Workshop Basic Arduino
Class 2 – Sensors

MSc. David Velásquez Rendón

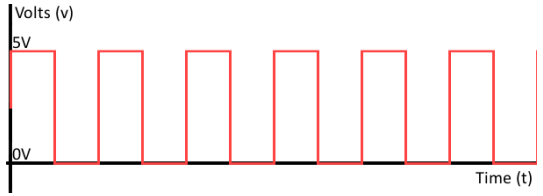
UNIVERSIDAD EAFIT®

Contents

1. Introduction
2. Digital sensors
3. Analog sensors
4. Specialized sensors

Introduction – Types of sensors

Digital



- 2 States (Binary):
  - "HIGH" or "$5V$" or "ON".
  - "LOW" or "$0V$" or "OFF".
- Follow the static discipline.
- Works as a switch.

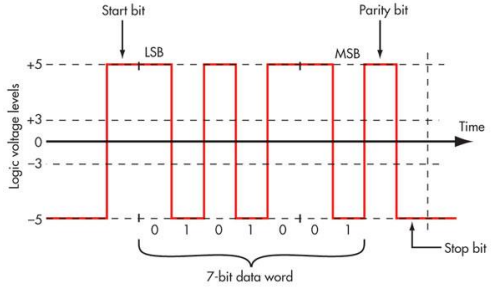
Arduino function: $digitalRead(PIN)$

Analog

- Multiple values (0 to 5V).
- Requires Analog to Digital Converter (ADC).
  - Arduino ADC has 10 bit.
  - Arduino ADC range is 0~1023.

Arduino function: $analogRead(PIN)$

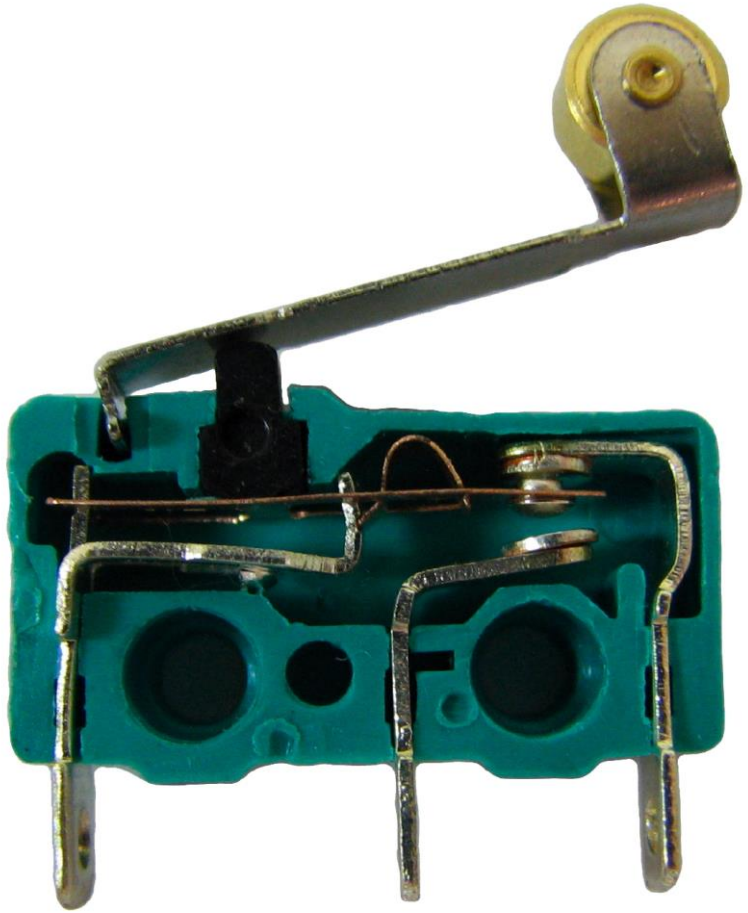
Specialized



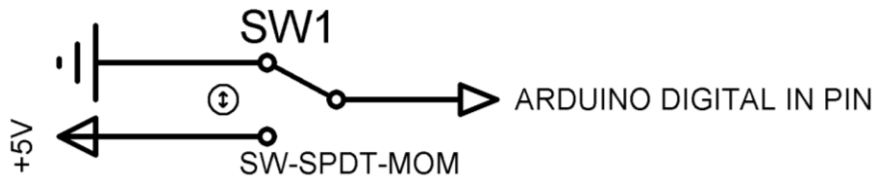
- Usually work as a digital sensor with **fast pulses**.
- The data is "**encoded**" in a bit stream.
- Commonly used with **libraries** to interpret data.
- Usually include a **microprocessor** handling encoding.
- Most used protocols: **UART**, **I2C**, **One-wire**.

Digital Sensors – Limit Switch

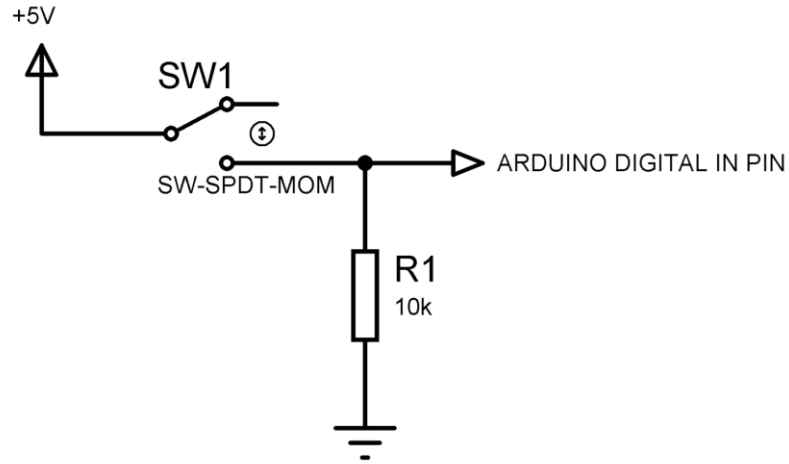
- Works as a button in SPDT configuration (Single Pole Dual Throw).
- Normally, it possess 3 terminals (Common, Normally Open, Normally Closed).
- Commonly used as Limit Switch for a path (security stops).



