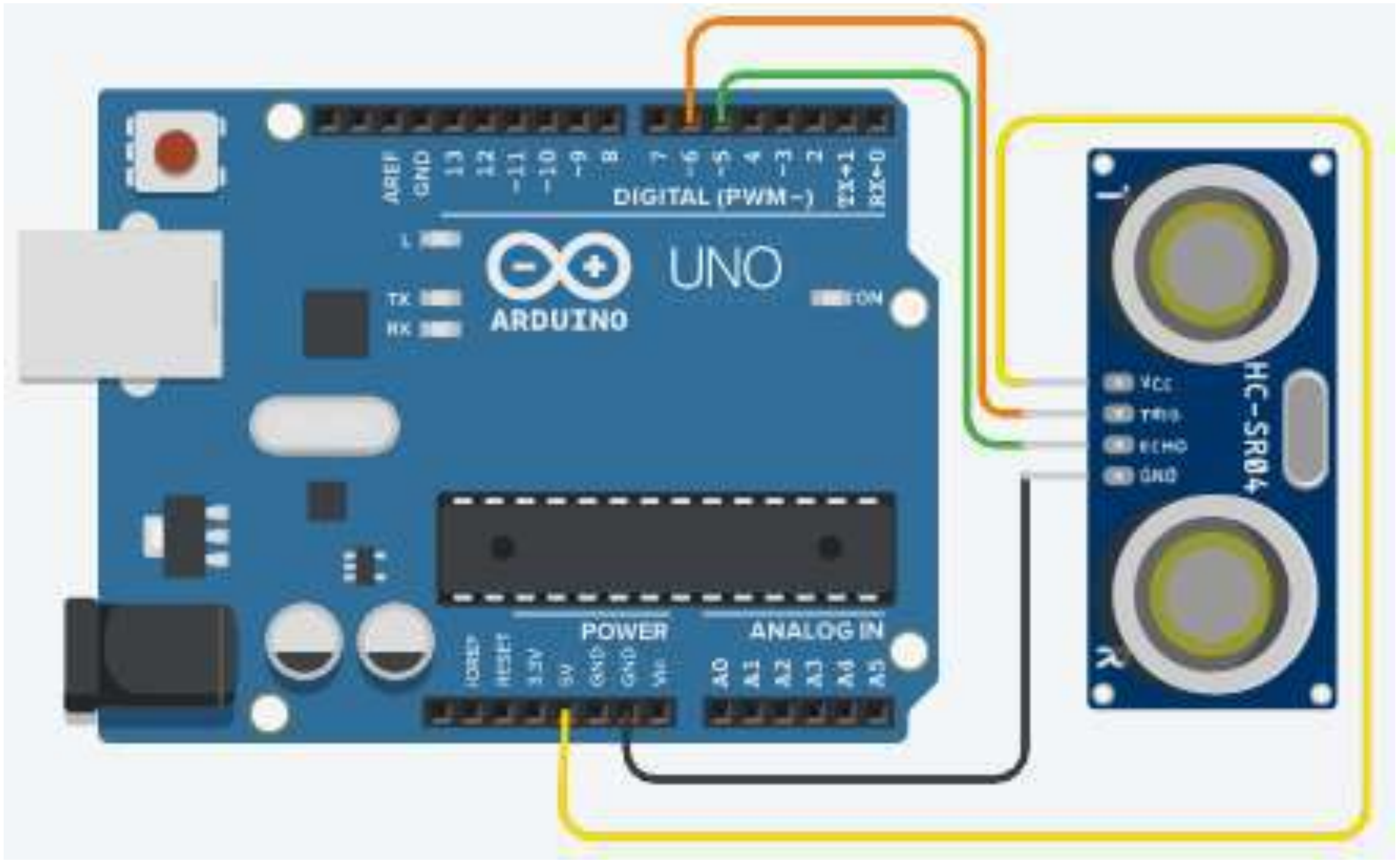
LEARN. LEAP. LEAD

MHRD'S INNOVATION CELL
(GOVERNMENT OF INDIA)

INSTITUTION'S INNOVATION COUNCIL
(Ministry of HRD Initiative)

CENTER FOR INNOVATION INCUBATION & ENTREPRENEURSHIP
(C I I E)

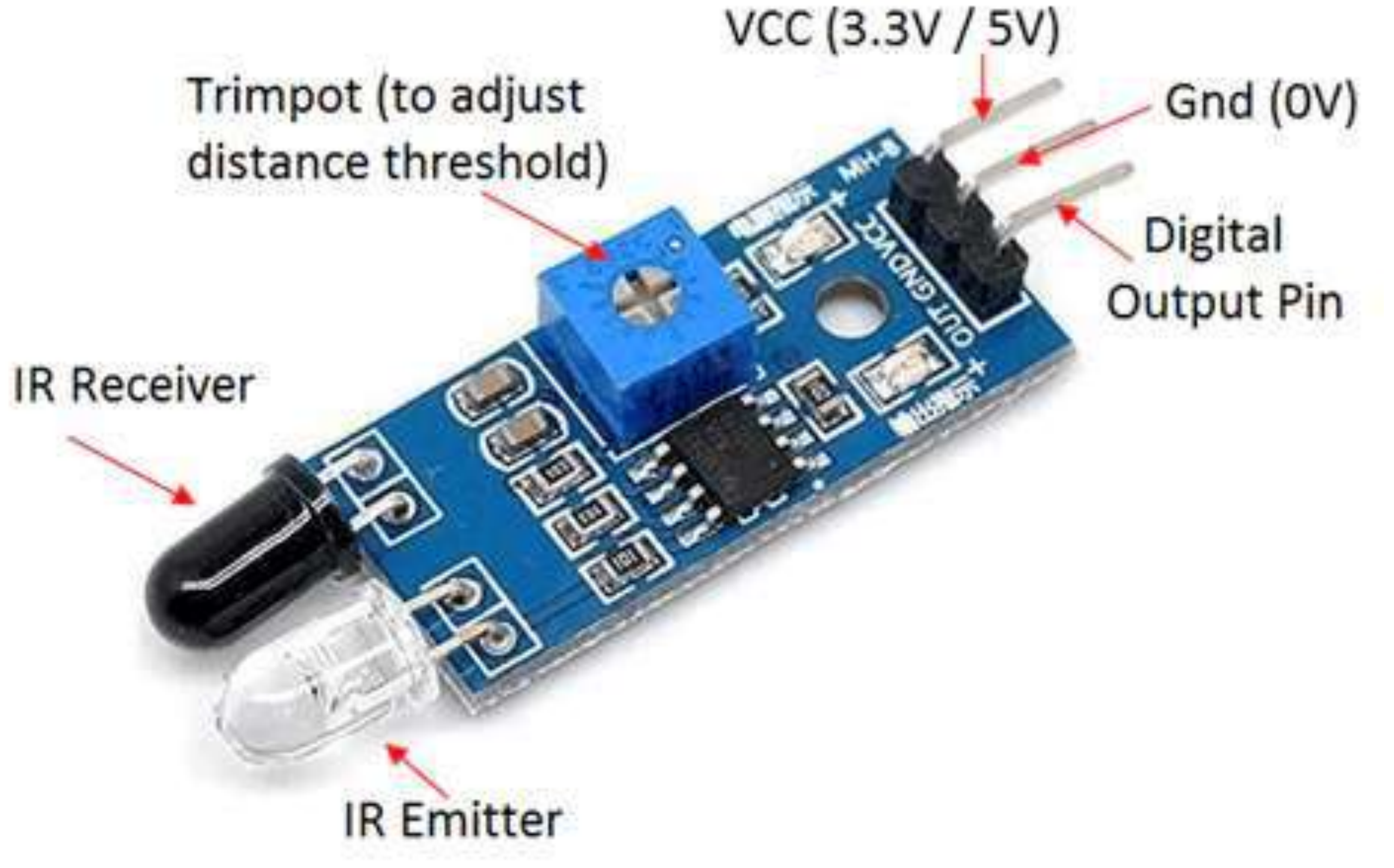
ARDUINO WORKSHOP



AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
POWER
IOREF
RESET
3.3V
5V
GND
GND
Vin
ANALOG IN
A0
A1
A2
A3
A4
A5
T
VCC
TRIG
ECHO
GND
HC-SR04
R

