

CSE423: Embedded System Summer-2020



Work with PIR Sensor and Ultrasonic Sensor



Today's Lecture



- ❑ What is PIR Sensor
- ❑ Application of PIR Sensor
- ❑ How to control PIR Sensor



- ❑ What is Relay
- ❑ Application of Relay
- ❑ How to use Relay to Real Life Appliances



PIR Sensor



- ❑ **P**assive **I**nfra **R**ed (PIR) sensors can detect movement of objects that radiate IR light (like human bodies). Therefore, using these sensors to detect human movement or occupancy in security systems is very common. Initial setup and calibration of these sensors takes about 10 to 60 seconds.
- ❑ The HC-SR501's infrared imaging sensor is an efficient, inexpensive and adjustable module for detecting motion in the environment. The small size and physical design of this module allow you to easily use it in your project.
- ❑ The output of PIR motion detection sensor can be connected directly to one of the Arduino (or any microcontroller) digital pins. If any motion is detected by the sensor, this pin value will be set to "1". The two potentiometers on the board allow you to adjust the sensitivity and delay time after detecting a movement.

PIR Sensor



- ❑ PIR modules have a passive infrared sensor that detects the occupancy and movement from the infrared radiated from human body.
- ❑ You can use this module in security systems, smart lighting systems, automation, etc. There are different PIR modules available in the market, but all of them are basically the same.
- ❑ They all have at least a Vcc pin, GND pin and digital output. In some of these modules, there is a ball like a lens on the sensor that improves the viewing angle.