OPTION 1



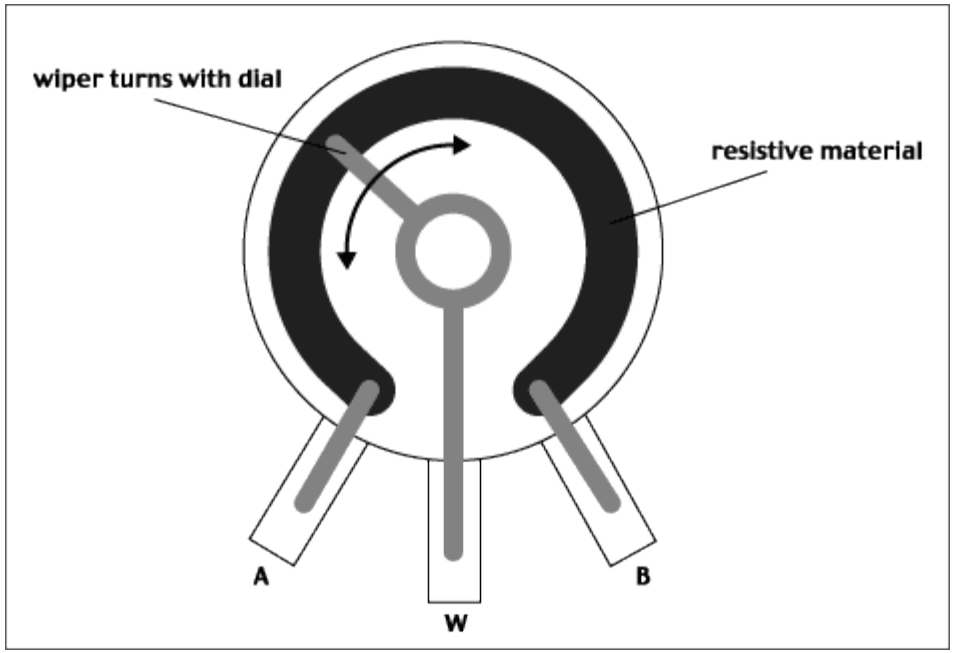
OPTION 2

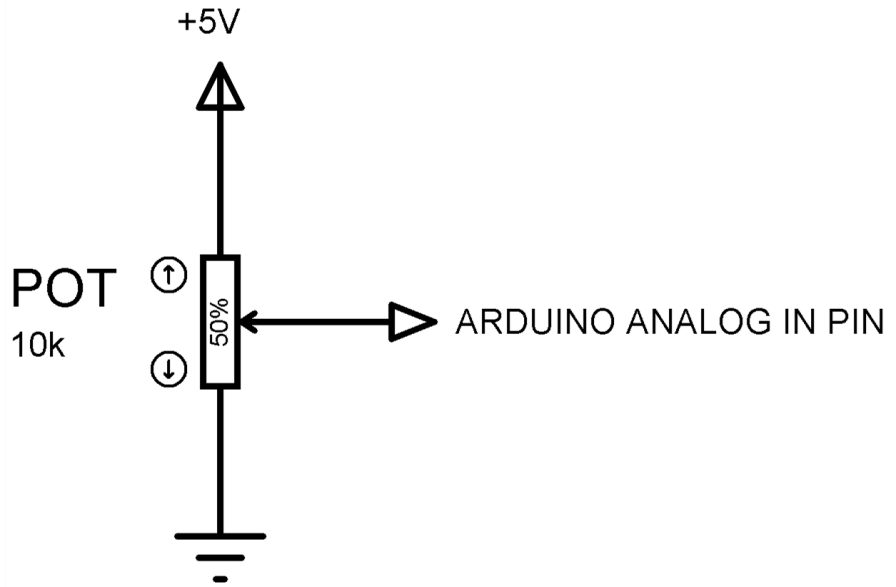


To use this sensor with Arduino, configure as a digital INPUT in the pin that it's connected. Use digitalRead to read the sensor.

Analog Sensors- Potentiometer

- Variable resistance.
- Allows the user to graduate certain variable.
- Can be used as angular sensors.
- Types:
  - Logarithmic: Less precision.
  - Linear: High precision.