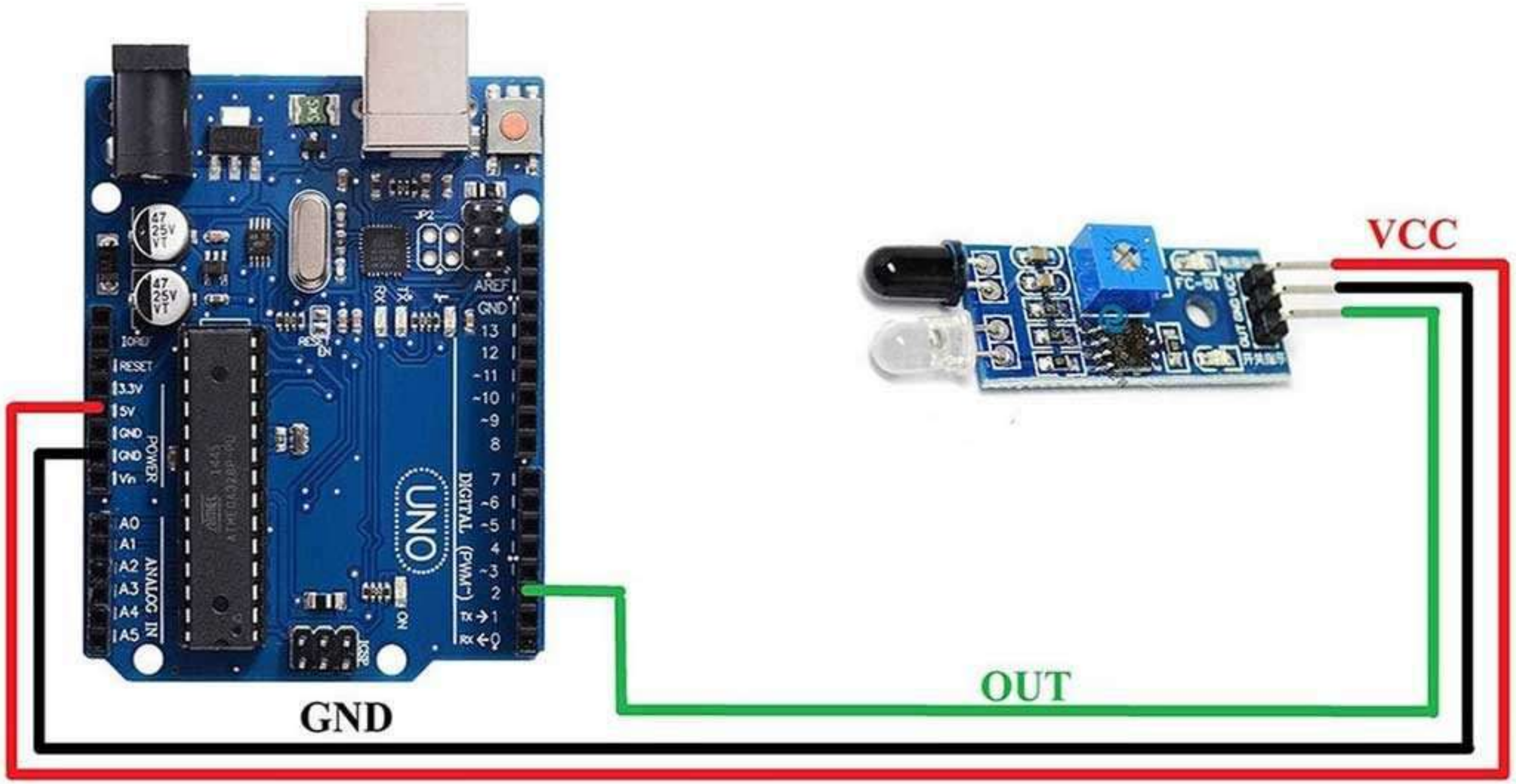
IR SENSOR

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. The sensor has a maximum range of around **40-50 cm indoors and around 15-20 cm outdoors**.



DIFFERENCE BETWEEN ULTRASONIC SENSOR AND IR SENSOR

ULTRASONIC SENSOR	IR SENSOR
OBSERVES DISTANCE	OBSERVES MOTION
USES ULTRASONIC WAVES	USES INFRARED WAVES


VCC
OUT
GND
FC-51
OUT GND VCC
UNO
DIGITAL (PWM~)
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
ON

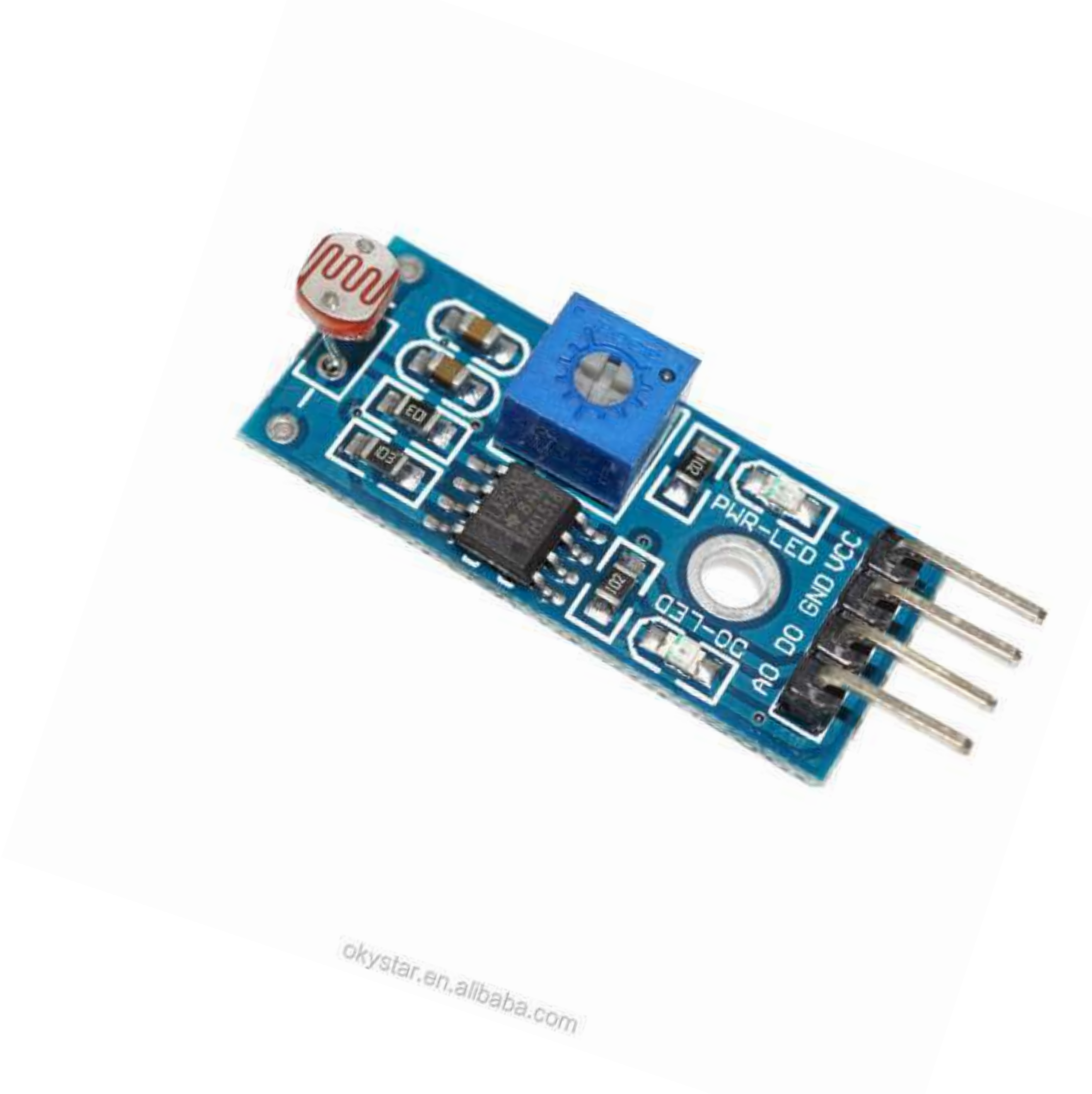
```
const int IR_PIN = 9;     // define IR sensor pin number as constant
void setup() {
          pinMode(IR_PIN, INPUT);     // set IR sensor pin as input
          Serial.begin(9600);      // initialize serial communication
          }
void loop() {
          int ir_value = digitalRead(IR_PIN);      // read IR sensor value
          if (ir_value == LOW)
           { // check if IR sensor is detecting an object
          Serial.println("Object detected!");    // print message to serial
monitor
          }
          delay(500); // wait for half a second
}
```