Application of PIR Sensor

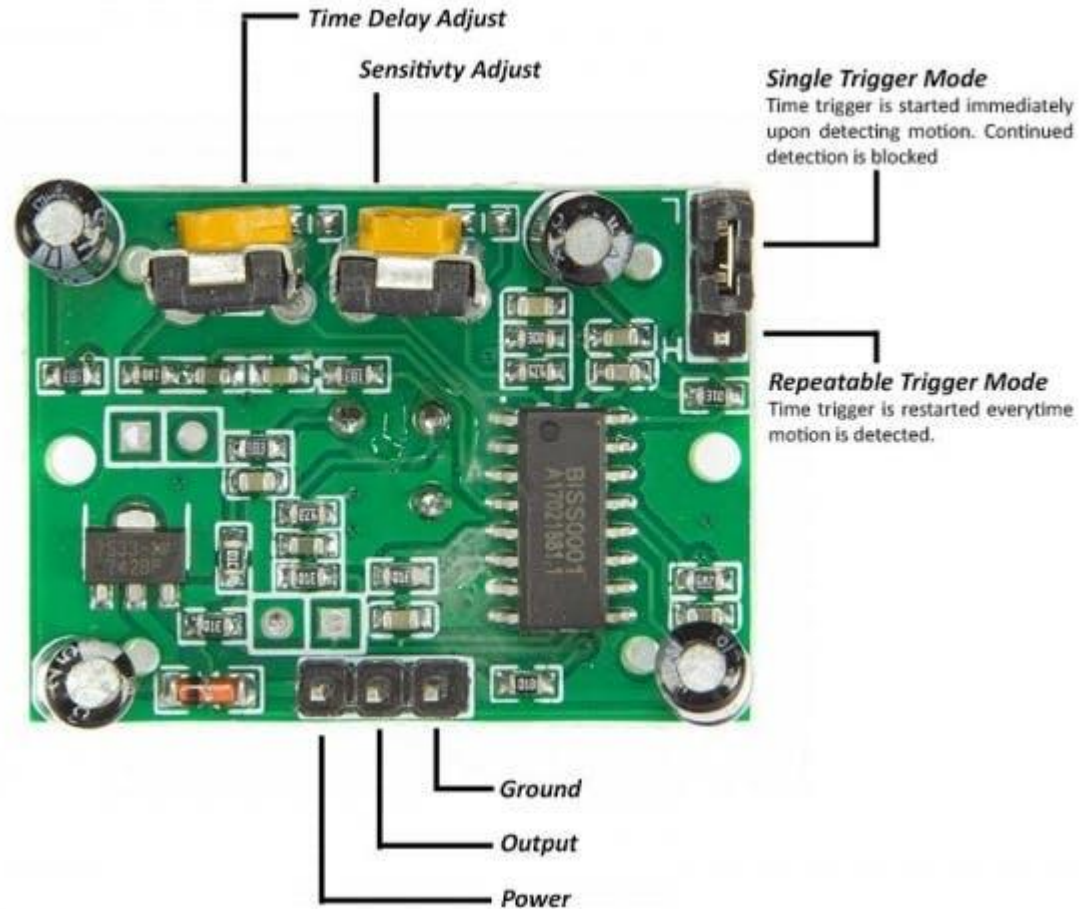


There is a jumper behind this module. If you move the jumper to **L** position, the sensor will 'toggle' (**change state**) whenever motion is detected. This is unlikely to be of much use in a practical applications. This mode is called non-triggering or Single Triggering mode.

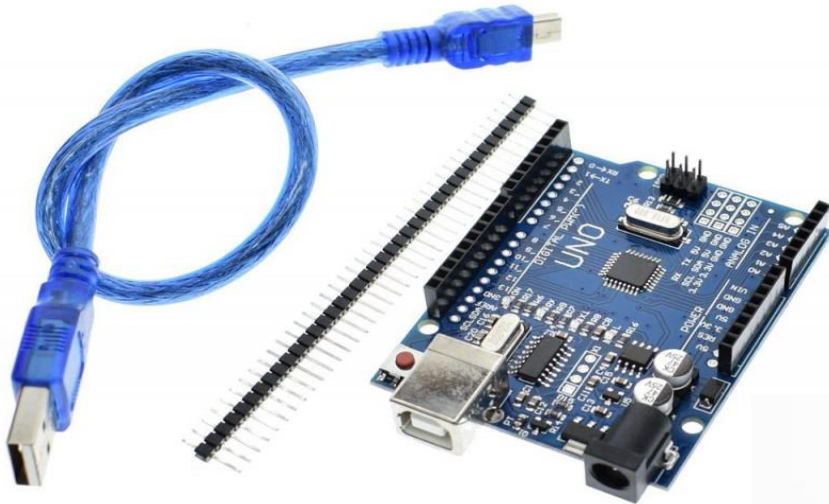
Moving the jumper to the **H** position will result in the more usual sensor logic. The sensor will turn on when motion is detected and turn off a while after the last motion is detected. This sensor will reset the timer (which would otherwise turn the output off) each time motion is detected; this would be applicable, for example, for room occupancy lighting control where you don't want the lights to blink off while the unit resets. This is called **Retriggering** mode. (or repeatable trigger mode).

There are also **two potentiometers** behind this module. By changing the SENSITIVITY potentiometer, you can reduce or increase the sensitivity of the sensor (clockwise increase), and also by changing TIME potentiometer the output delay after movement detection will be changed.