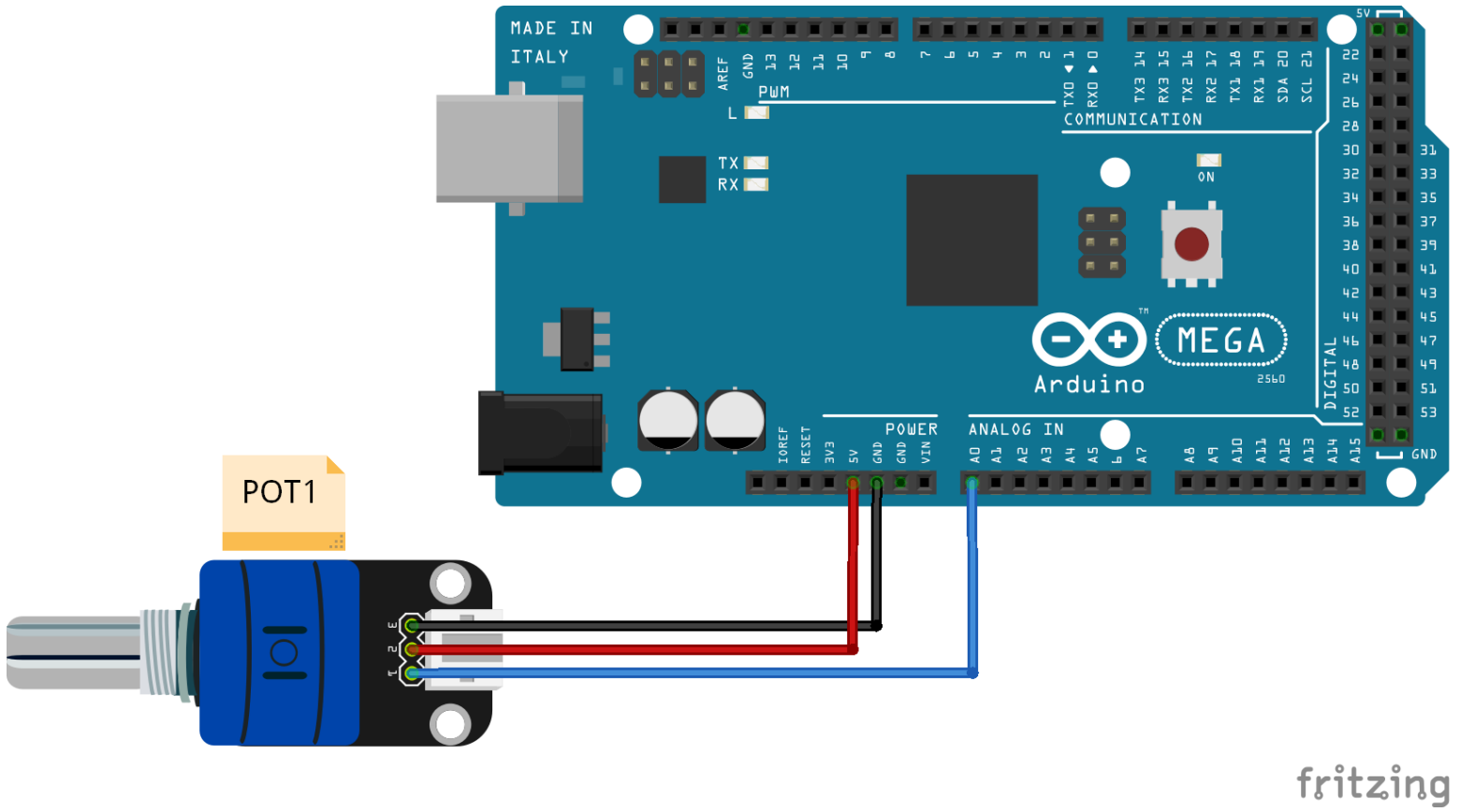




To use this sensor with Arduino, connect to an Analog Pin (Voltage will vary from 0V to 5V) and use the analogRead function

Example 2.1 - Potentiometer

- Do an Arduino program that monitors the value of the potentiometer and shows it on the Serial Monitor



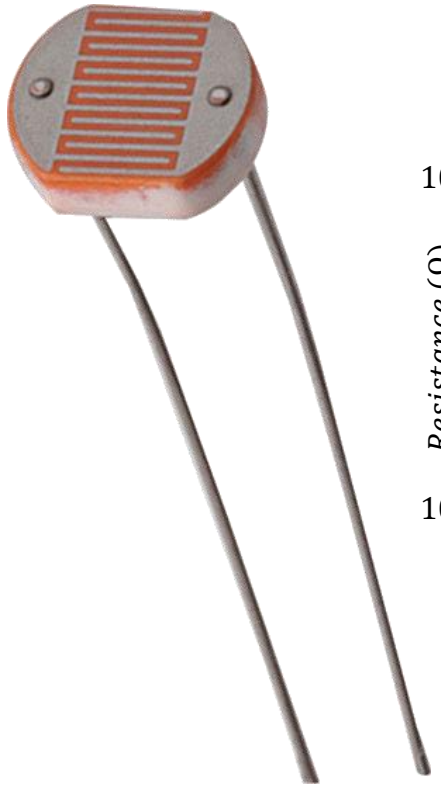
Example 2.1 - Potentiometer

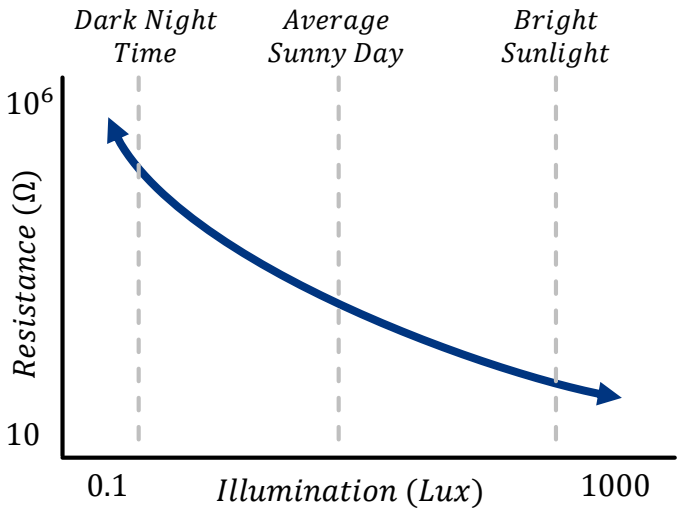

```
//I/O Pin Labeling
#define POT1 0  //Potentiometer POT1 connected to pin A0

//Variable declaration
unsigned int valuePOT = 0; //Variable to store the value of the potentiometer (valuePOT)

void setup() {
  //I/O Pin Configuration
  //Note: The analog inputs are not declared as INPUTS, they come like this as default

  //Communications
  Serial.begin(9600);  //Begin Serial communications with the computer using the Serial0
ports (TX0 RX0) and 9600 speed bauds rate
}

void loop() {
  valuePOT = analogRead(POT1); //Read the ADC value of POT1 pin and store it on valuePOT
  Serial.print("POT value: "); //Print result on Serial monitor
  Serial.println(valuePOT); //Print result on Serial monitor
}
```