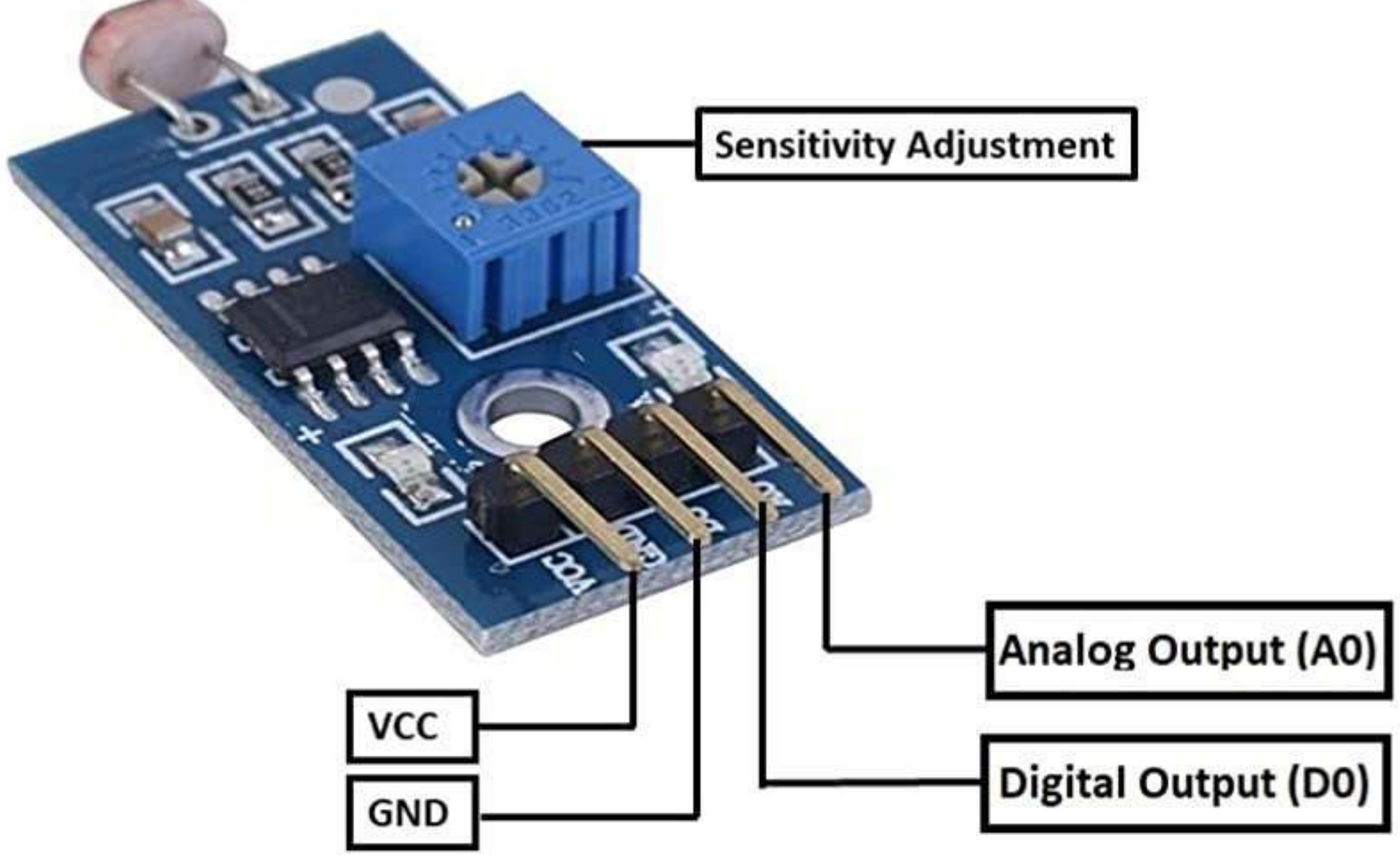
LDR

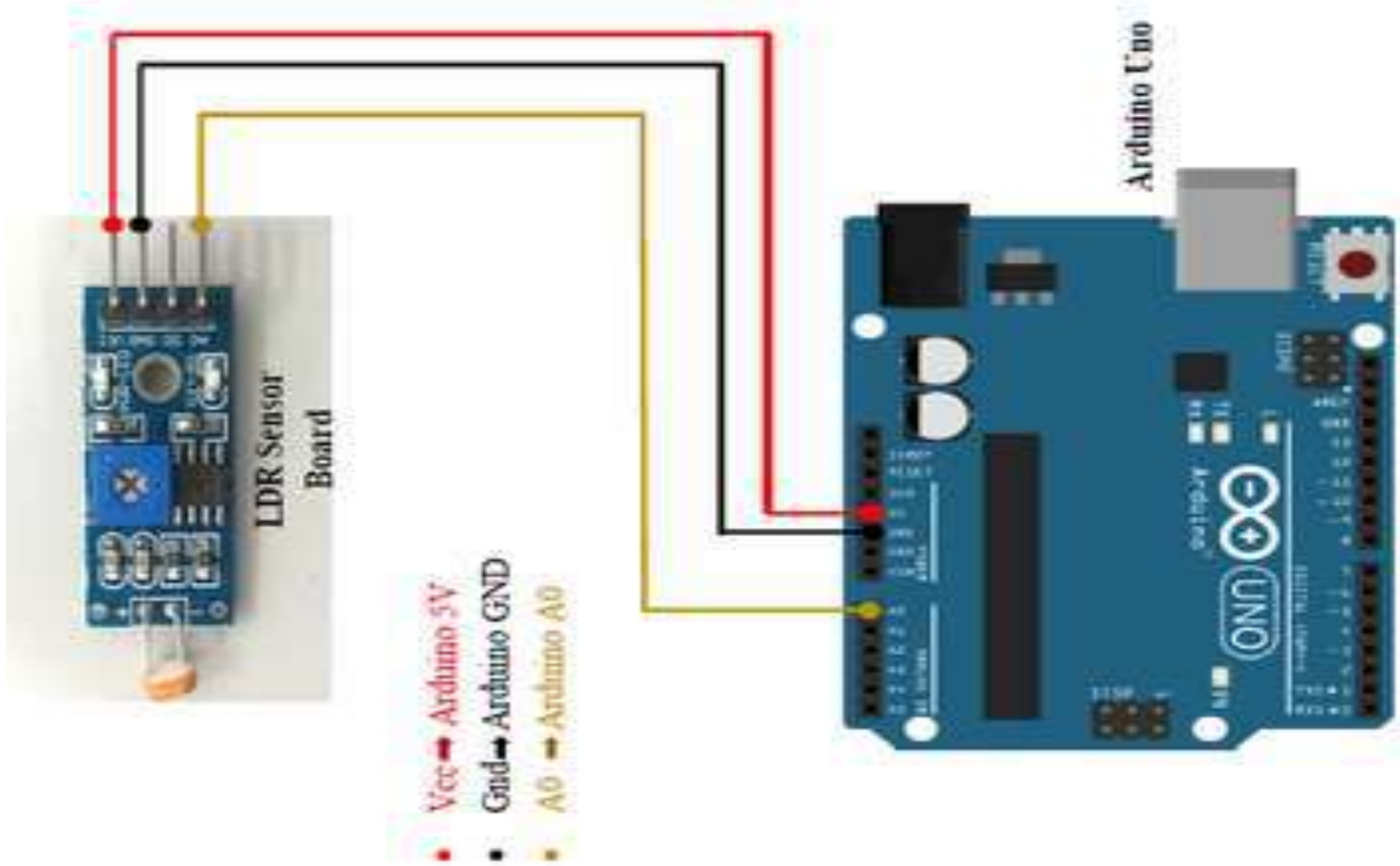
LDR (Light Dependent Resistor) uses the photoresistor to light an LED. The LED will light up whenever there is dark or no light over the sensor.

What is photoresistor?

It is defined as a light-controlled resistor, which is also called as LDR. It is a variable resistor that controls the resistance in accordance with the received light intensity. It means, the resistance decreases as intensity of light increases.



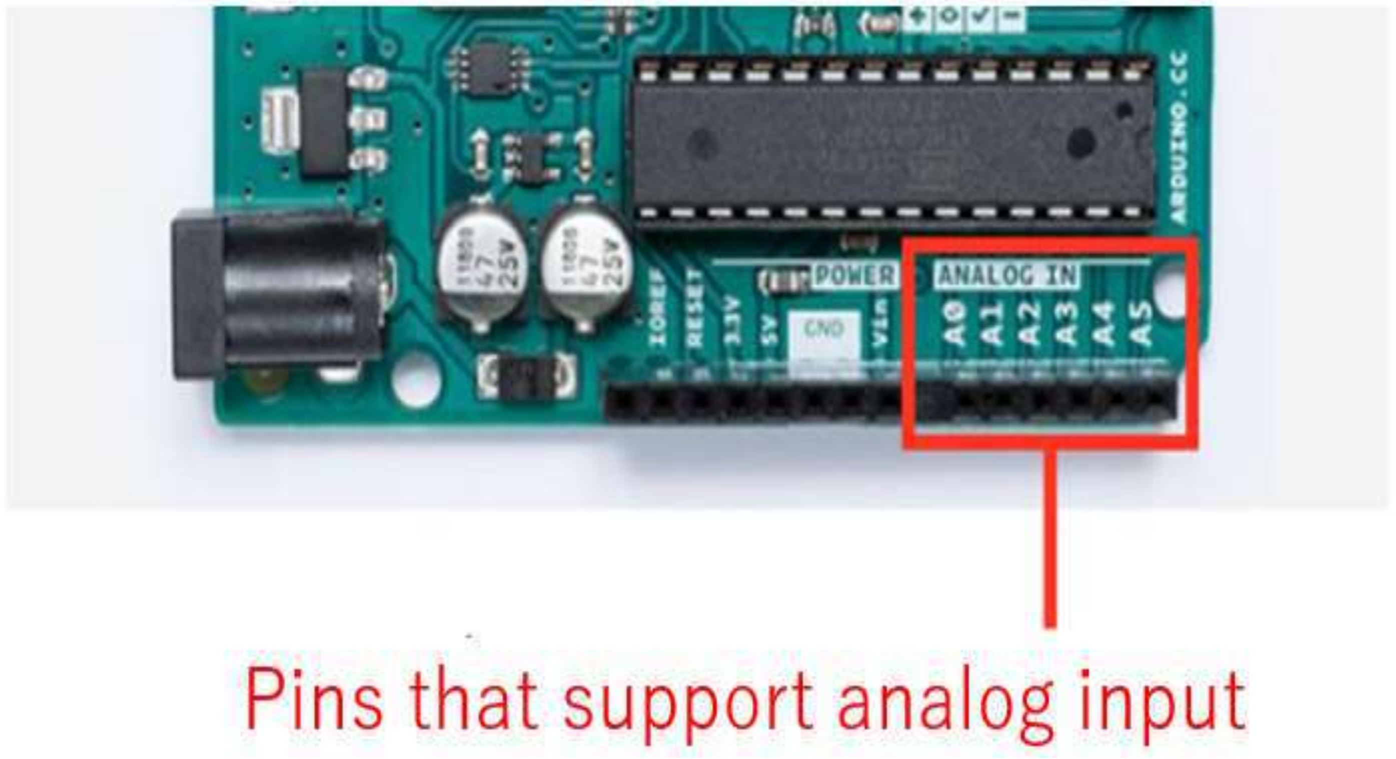

Sensitivity Adjustment
Analog Output (A0)
Digital Output (D0)
VCC
GND


Arduino Uno
LDR Sensor Board
● Vcc → Arduino 5V
● Gnd → Arduino GND
● A0 → Arduino A0

```
const int LDR_PIN = 3;
//define LDR pin number as constant

void setup() {
  pinMode(LDR_PIN, INPUT); // set LDR pin as input
  Serial.begin(9600); // initialize serial communication
}

void loop() {
  int ldr_value = digitalRead(LDR_PIN); // read LDR value

  Serial.print("LDR value: ");
  Serial.println(ldr_value);
// print LDR value to serial monitor

  delay(500); // wait for half a second
}
```

