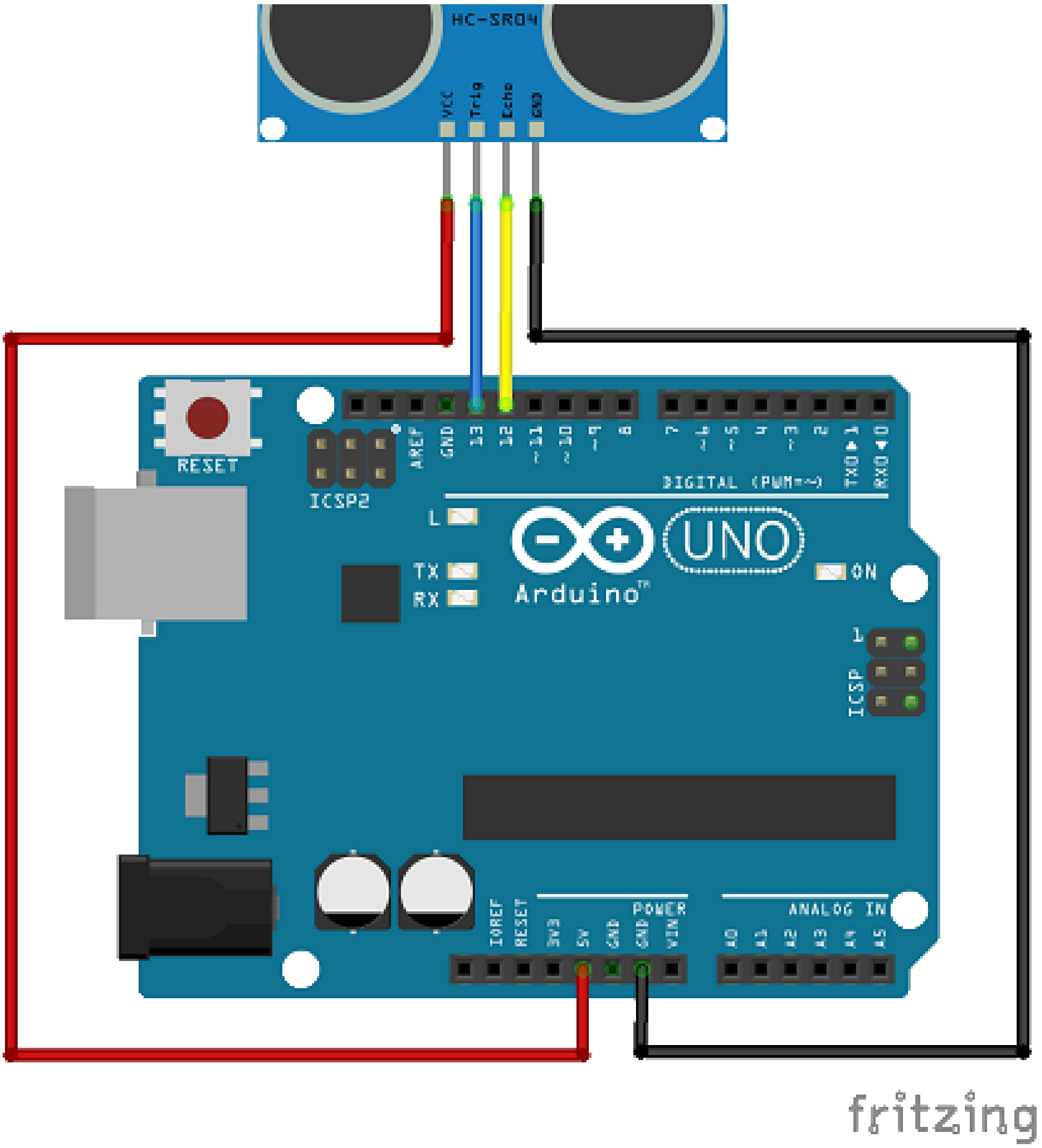
< foreach

HC-SR04 Ultrasonic Sensor

Materials to be used:

- Arduino UNO as Development Board
- HC-SR04 Ultrasonic Sensor
- Jumper wires

Schematic:



Theory:

HC-SR04 Ultrasonic Sensor has a range from 2cm to 400cm, with up to 3mm of accuracy.

That sensor includes an Ultrasonic Transmitter and a Receiver.

travel back after bouncing from the obstacle is measured by the *pulseIn().*
Based on that duration the actual distance to the obstacle is calculated and printed on the *Serial Monitor of the Arduino IDE*

Procedure:

- Connect the components as shown in schematic.
- Upload the code.
- After uploading is done, go to **Tools > Serial Monitor** and you can see the distance values at 500ms intervals as set in the code.

Code:

```
const int trigPin = 13;
const int echoPin = 12;
long duration, distance;
void setup(){
  // initialize serial communication:
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}


void loop(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  if (distance >= 200 || distance <= 0){
    Serial.println("Out of range");
   }
   else {
     Serial.print(distance);
     Serial.println(" cm");
   }
   delay(500);
}
```

This is a summary of how an Ultrasonic Sensor operates:

First with your code you send a pulse to the Ultrasonic Sensor module through the trigPin.

The pulse that's sent through the trigPin is an electrical signal. Therefore it needs to be converted into an ultrasonic signal before being sent in the air.
That is why at the end of the electrical signal, the ultrasonic sensor module generates and sends in the air 8 bursts of 40 kHz pulses as ultrasonic signal to measure distance.

At exactly the moment the first burst is sent, the echoPin sets itself HIGH. And according to the line of code **pulseIn(echoPin, HIGH);**
When echoPin is in HIGH the microcontroller starts timing.

When the first pulse of the 40 kHz bounces onto the object and gets back to the receiver of the sensor module, the echoPin sets

Then the program continues running.

How do we calculate the distance between the Ultrasonic Sensor Module and an Object?

pulseIn(echoPin, HIGH) reads a pulse on echoPin continuously. And when echoPin goes HIGH, pulseIn() starts timing. Once echoPin goes LOW pulseIn() stops timing and returns the length of the pulse in microseconds (µs).

The Pace of Sound ps = 1 / Speed of Sound
Speed of sound is 343.5m/s
Let's change the unit of measure from m/s to cm/µs
1m/s = 100cm/1000000µs
Or 1m/s = cm/10000µs
Or 1m/s = 0.0001cm/µs

The speed of sound ss = 343.5m/s can now be converted to the pace of sound ps as follows:
ss= 343.5×0.0001 cm/µs
or
ss = 0.03435cm/µs

Pace of Sound (ps) is expressed as the inverse of Speed of Sound (ss) expressed in cm/µs
ps = 1 / 0.03435

< foreach

Back to the duration, remember it's returned by pulseIn() and it's in measured µs

distance = (duration[µs]/ 2) ÷ 29.1[µs/cm];

Example:

Let's say it took 20µs for an ultrasonic signal to leave the transmitter of the ultrasonic sensor module, hit an object, and return to the receiver of the ultrasonic sensor module.
What's the distance between the sensor module and the object.

distance = (1164µs/2)/29.1µs/cm
distance = 582µs/29.1µs/cm
distance = 20cm