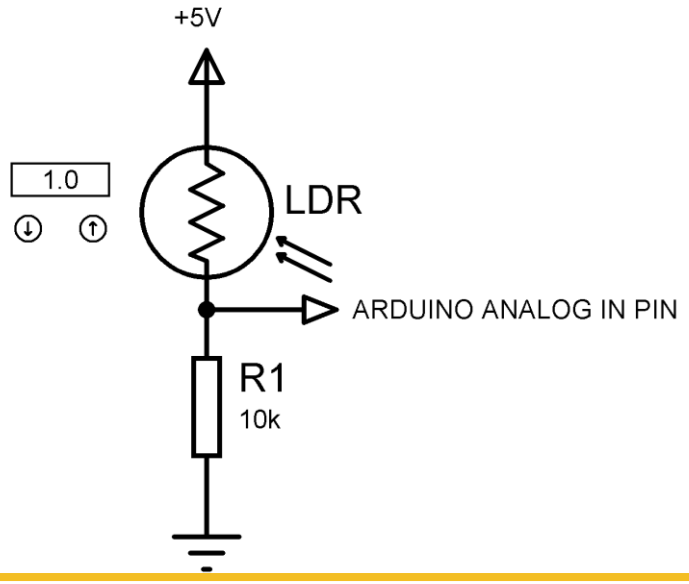
https://github.com/tidusdavid/arduino-workshop-paris/blob/master/Arduino%20Codes/E21_Potentiometer/E21_Potentiometer.ino

Analog Sensors- LDR

- Light Dependent Resistor (LDR).
- Changes its resistance depending on the light intensity.
  - Lowers its resistance if there is more light.
  - Increases its resistance if There is less light.
- Used for applications where the brightness level needs to be measured in order to be controlled (e.g. Smartphone screen brightness)



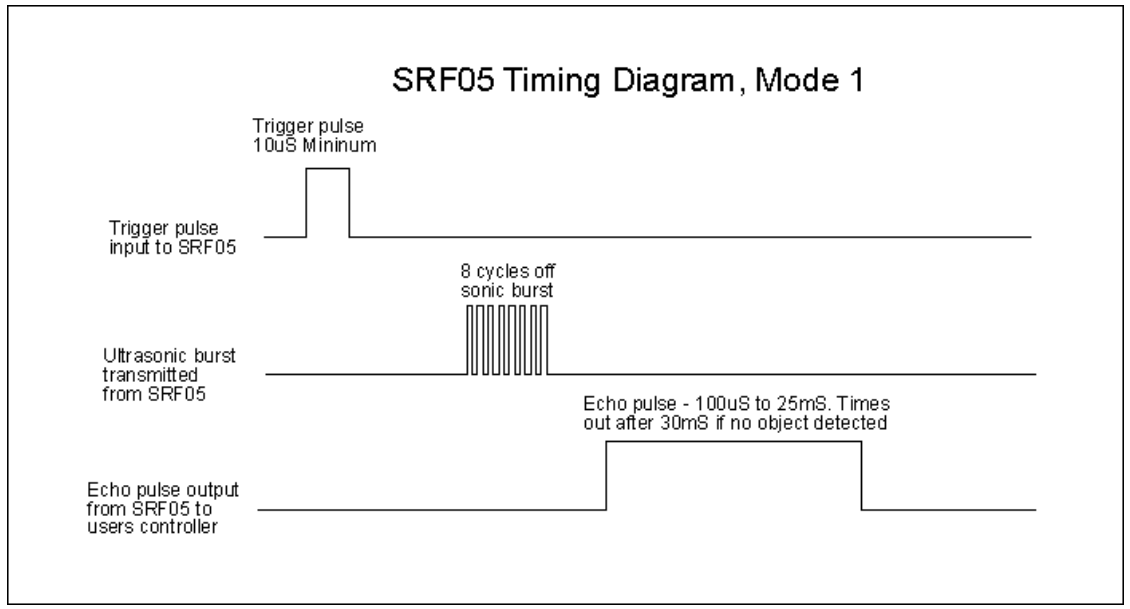




To use this sensor with Arduino, connect the sensor in voltage divider configuration (see picture). The output of the divider goes to an analog pin of the Arduino, where the analogRead function must be used. The voltage will vary depending on the resistance that is selected as Pull Down on the divider

Digital/Analog Sensor – Ultrasonic Sensor

- Used in order to **measure distance** to an object.
- They reach "large" distances (approximately 5 meters).
- Works as a **bat**.
- **Returns the time** (on microseconds) that the sound signal took to go to an object and return to the sensor.