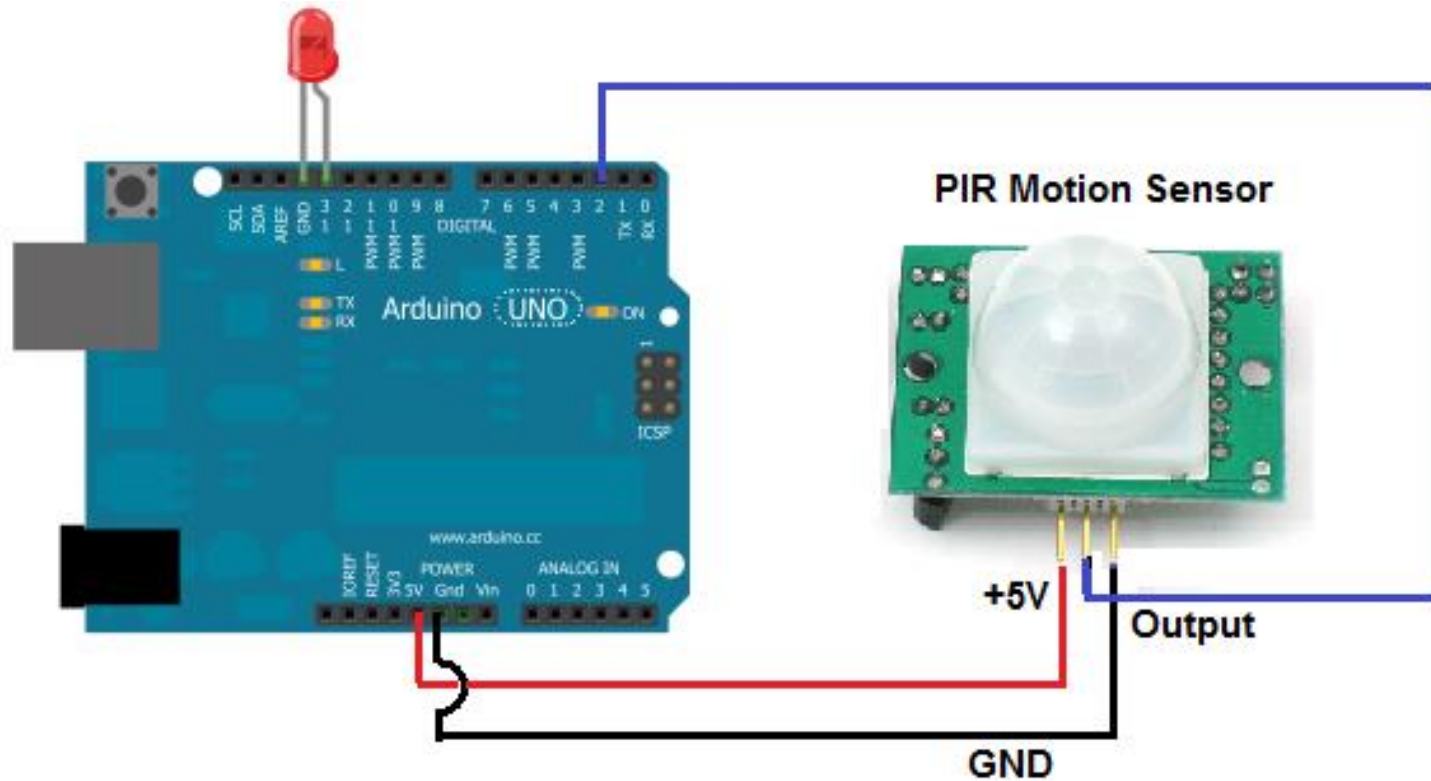
PIN Layout of PIR



Required Components



Connection Diagram



Code for PIR Sensor



```
void setup()
{
  pinMode(2,INPUT);

  // Connected with PIR Sensor

  pinMode(13,OUTPUT);
  // Connected with LED Output

  digitalWrite(12,LOW);
  Serial.begin(9600);
  Serial.print("Calibrating Sensor...");
  for (int i=0;i<30;i++)
  {
    Serial.print(".");
    delay(1000);
  }
  Serial.println("Calibration done");
  Serial.println("Sensor Activated");
}
```

```
void loop()
{
  int r=digitalRead(2);
  if (r==HIGH)
  {
    digitalWrite(13,HIGH);
    delay(5000);
  }
  else
  {
    digitalWrite(13,LOW);
  }
}
```

Is it working?