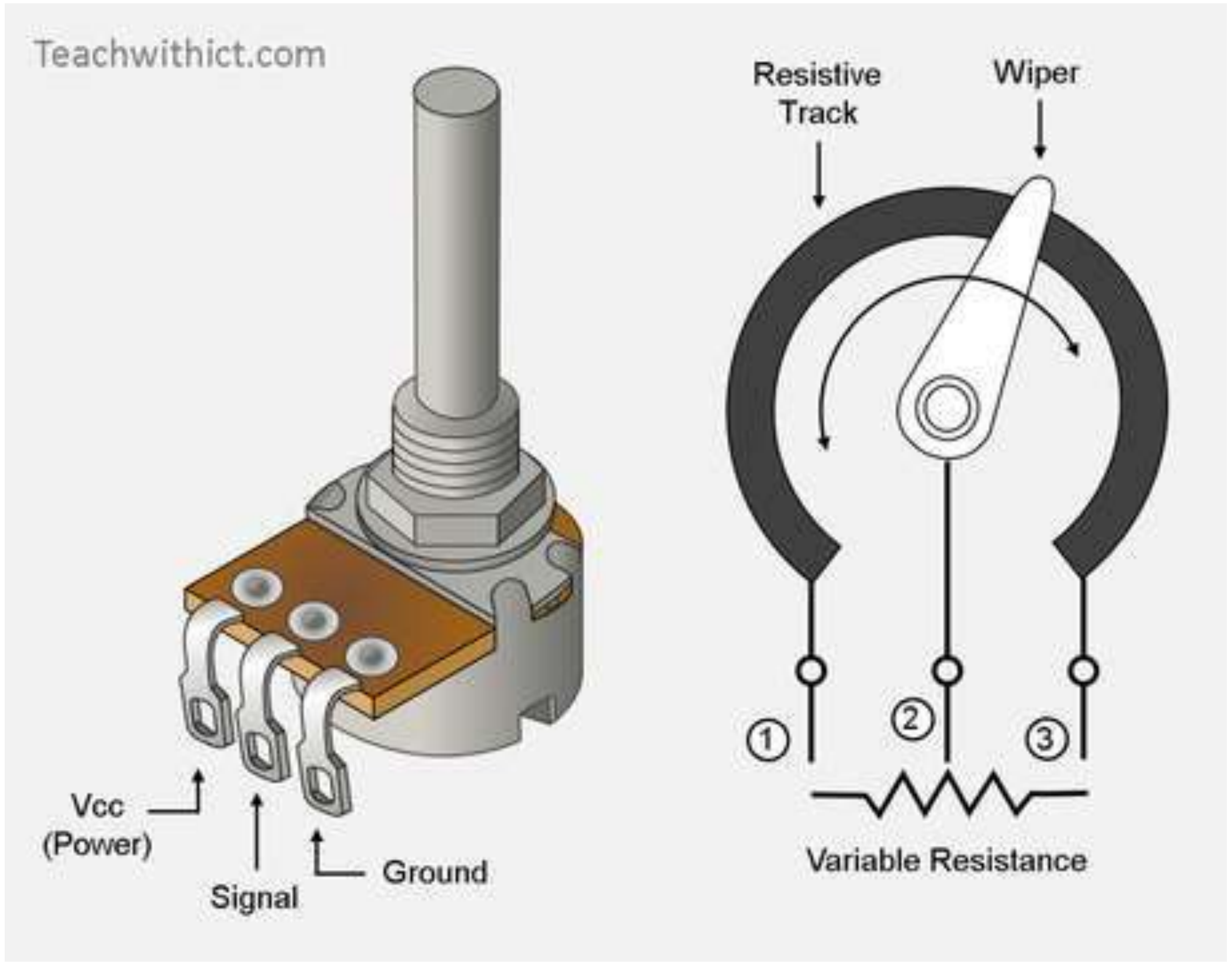
ANALOG PIN

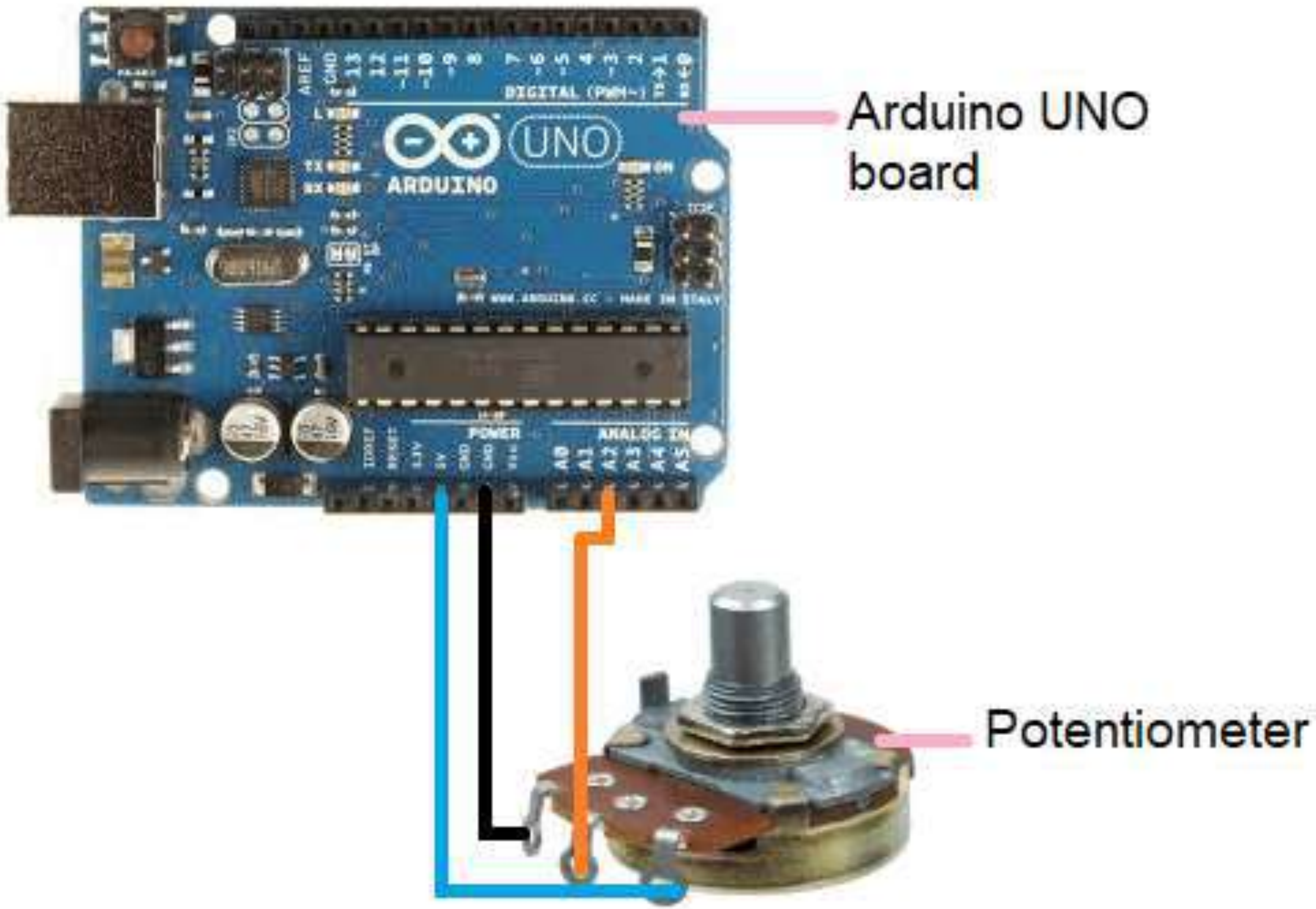
The main function of the analog pins for most Arduino users is to read analog sensors, the analog pins also have all the functionality of general purpose input/output (GPIO) pins (the same as digital pins 0 - 13).
The Arduino Uno has 6 **analog pins**, which utilize ADC (Analog to Digital converter).
These pins serve as analog inputs but can also function as digital inputs or digital outputs.



POTENTIOMETER

A **potentiometer** is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a **variable resistor** or **rheostat**. Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment.




Arduino UNO board
Potentiometer

```
const int POT_PIN = A0;
// define potentiometer pin number as constant

void setup() {
  pinMode(POT_PIN, INPUT); // set potentiometer pin as input
  Serial.begin(9600); // initialize serial communication
}
void loop() {
  int pot_value = analogRead(POT_PIN);
// read potentiometer value

  Serial.print("Potentiometer value: ");
  Serial.println(pot_value);
// print potentiometer value to serial monitor

  delay(500); // wait for half a second
```

SERVO MOTORS

WHAT IS SERVO MOTOR?

Servo motor is a component that can rotate its handle (usually between 0° and 180°). It used to control the angular position of the object.

