LAB - 10

Interfacing PIR Motion Sensor with Node

PIR Motion Sensor – (HC-SR501 Module) Overview

All objects including human bodies emit heat in the form of infrared radiation if their temperature is above Absolute Zero (0 Kelvin / -273.15 °C). The radiation emitted will be more if the objects are hotter. Passive infrared (PIR) sensor is used to detect various levels of Infrared Radiations. It basically consists of two main parts: A Pyroelectric Sensor and A special lens called Fresnel lens which focuses the infrared signals onto the pyroelectric sensor.

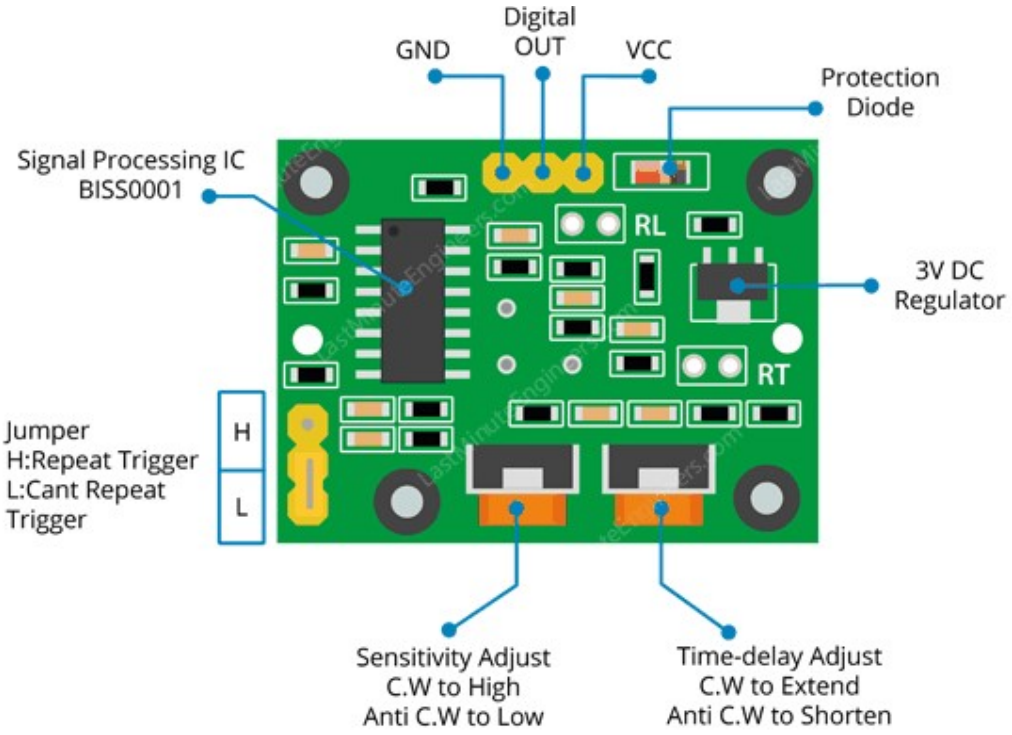
A Pyroelectric Sensor actually has two rectangular slots in it made of a material that allows the infrared radiation to pass. Behind these, are two separate infrared sensor electrodes, one responsible for producing a positive output and the other a negative output. The reason for that is that we are looking for a change in IR levels and not ambient IR levels. The two electrodes are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

When the sensor is idle, i.e. there is no movement around the sensor; both slots detect the same amount of infrared radiation, resulting in a zero output signal.

But when a warm body like a human or animal passes by; it first intercepts one half of the PIR sensor, which causes a positive differential change

between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. The corresponding pulse of signals results in the sensor setting its output pin high.

HC-SR501 PIR sensor has three output pins VCC, Output and Ground as shown in the diagram below. It has a built-in voltage regulator so it can be powered by any DC voltage from 4.5 to 12 volts, typically 5V is used. Other than this, there are a couple options you have with your PIR. Let's check them out.