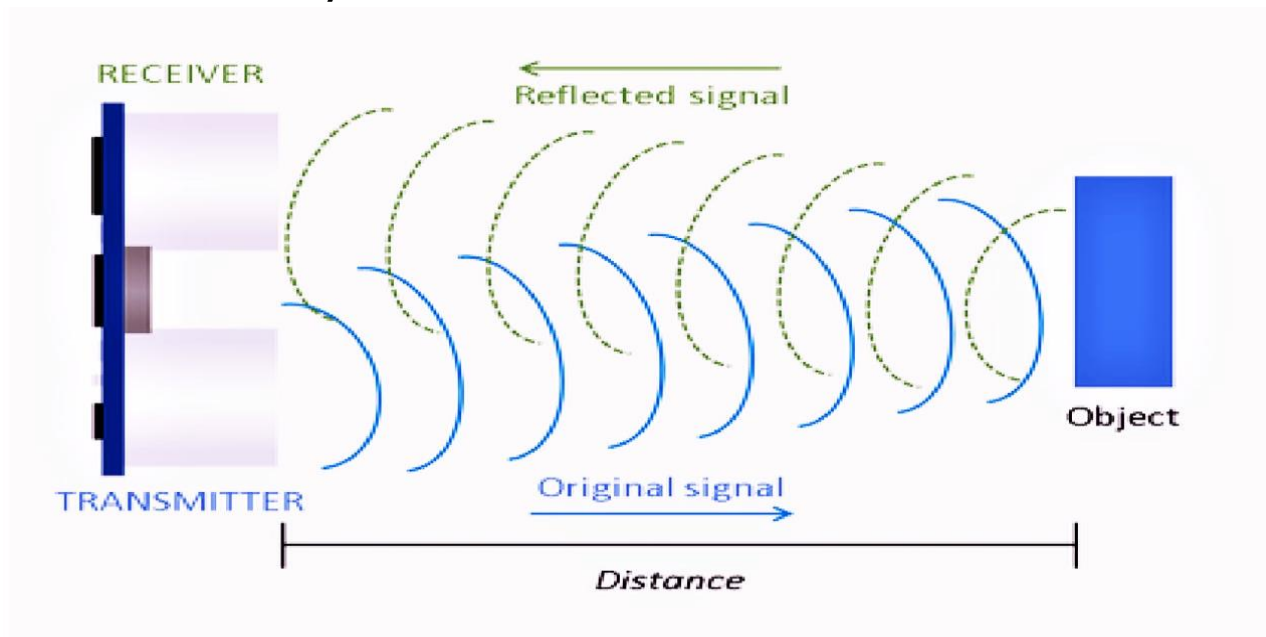
Ultrasonic Sensor



Ultrasonic Sensor







Ultrasonic Sensor HC-SR04 is a sensor that can measure distance. It emits an ultrasound at 40 000 Hz (40kHz) which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.



PIN Layout of Ultrasonic Sensor



The configuration pin of HC-SR04 is VCC (1), TRIG (2), ECHO (3), and GND (4). The supply voltage of VCC is +5V and you can attach TRIG and ECHO pin to any Digital I/O in your Arduino Board.

			
0.3 CM	<15'	<2MA	2-450CM
RESOLUTION	ANGLE	CURRENT	DETECTION RANGE



- 1. VCC**
- 2. TRIG**
- 3. ECHO**
- 4. GND**

Required Components



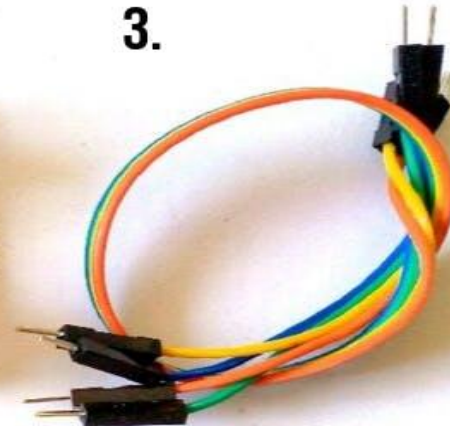
1.



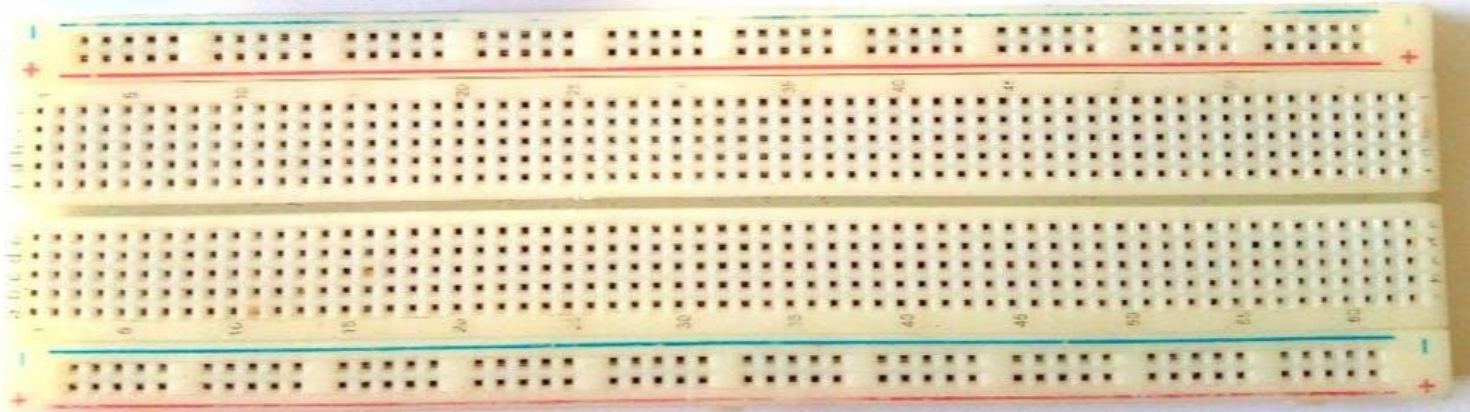
2.