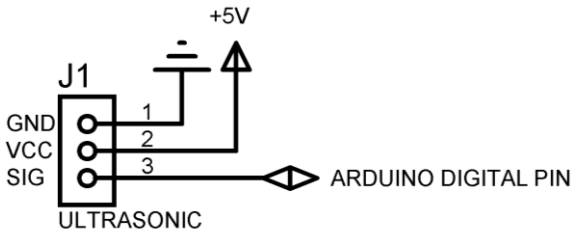


$$x(cm) = 343\frac{m}{s} \cdot \frac{100\ cm}{1\ m} \cdot \frac{t(\mu secs)}{2} \cdot \frac{1\ s}{10^6 \mu secs}$$

$$x(cm) \approx \frac{t(\mu secs)}{58}$$

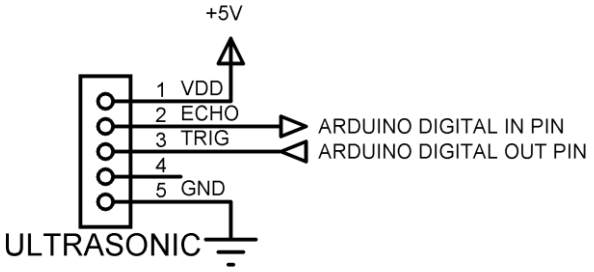
Use `pulseIn(PIN, STATE)` *function to compute this time*

Joined ECHO & TRIG Variation Type (SEEED)



To use this sensor with Arduino, the SIG pin must be connected to a digital pin of the Arduino and finally the VDD and GND pins must be supplied with a power supply.

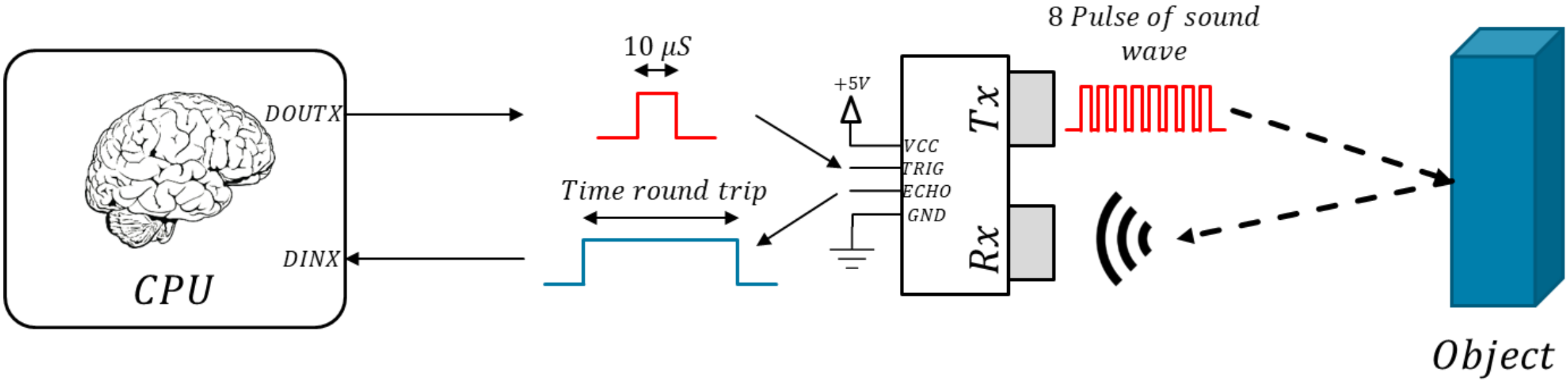
Separate ECHO & TRIG Variation Type (SFR05)



To use this sensor with Arduino, the ECHO pin must be connected to a digital input pin of the Arduino, the TRIG pin to a digital output pin and finally the VDD and GND pins must be supplied with a power supply.

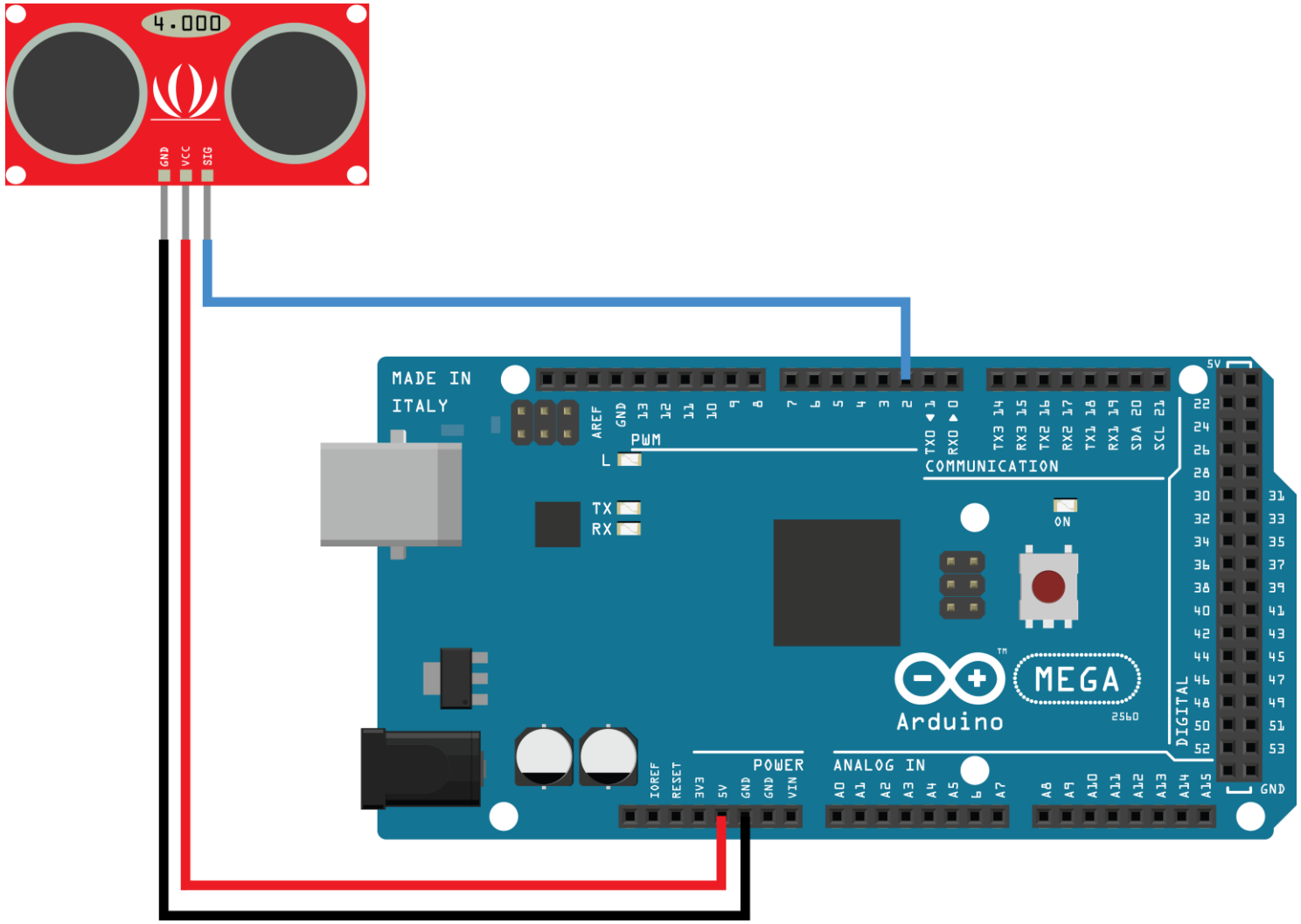
Digital/Analog Sensor – Ultrasonic Sensor