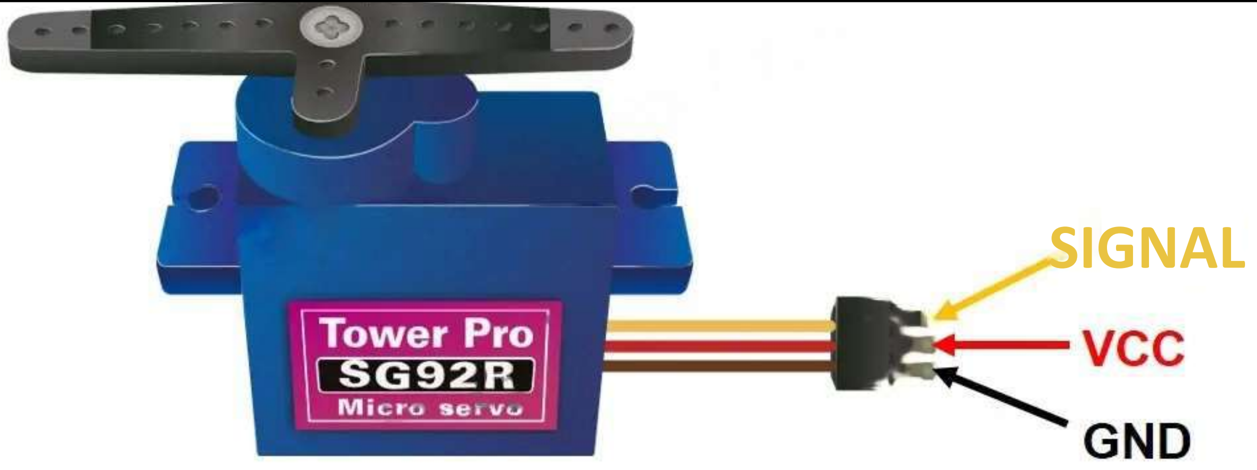
APPLICATIONS OF SERVO MOTOR

MOTOR	APPLICATIONS
SERVO MOTOR	MACHINE TOOLS, AEROPLANES AND DVDs

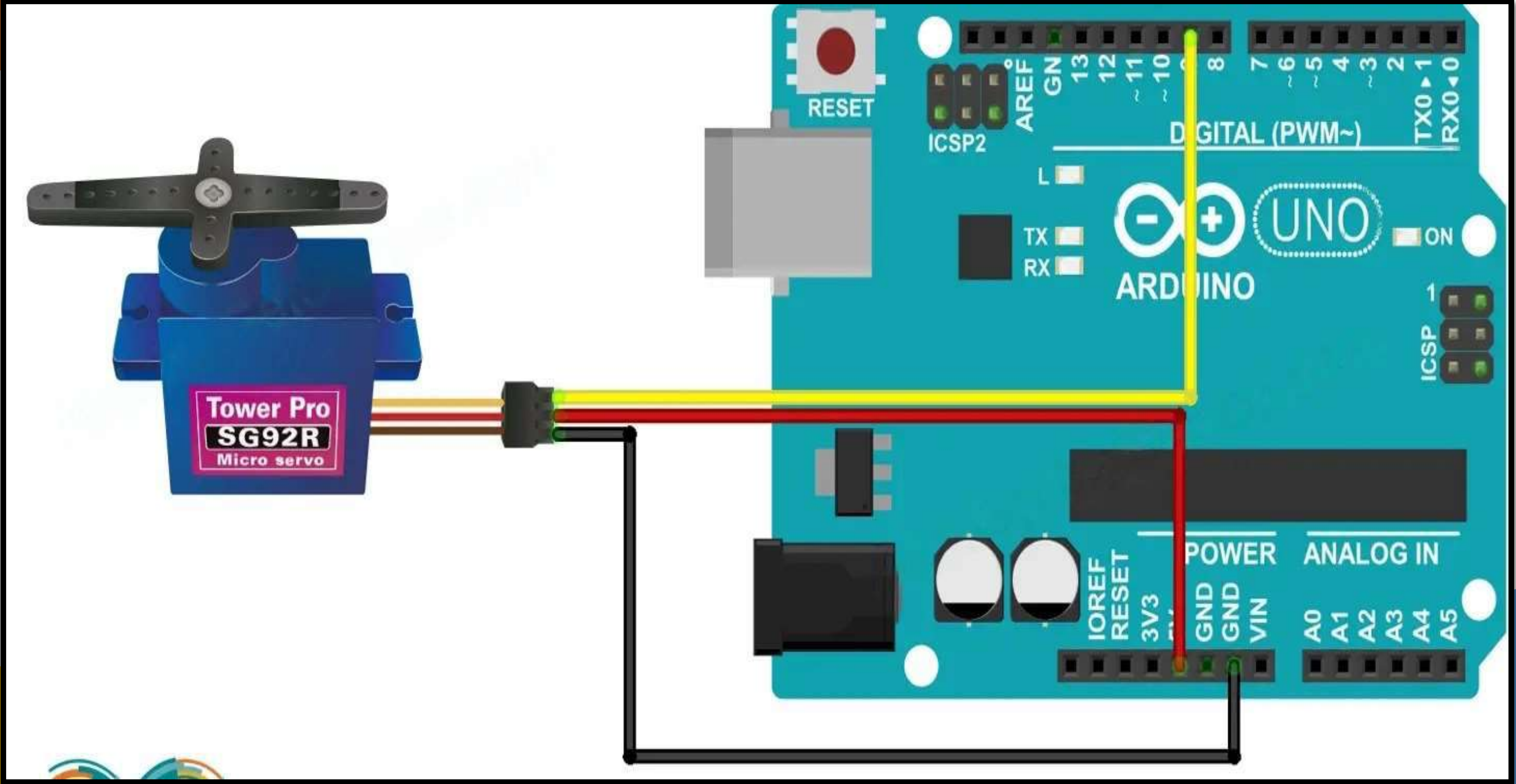
PIN-OUTS OF SERVO MOTOR

The servo motor used in this example includes three pins:

VCC PIN	**(typically red) needs to be connected to VCC (5V)**
GND PIN	**(typically brown or black) needs to connected to GND (0V)**
SIGNAL PIN	**(typically yellow or orange) receives the PWM control signal from an Arduino's pin**



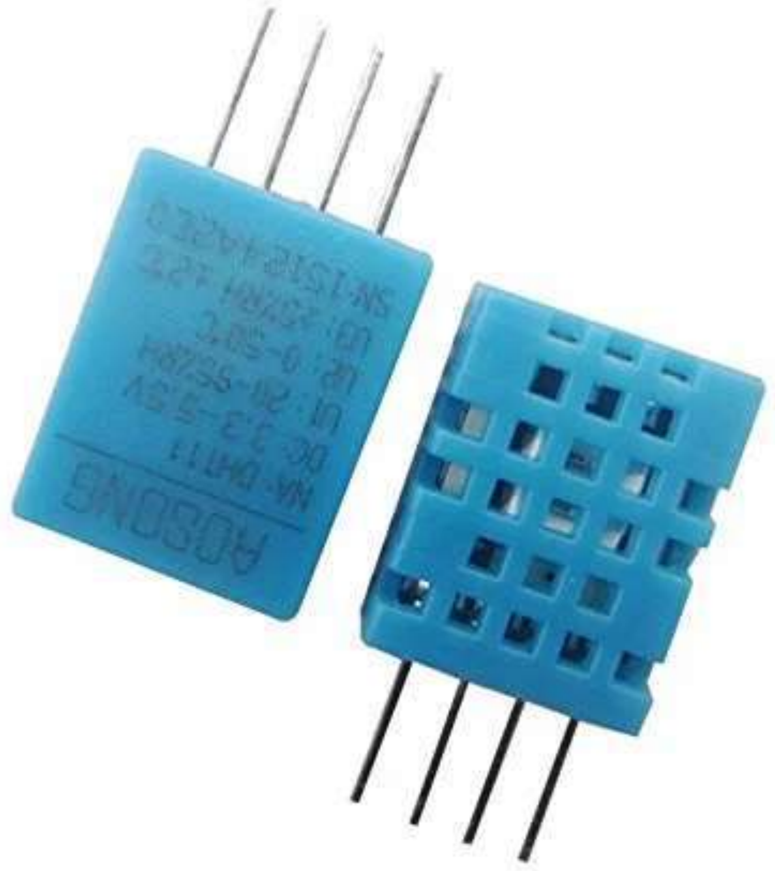
CIRCUIT DIAGRAM



```
#include <Servo.h>

const int SERVO_PIN = 9;   define the pin number for the servo motor

Servo myservo;   create a servo object

void setup() {
  myservo.attach(SERVO_PIN);   attaches the servo on SERVO_PIN to the servo object
}

void loop() {
  myservo.write(0);   set servo position to 0 degrees
  delay(1000);       wait for 1 second
  myservo.write(90);  set servo position to 90 degrees
  delay(1000);       wait for 1 second
  myservo.write(180);  set servo position to 180 degrees
  delay(1000);       wait for 1 second
}
```

DHT SENSORS

- The DHT11 and DHT22 sensors are commonly used to measure temperature and humidity.
- These sensors are easy to use and cost-effective.
- DHT11 and DHT22 are digital sensors that can measure temperature and humidity.
- They have a built-in thermistor (**resistor whose resistance is dependent on temperature**) and a capacitive humidity sensor.