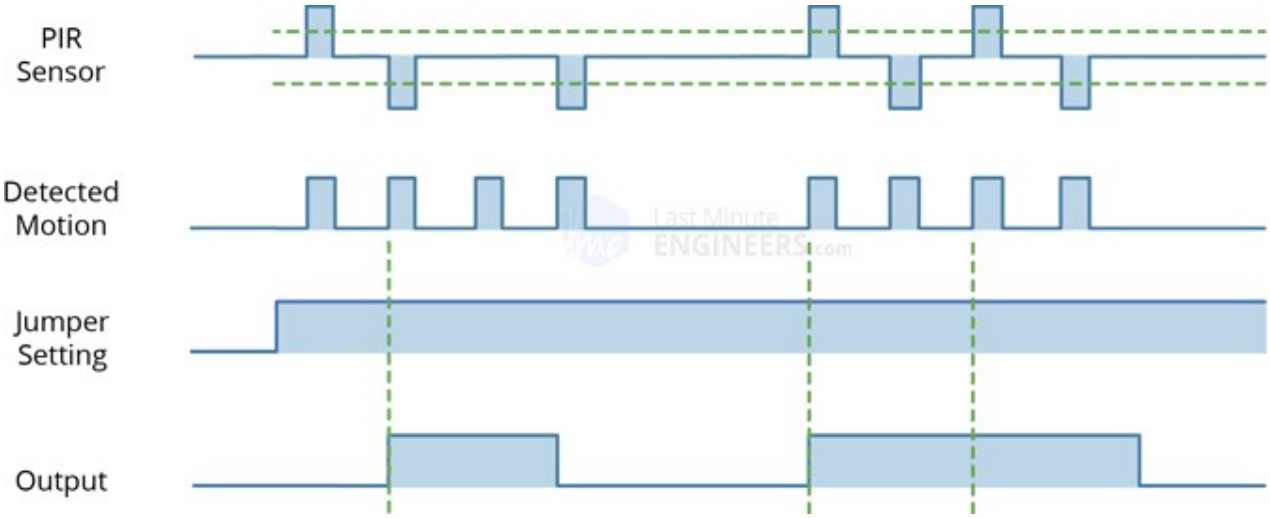


There are two potentiometers on the board to adjust a couple of parameters:

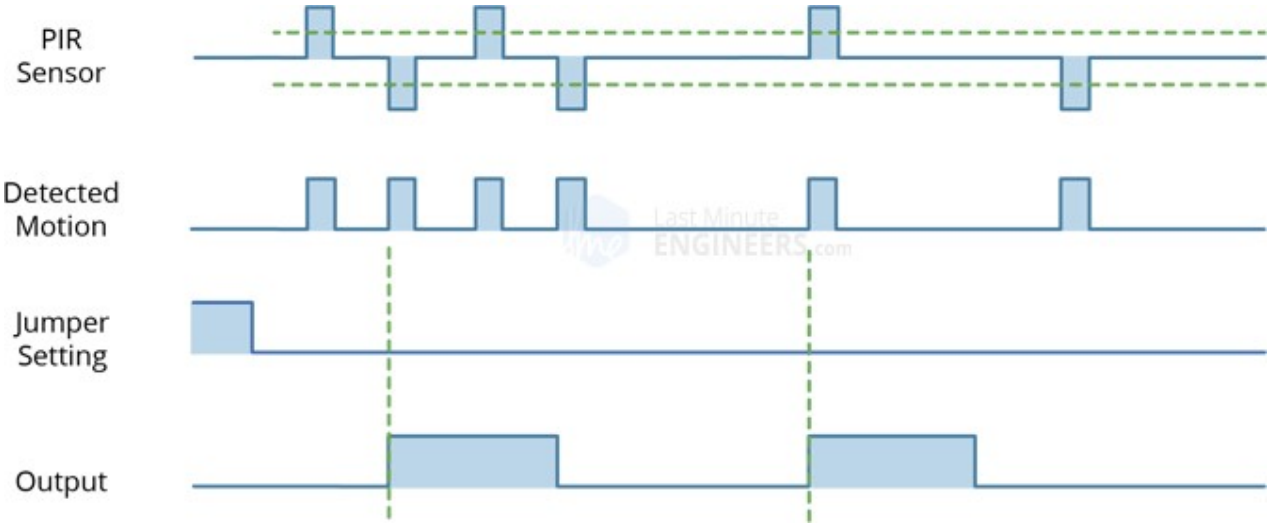
- Sensitivity– This sets the maximum distance that motion can be detected. It ranges from 3 meters to approximately 7 meters. The topology of your room can affect the actual range you achieve.
- Time– This sets how long the output will remain HIGH after detection. At minimum it is 3 seconds, at maximum it is 300 seconds or 5 minutes.

Finally the board has a jumper (on some models the jumper is not soldered in). It has two settings:

1. H– This is the Hold/Repeat/Retriggering In this position the HC-SR501 will continue to output a HIGH signal as long as it continues to detect movement.

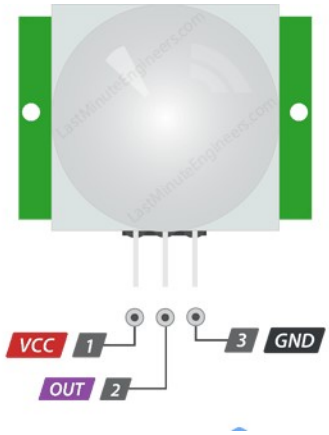


2. L– This is the Intermittent or No-Repeat/Non-Retriggering In this position the output will stay HIGH for the period set by the TIME potentiometer adjustment.