Working Principle



Example 2.2A – Ultrasonic Sensor (SEEED)

- Measure the distance in cm using an ultrasonic sensor.



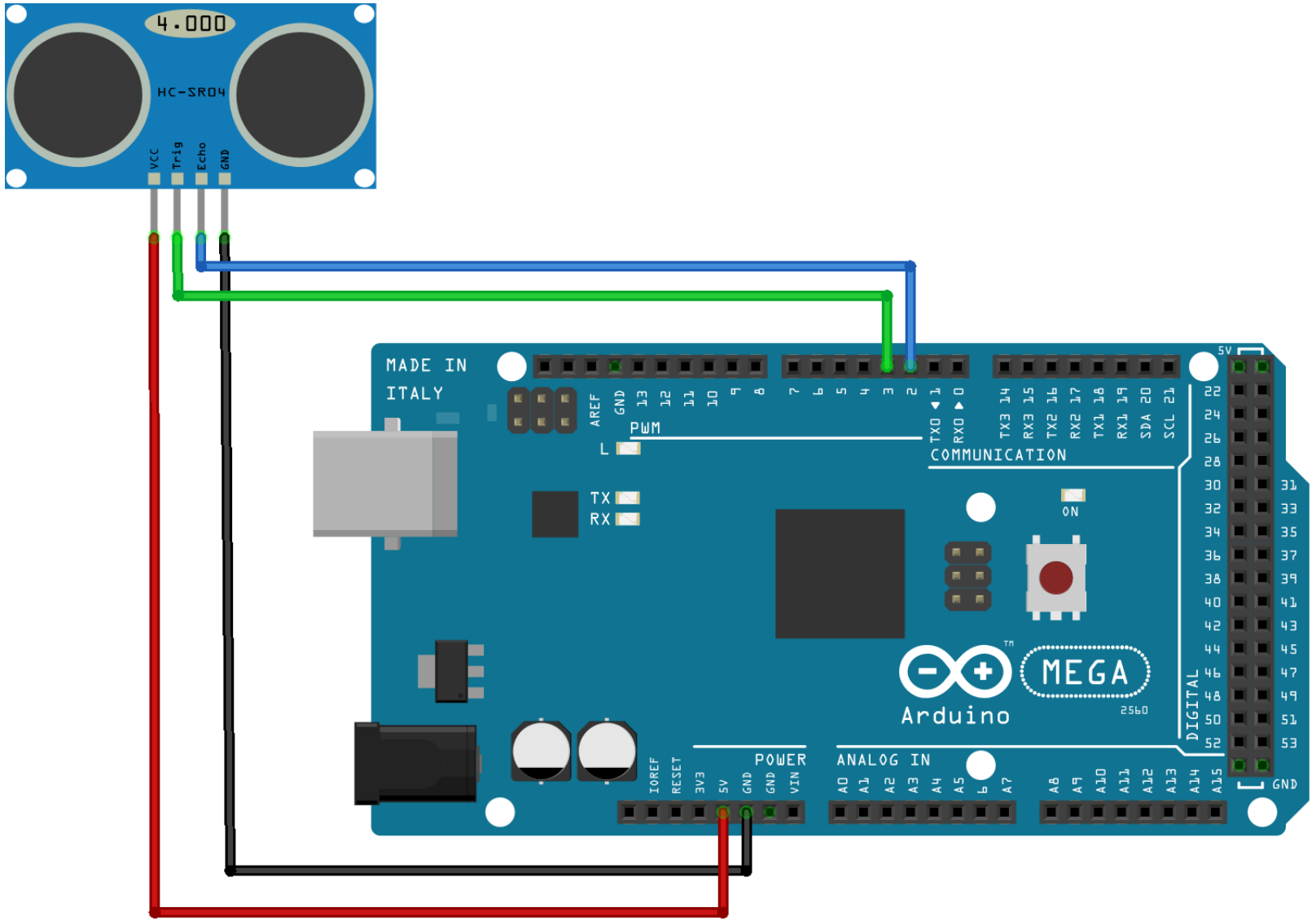
Example 2.2A – Ultrasonic Sensor (SEEED)


```
//I/O Pin Labeling
#define SIG 2 //Ultrasonic SIG pin connected to
Arduino pin 2

//Variable declaration
unsigned int distance = 0;  //Variable for
storing the value of the potentiometer (distance)

//Subroutines and functions
unsigned int ultraMeas(unsigned int SIGPIN) {
  delay(50);  //Delay of 50 ms before the next
ranging
  pinMode(SIGPIN, OUTPUT); //SIGPIN as input (for
transmitting)
  digitalWrite(SIGPIN, HIGH); //Turn ON the
SIGPIN for measuring the distance
  delayMicroseconds(10);  //Wait 10uSecs with the
TRIG ON
  digitalWrite(SIGPIN, LOW); //Turn OFF the
SIGPIN
  pinMode(SIGPIN, INPUT); //SIGPIN as input (for
measuring)
  return pulseIn(SIGPIN, HIGH)/58.0; //Return the
distance on centimeters
}
```

```
void setup() {
  //I/O Pin Definition
  pinMode(SIGPIN, OUTPUT); //SIGPIN as output

  //Physical Output Cleaning
  digitalWrite(SIGPIN, LOW); //Turn OFF SIGPIN

  //Communications
  Serial.begin(9600); //Begin Serial
Communications with the computer by the Serial 0
port (TX0 RX0) at 9600 bauds
}

void loop() {
  distance = ultraMeas(SIG);  //Measure distance
  Serial.print("Distance (cm): ");
  Serial.println(distance);
}
```