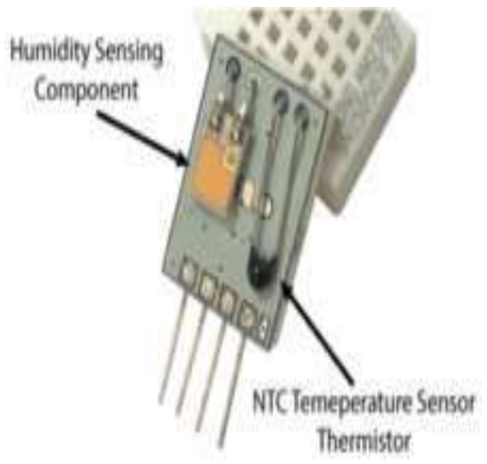


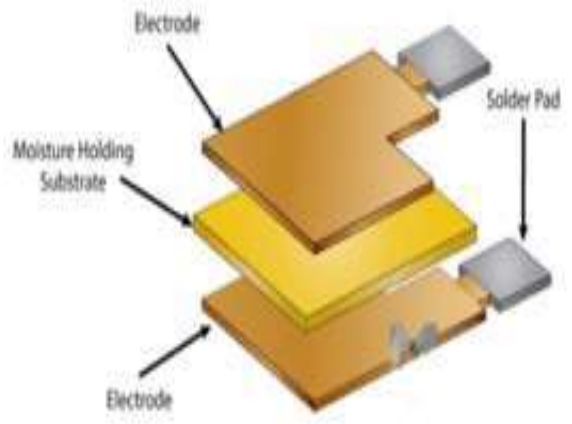
DHT11 V/S DHT22

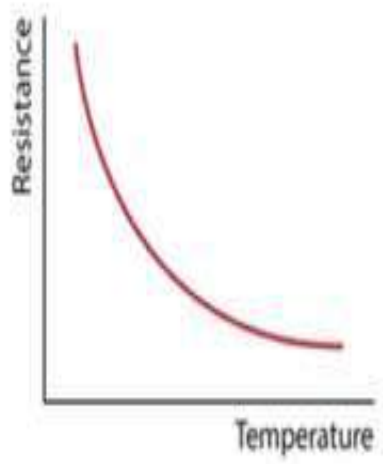
1. **Measurement range**: The DHT11 measures temperature in the range of 0-50°C with an accuracy of ±2°C, and humidity in the range of 20-90% RH with an accuracy of ±5% RH. The DHT22, on the other hand, measures temperature in the range of -40-80°C with an accuracy of ±0.5°C, and humidity in the range of 0-100% RH with an accuracy of ±2-5% RH.
2. **Response time**: The DHT11 has a slower response time compared to the DHT22. The DHT11 takes around 2 seconds to provide a new reading, while the DHT22 takes around 0.5 seconds.
3. **Sampling rate**: The DHT11 has a lower sampling rate than the DHT22. The DHT11 can provide a new reading once every 2 seconds, while the DHT22 can provide a new reading once every 0.5 seconds.
4. **Power consumption**: The DHT11 consumes less power than the DHT22, making it more suitable for battery-powered applications.

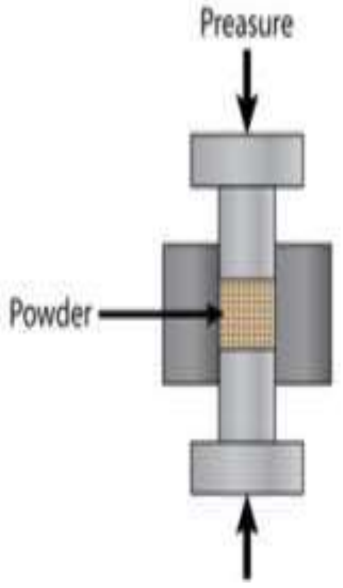
How DHT11 Measures Temperature and Humidity?

- Inside the DHT11, there is a thermistor (**resistor whose resistance is dependent on temperature**) with a humidity sensor component.
- The humidity sensor component consists of two electrodes containing a dehydrated substrate sandwich.
- The ions are released into the substrate as water vapor absorbs it, which in turn increases the conductivity between the electrodes.
- The resistance change between the two electrodes is proportional to the relative humidity. High relative humidity reduces the resistance between the electrodes, while low relative humidity increases the


Humidity Sensing Component
NTC Temeperature Sensor Thermistor







DHT11 also includes an NTC / Thermistor(**resistor whose resistance is dependent on temperature**) to measure temperature. A thermistor is a thermal resistor whose resistance changes rapidly with temperature. The word “NTC” means “negative temperature coefficient“, which means that the resistance decreases with increasing temperature.

<dht.h> LIBRARY

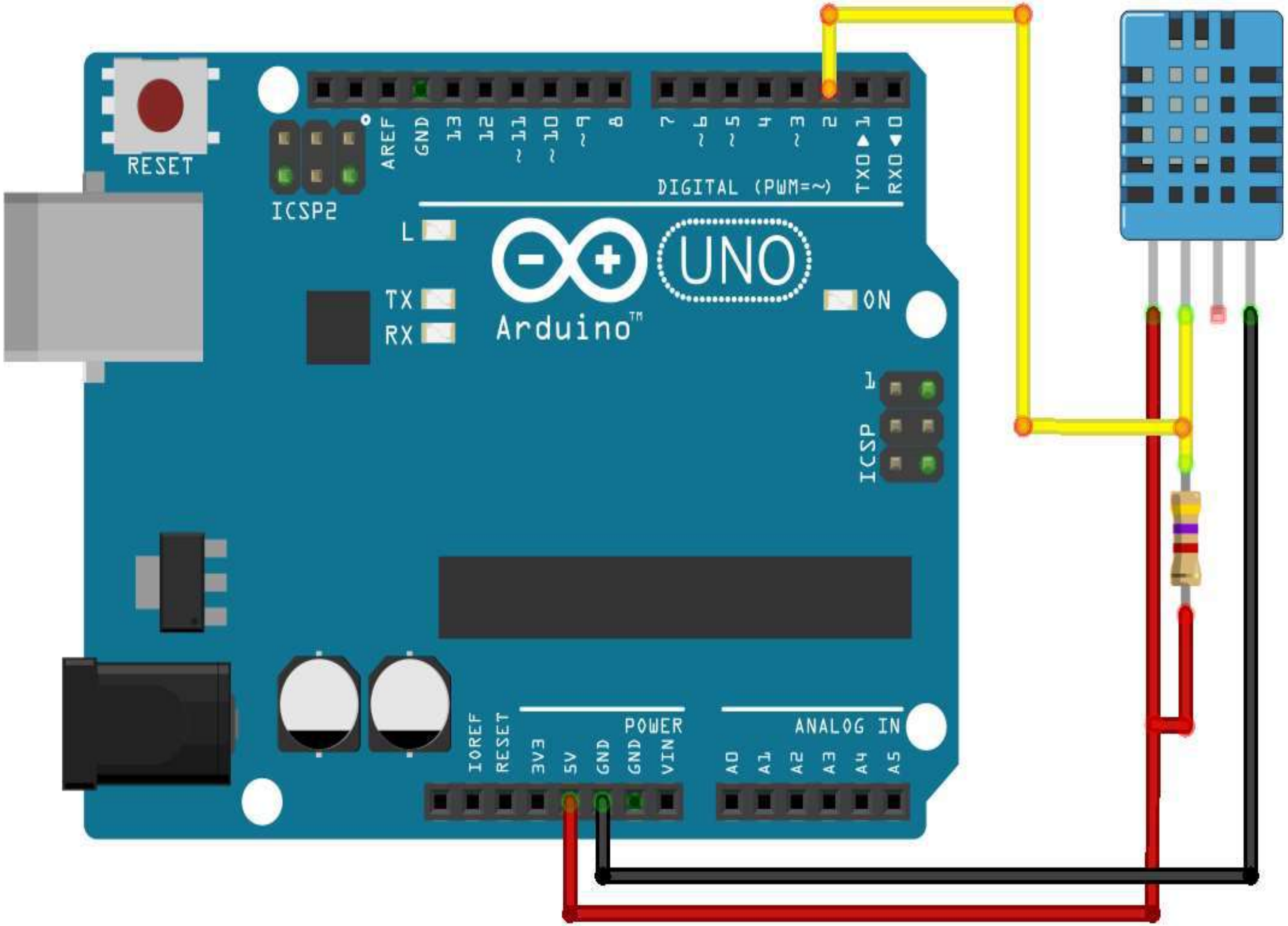
The <dht.h> library is a C++ library for the Arduino platform that provides an easy way to interface with DHT11, DHT21, and DHT22 digital temperature and humidity sensors.
The DHT sensors are low-cost, low-power devices that use a one-wire communication protocol to transmit temperature and humidity data to a microcontroller such as an Arduino. The <dht.h> library provides a set of functions that enable the Arduino to read temperature and humidity data from these sensors.

The <dht.h> library is typically used in conjunction with the Arduino IDE (Integrated Development Environment) to develop applications that require temperature and humidity sensing. To use the library, you need to include the header file "dht.h" in your sketch and create an instance of the DHT class. Then you can call the functions provided by the library to read temperature and humidity data from the sensor.