HC-SR501 PIR Sensor Pinout

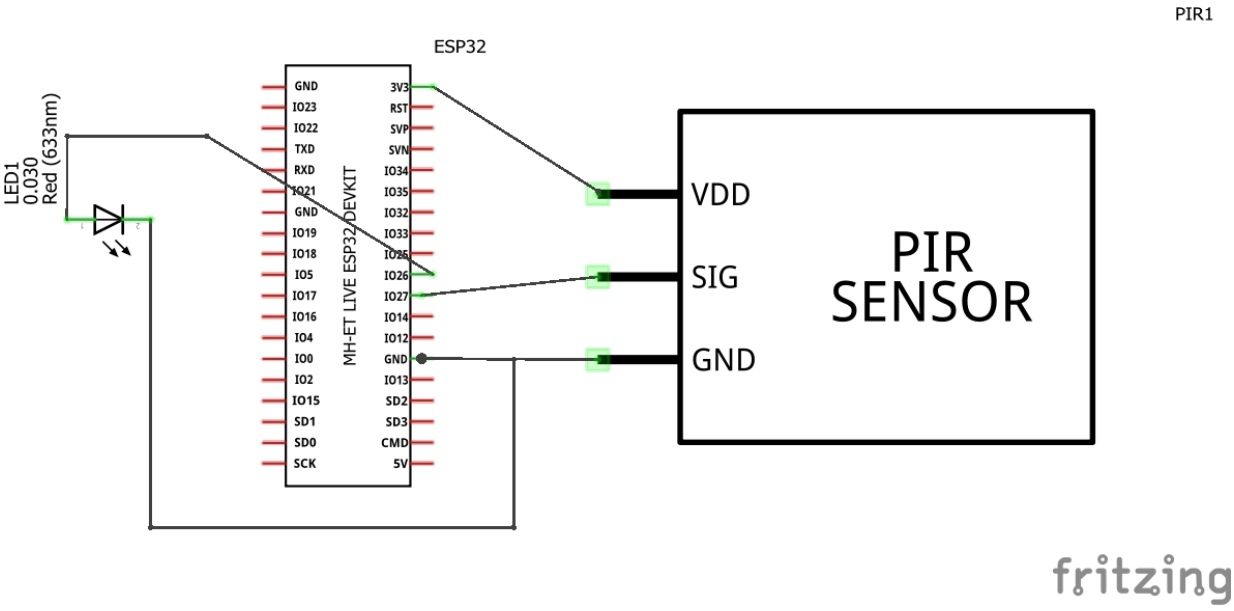
The HC-SR501 has a 3-pin connector that interfaces it to the outside world. The connections are as follows:

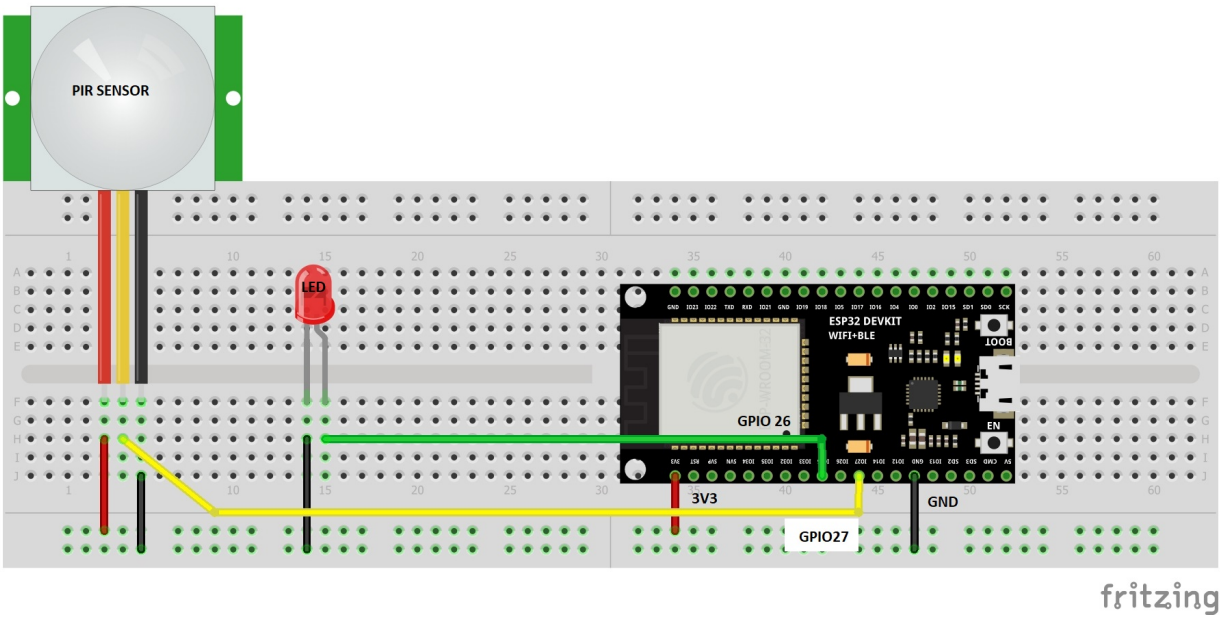


VCC is the power supply for HC-SR501 PIR sensor which we connect the 5V pin on the Arduino.
Output pin is a 3.3V TTL logic output. LOW indicates no motion is detected, HIGH means some motion has been detected.
GND should be connected to the ground of Arduino.

Schematic

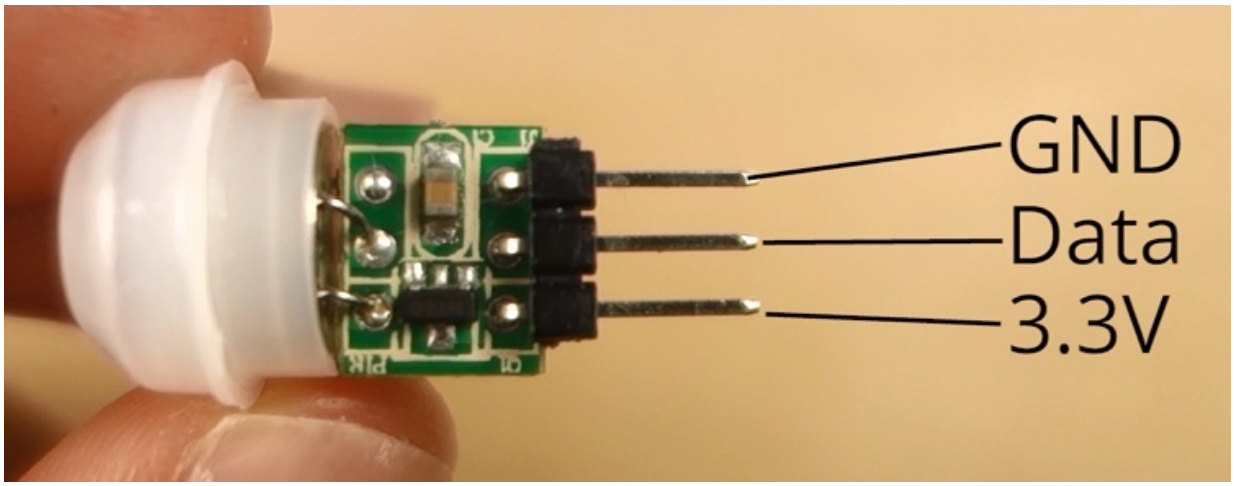
The circuit we'll build is easy to assemble, we'll be using an LED with a resistor. The LED is connected to GPIO 26. We'll be using the Mini AM312 PIR Motion Sensor that operates at 3.3V. It will be connected to GPIO 27. Simply follow the next schematic diagram.





Important:the Mini AM312 PIR Motion Sensor used in this project operates at 3.3V. However, if you're using another PIR motion sensor like the HC-SR501, it operates at 5V. You can either modify it to operate at 3.3V or simply power it using the Vin pin.

The following figure shows the AM312 PIR motion sensor pinout.



Algorithm:

1. Define GPIO26 as LED pin and GPIO27 as motionSensor pin.
2. In setup, define buzzer as OUTPUT pin and motionSensor pin as INPUT pin.
3. begin Serial communication with a certain baudrate.
4. In loop, If the sensor detects any movement then the buzzer will ring and prints motion detected.
5. if no motion is sensed from the sensor for specified interval then the buzzer goes off and prints no motion detected in the serial monitor.

Code:

```
int buzzer = 12;    // led is connected to pin 13

int pir = 3;  // output pin of the sensor is connected to digital pin 2

int motion = 0;     // variable to store the sensor status (value)

void setup()

{
```

```
   pinMode(buzzer, OUTPUT);   // initialise LED as an output

   pinMode(pir, INPUT); // initialise sensor as an input

   Serial.begin(9600);     // initialise serial

}

void loop()

{

   motion = digitalRead(pir);
   if (motion == HIGH)
   {

      digitalWrite(buzzer, HIGH); // turn LED ON
      Serial.println("Motion detected!");
   }

   else
   {

      digitalWrite(buzzer, LOW); // if val is low , led pin set to low
      Serial.println("Motion stopped!");

   }
   return;

}
```

Result:

Thus the interfacing with PIR sensor is done successfully.