https://github.com/tidusdavid/arduino-workshop-paris/blob/master/Arduino%20Codes/E22A_Ultrasonic/E22A_Ultrasonic.ino

Example 2.2B – Ultrasonic Sensor (SFR05)



Example 2.2B – Ultrasonic Sensor (SFR)

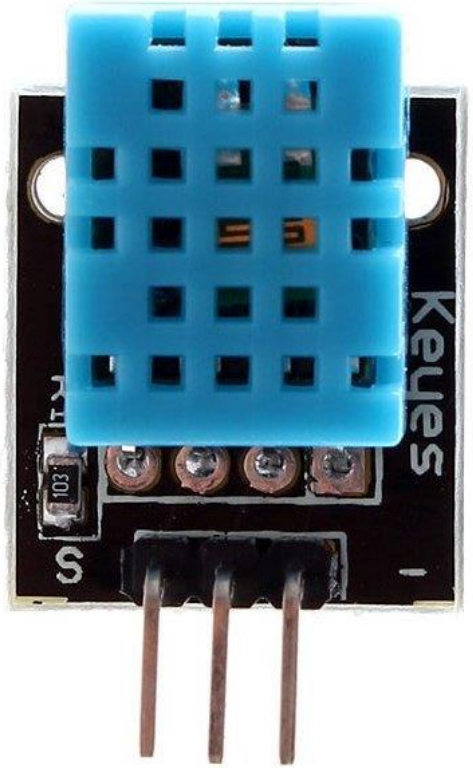

```
//I/O Pin Labeling
#define ECHO 2 //Ultrasonic ECHO pin connected to
Arduino pin 2
#define TRIG 3 //Ultrasonic TRIG pin connected t
Arduino pin 3

//Variable declaration
unsigned int distance = 0;  //Variable for storing
the value of the potentiometer (distance)

//Subroutines and functions
unsigned int ultraMeas(unsigned int ECHOPIN,
unsigned int TRIGPIN) {
  delay(50);  //Delay of 50 ms before the next
ranging
  digitalWrite(TRIGPIN, HIGH); //Turn ON the TRIG
for measuring the distance
  delayMicroseconds(10);  //Wait 10uSecs with the
TRIG ON
  digitalWrite(TRIGPIN, LOW); //Turn OFF the TRIG
  return pulseIn(ECHOPIN, HIGH)/58.0; //Return the
distance on centimeters
}
```

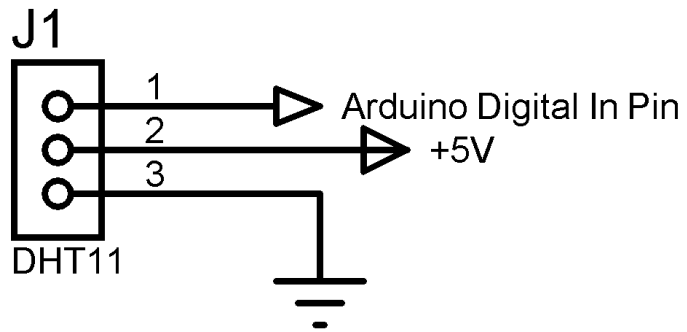
```
void setup() {
  //I/O Pin Definition
  pinMode(ECHO, INPUT); //ECHO Pin as input
  pinMode(TRIG, OUTPUT);  //TRIG Pin as output

  //Physical Output Cleaning
  digitalWrite(TRIG, LOW);

  //Communications
  Serial.begin(9600); //Begin Serial
Communications with the computer by the Serial 0
port (TX0 RX0) at 9600 bauds
}

void loop() {
  distance = ultraMeas(ECHO, TRIG);  //Measure
distance
  Serial.print("Distance (cm): ");
  Serial.println(distance);
}
```

https://github.com/tidusdavid/arduino-workshop-paris/blob/master/Arduino%20Codes/E22B_Ultrasonic/E22B_Ultrasonic.ino

Specialized sensors – DHT11

- Allows to measure temperature and humidity.
- Can be used for weather telemetry.
- Very useful for Green houses.
- Sends a data stream via One-Wire protocol that can be interpreted with a Library.