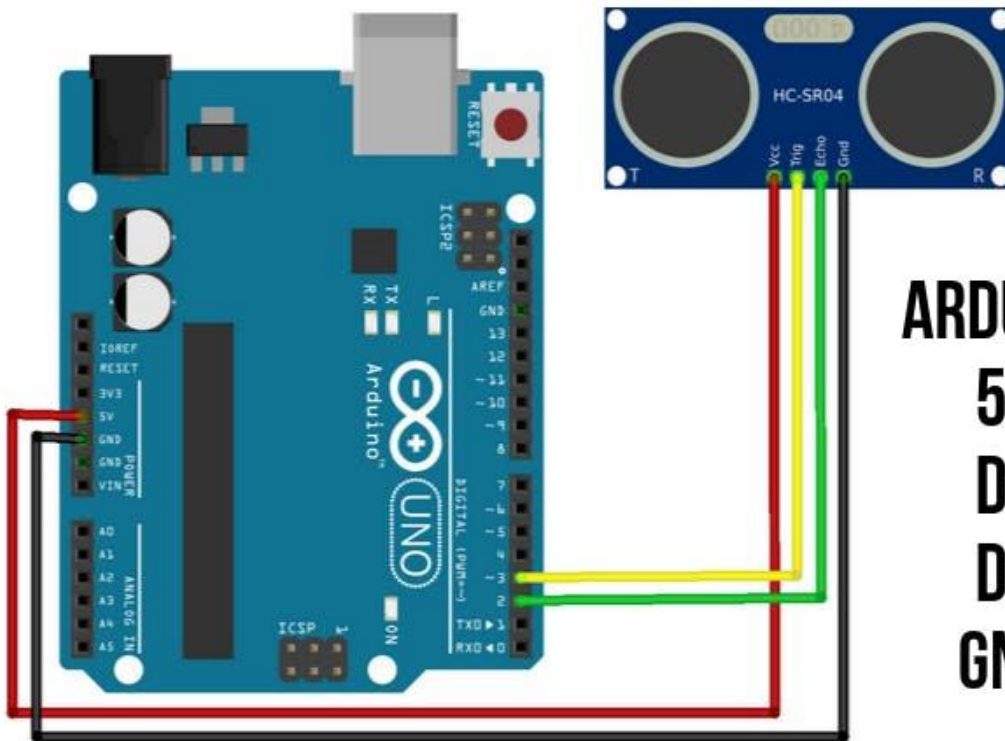
3.







4.



Connection Diagram



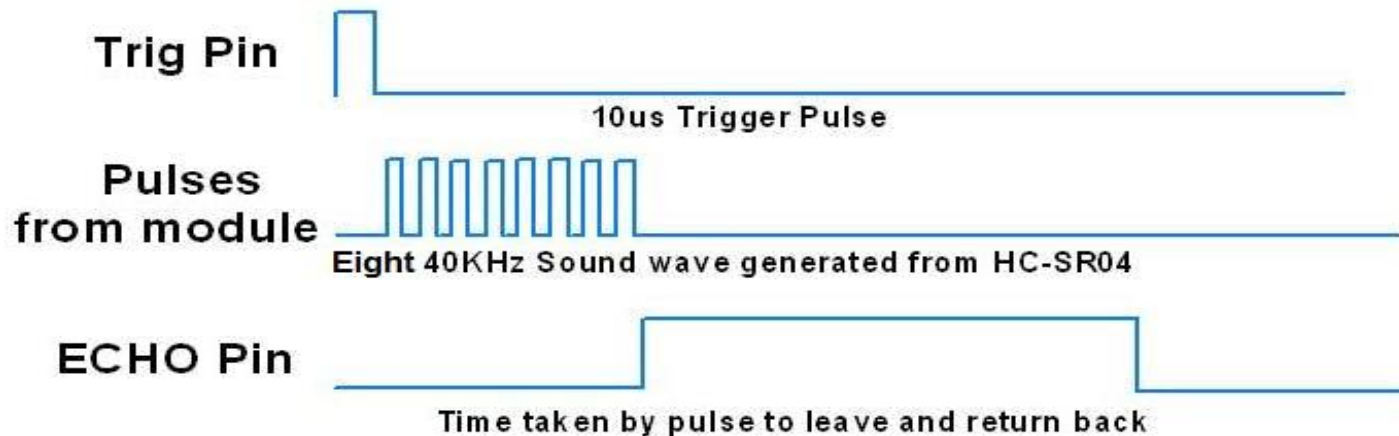
ARDUINO	>>	HC-SR04
5V		VCC
D2		ECHO
D3		TRIG
GND		GND