Here are some examples of functions provided by the <dht.h> library:

- dht.read11() - reads temperature and humidity data from a DHT11 sensor
- dht.read22() - reads temperature and humidity data from a DHT22 sensor
- dht.temperature - returns the temperature data in Celsius
- dht.humidity - returns the relative humidity data in percent

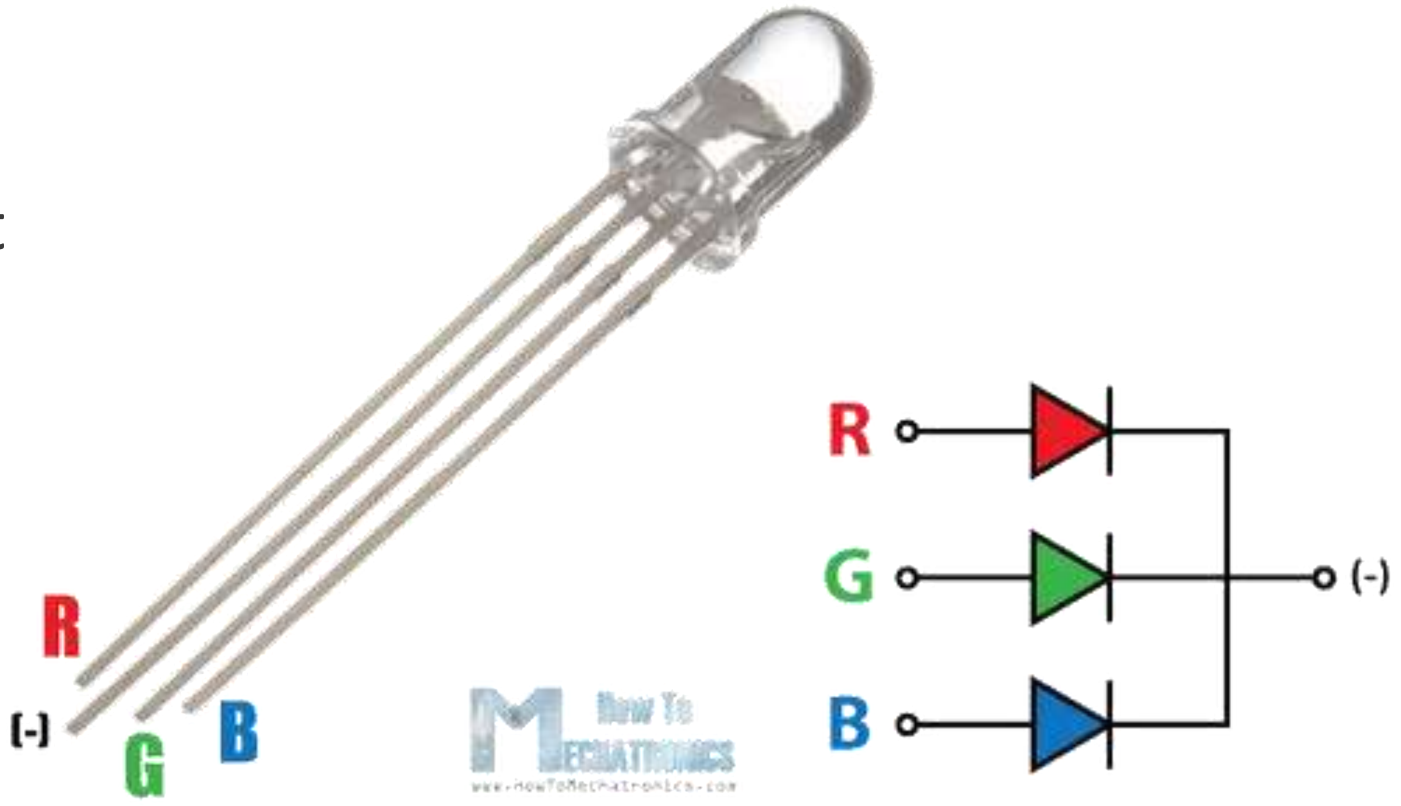
Overall, the <dht.h> library simplifies the process of interfacing with DHT temperature and humidity sensors, allowing developers to focus on their application logic rather than the intricacies of sensor communication.


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM=~)
L
TX
RX
Arduino™
UNO
ON
1
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.print(" Humidity: ");
  Serial.print(h);
  Serial.print("%  Temperature:  ");
  Serial.print(t);
  Serial.println(" *C");
}
```

RGB

The RGB LED can emit different colors by mixing the 3 basic colors red, green and blue. So it actually consists of 3 separate LEDs red, green and blue packed in a single case. That’s why it has 4 leads, one lead for each of the 3 colors and one common cathode or anode depending of the RGB LED type.

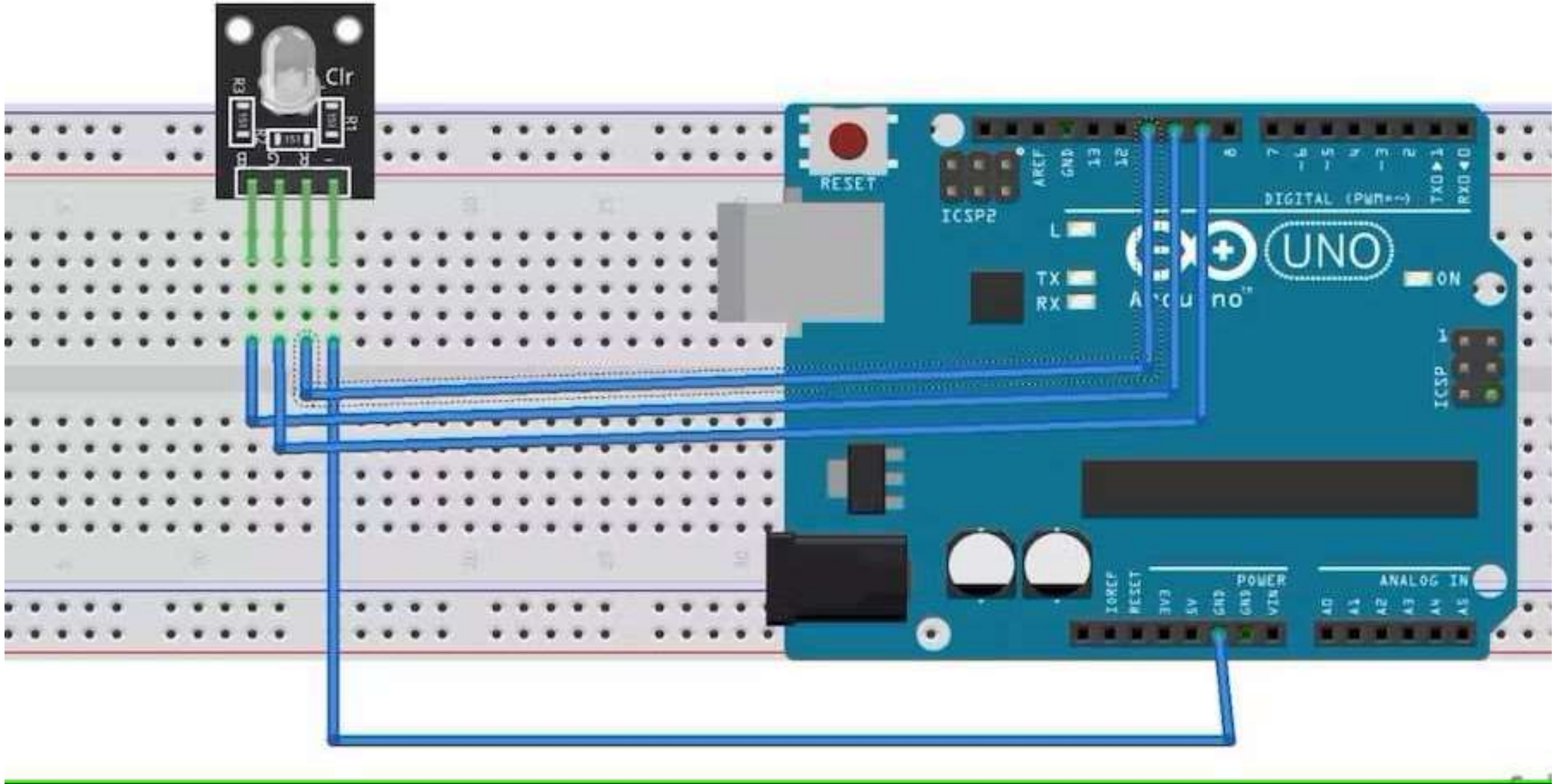


In the nature of physics, a color is composed of three color values: Red (R), Grean (G) and Blue (B). Each color value ranges from 0 to 255.

The mix of three values creates 256 x 256 x 256 colors in total.

If we provide PWM signals (with duty cycle from 0 to 255) to R, G, B pins, we can makes RGB LED displays any color we want.

The duty cycle of PWM signals to R, G and B pins correspond to color values of Red (R), Grean (G) and Blue (B) www.itdealsindia.com



```
const int RED_PIN = 9; // define red pin number as constant
const int GREEN_PIN = 10; // define green pin number as constant
const int BLUE_PIN = 11; // define blue pin number as constant

void setup() {
  pinMode(RED_PIN, OUTPUT); // set red pin as output
  pinMode(GREEN_PIN, OUTPUT); // set green pin as output
  pinMode(BLUE_PIN, OUTPUT); // set blue pin as output
}
void loop() {
  // turn on red
  digitalWrite(RED_PIN, HIGH);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BLUE_PIN, LOW);
  delay(1000); // wait for 1 second
```

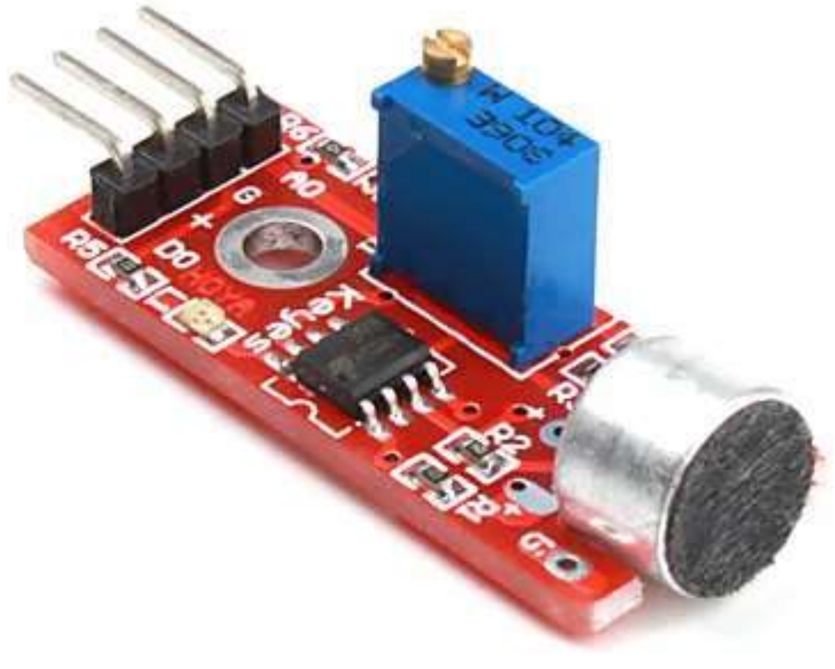
```
 // turn on green
  digitalWrite(RED_PIN, LOW);
  digitalWrite(GREEN_PIN, HIGH);
  digitalWrite(BLUE_PIN, LOW);
  delay(1000); // wait for 1 second