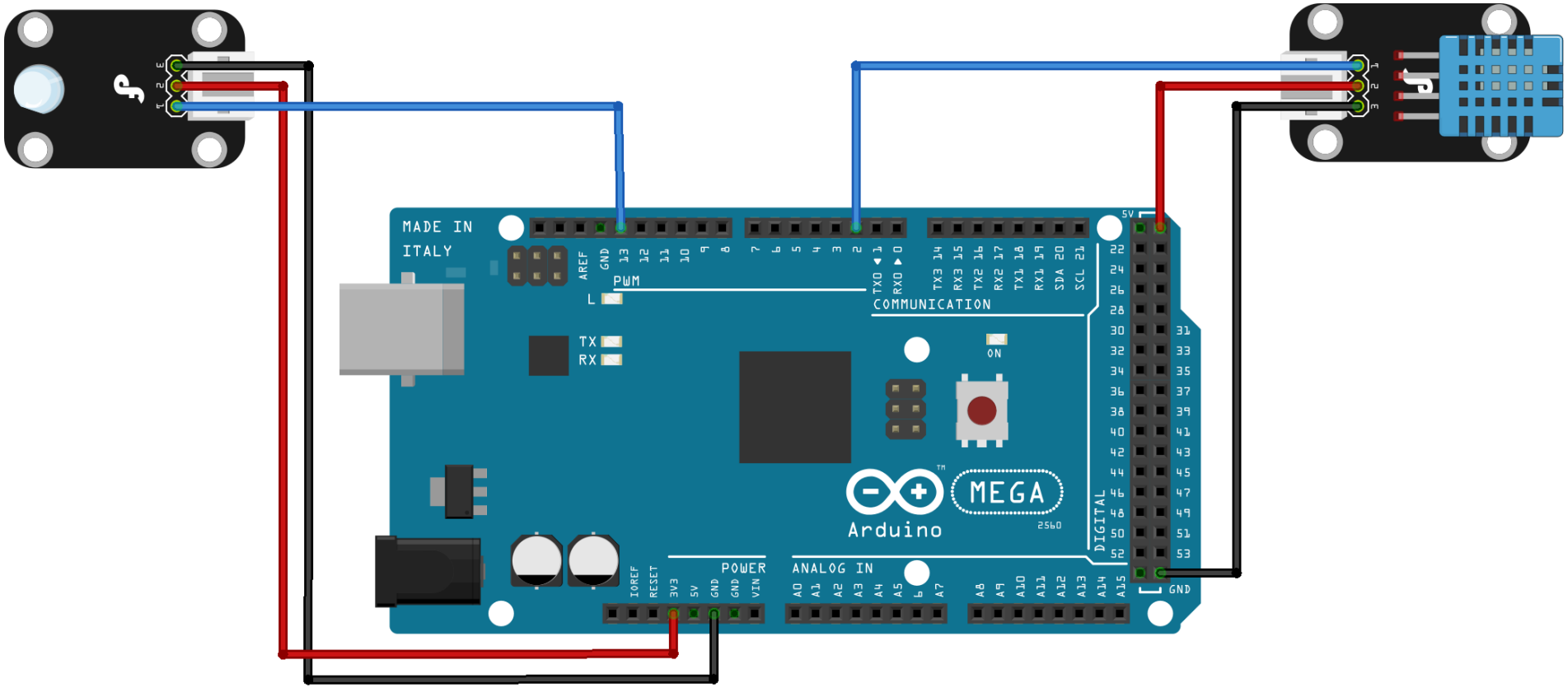




For using this sensor with Arduino, a library that allows to interpret data stream from sensor must be used

Example 2.3 – DHT11

- Do a program on Arduino that monitors the temperature and humidity of the DHT11 and prints each values on the Serial Monitor every second. If the temperature rises 25 °C, turn a LED L1.



Example 2.3 – DHT11

```
//Library Declaration
#include <DHT.h>

//I/O Pin Labeling
#define L1 13 //LED L1 connected on pin 13
#define DHTPIN 2 //DHT11 sensor connected to pin
2

//Constants declaration
const int Tmax = 25;  //Max temperature constants
initiliazed
//on 25 °C.
//Variable declaration
float temperature = 0;  //Variable to store the
temperature
float humidity = 0;  //Variable to store humidity

//Library Vars
#define DHTTYPE DHT11 //Use DHT11 sensor variant
DHT dht(DHTPIN, DHTTYPE); //DHT object var

//Subroutines & Functions
void readDHT() {
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
}
```

```
//Configuration
void setup() {
  //Pin Configuration
  pinMode(L1, OUTPUT);  //LED L1 as a digital
output

  //Physical Outputs Cleaning
  digitalWrite(L1, LOW);  //Turn off L1

//Communications
  Serial.begin(9600); //Begin Serial
Communications with the computer by the Serial 0
port (TX0 RX0) at 9600 bauds
  dht.begin(); //Initialize communications with
DHT11 sensor
}

//Run-time
void loop() {
    readDHT();
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.print(" Humidity: ");
    Serial.println(humidity);
    if (temperature >= Tmax) {
      digitalWrite(L1, HIGH);
    }
    delay(1000);
}
```

https://github.com/tidusdavid/arduino-workshop-paris/blob/master/Arduino%20Codes/E23_DHT11/E23_DHT11.ino

FIN

Thanks!