Working principle explained before coding:



- ❑ **Timing Diagram:** In order to generate the ultrasound we need to set the Trigger Pin on a High State for $10\ \mu\text{s}$. That will send out an 8 cycle sonic burst which will travel at the speed of sound and it will be received in the Echo Pin. The Echo Pin will output the time in microseconds the sound wave traveled.

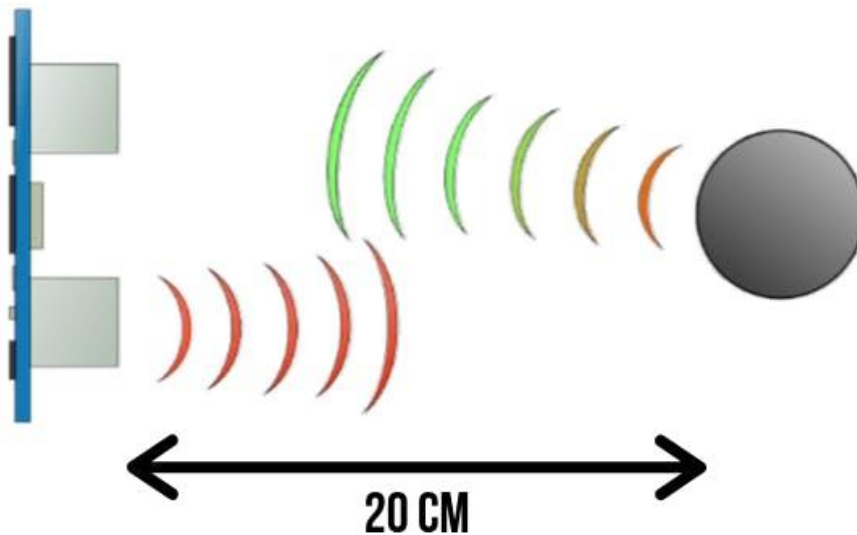
Ultrasonic HC-SR04 module Timing Diagram



Working principle explained before coding:



- ❑ **Calculation of distance:** For example, if the object is 20 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/ μ s the sound wave will need to travel about 588 microseconds. But what you will get from the Echo pin will be double that number because the sound wave needs to travel forward and bounce backward. So in order to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034



SPEED OF SOUND:

$$v = 340 \text{ m/s}$$

$$v = 0.034 \text{ m/s}$$

TIME = DISTANCE/SPEED

$$t = s/v = 20/0.034$$

$$= 588 \text{ us}$$

$$s = t \times 0.034/2$$

Steps to be followed before coding:



Step-1: For the programming code, first we need to define the Trigger Pin and Echo Pin that connected to Arduino board. In this project EchoPin is attached to D2 and TrigPin to D3.

Step-2: Define variables for the distance (int) and duration (long).

Step-3: In the loop first you have to make sure that the trigPin is clear so we have to set that pin on a LOW State for just 2 μ s.

Step-4: Set the trigPin on HIGH State for 10 μ s using the *pulseIn()* function for generating the ultrasound wave.

Steps to be followed before coding:



Step-5: we have to you have to read the travel time and put that value into the variable "duration". This function has 2 parameters, the first one is the name of the echo pin and for the second one you can write either HIGH or LOW. In this case, HIGH means that the *pulseIn()* function will wait for the pin to go HIGH caused by the bounced sound wave and it will start timing, then it will wait for the pin to go LOW when the sound wave will end which will stop the timing. At the end the function will return the length of the pulse in microseconds.

Step-6: For getting the distance we will multiply the duration by 0.034 and divide it by 2 as we explained this equation previously. At the end we will print the value of the distance on the Serial Monitor.

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

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.println("Ultrasonic Sensor Test");
  Serial.println("with Arduino UNO R3");
}
```

Code



```
void loop()
{
    digitalWrite(trigPin, LOW);    // Clears the trigPin condition
    delayMicroseconds(2);

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);        // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
}
```