  // turn on blue
  digitalWrite(RED_PIN, LOW);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BLUE_PIN, HIGH);
  delay(1000); // wait for 1 second
}
```

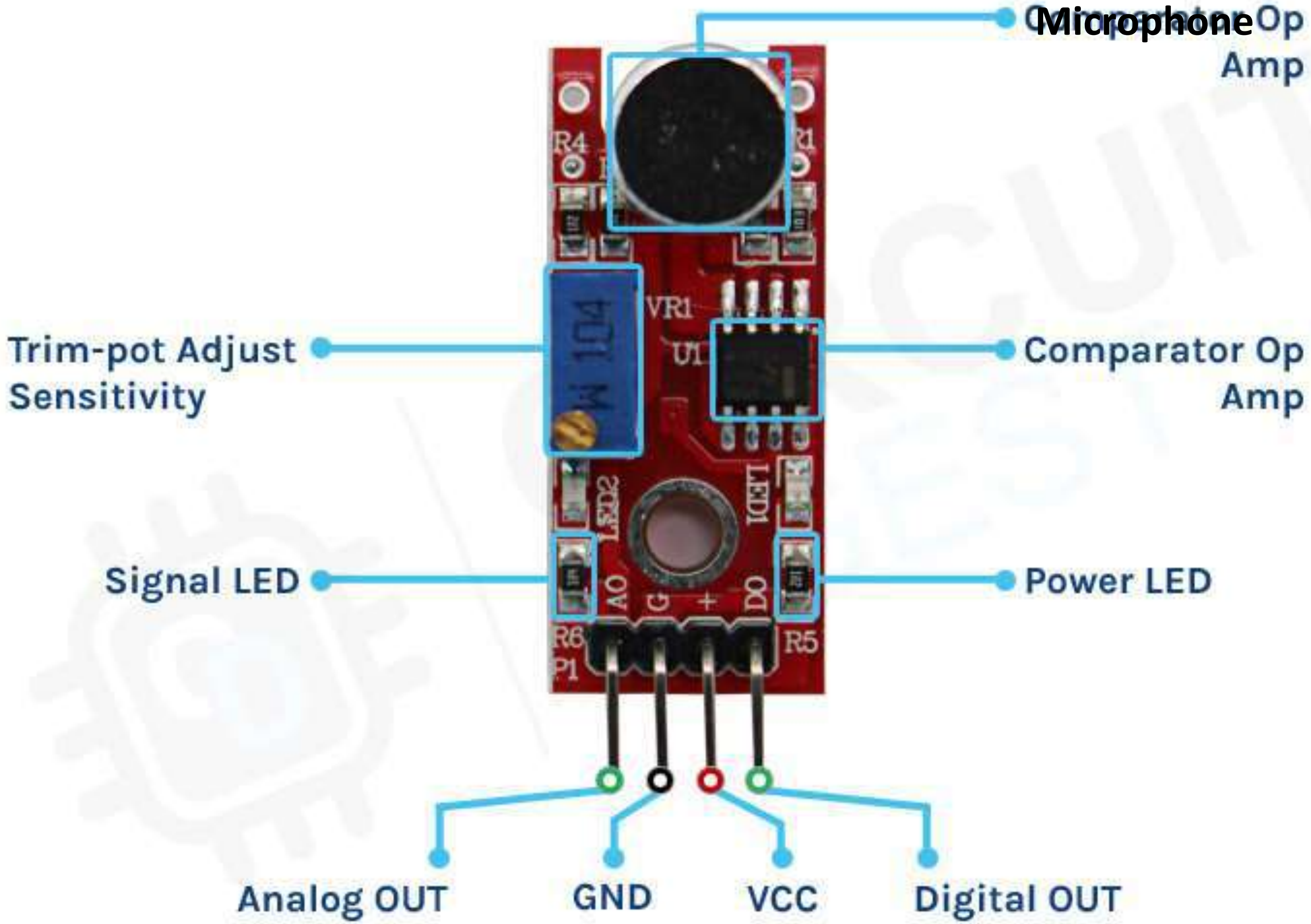
SOUND SENSOR

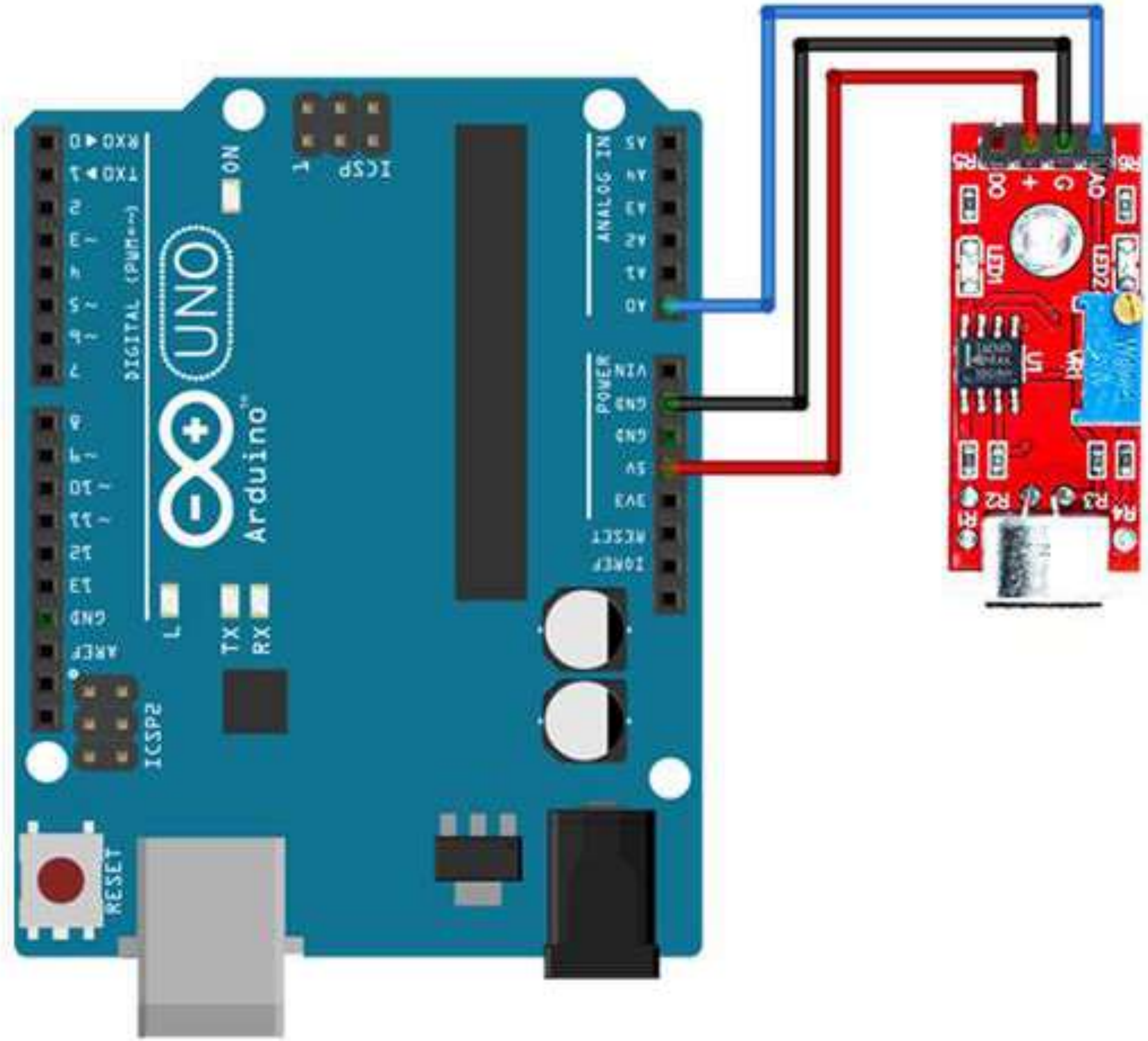
The sound sensor is a module that monitors and detects the sound signals like voice, claps, snaps, knocks, etc. It is also known as an acoustic sensor or sound detector. Used in various applications such as security systems, monitoring systems, radios, telephones, mobile phones, computers, home automation systems, consumer electronic appliances, etc These sensors are very simple to use

•It gives analog o/p signal

•Simply incorporates using logic modules on the input area




Microphone
Op
Amp
Trim-pot Adjust Sensitivity
Comparator Op Amp
Signal LED
Power LED
Analog OUT
GND
VCC
Digital OUT
R4
R1
VR1
U1
104
LED2
LED1
A0
G
+
D0
R6
P1
R5



DO
+
G
AO
A0
GND
5V

```
int led = 6;
int sound_analog = A0;

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


void loop(){
  int val_analog = analogRead(sound_analog);
  int voice = map(val_analog,0,1023,0,255);
  analogWrite(led,voice);
}
```

THANK YOU

FOR JOINING US