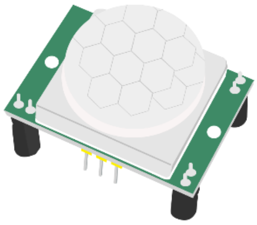
HC-SR501 PIR Sensor

DESCRIPTION:

The HC-SR501 PIR motion sensor is one of the easiest to connect to an Arduino -or any other microcontroller for that matter- and can also be used as a stand alone motion detector.

The HC-SR501 board has 2 variable resistors: looking at the back, with the connections facing upwards and the variable resistors at the bottom, the left resistor is for sensitivity and the right one is for output timing. for the sensitivity goes: Clockwise=>High sensitivity CCW=> low sensitivity (3-7 m). for the Output timing it is CW=>long, CCW=> short (3-300 sec) The right prong of the connector is for Vcc (+5-20V), the middle one is signal out and the left one is ground. The output is either high (3.3V) or low (0v)



There are two versions of the board. One with a 3 prong jumper and one with solder pads instead of a jumper. If the jumper is put in its bottom position (with the board still facing as described) there is no reset. If it is in its top position (H) it is in auto reset mode. If set to Auto-reset the sensor will stay high until the motion stops. After motion is no longer detected the output will go low. If set to No reset (L) the sensor will stop sensing once it has triggered, and stays high for the preset time period.

To choose one of these settings, simply create a solder-bridge between the labelled pad and the pad in the middle. the default as i understand is ‘L’ If you search internet for info about this module, you may come across contradicting info about which is ‘L’ and which is ‘H’ On my board it is as described: if the solder pads are in the lower-left corner, ‘L’ is the bottom one and ‘H’ the top one.

Specification：

● Voltage: 5V

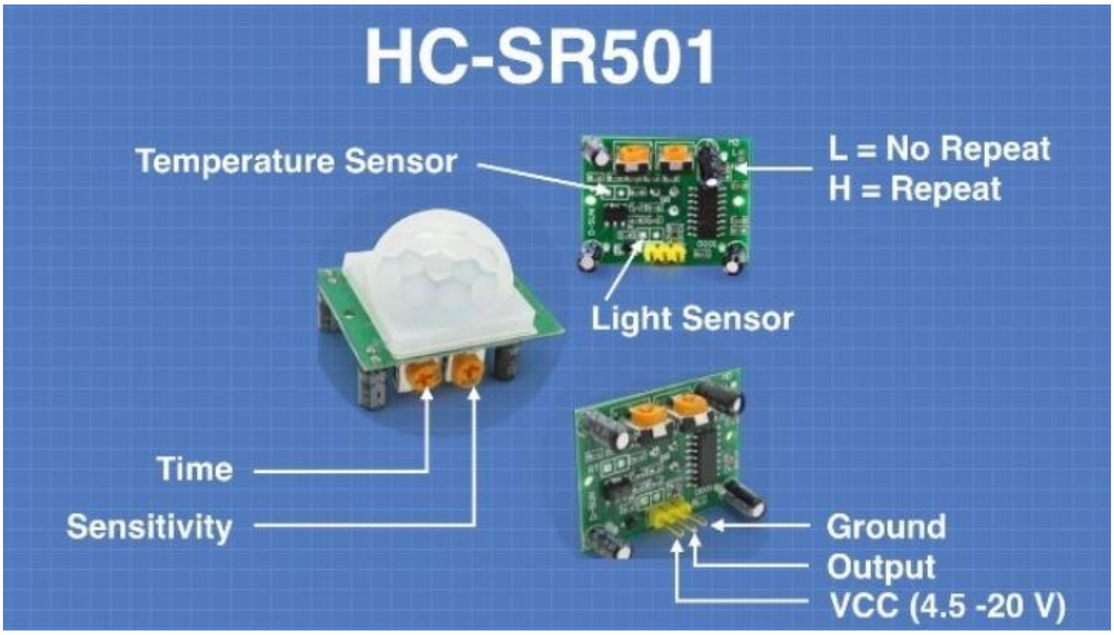
● Power Consumption: 65 mA

● TTL output: 3.3V, 0 V

● Trigger methods: L – disable repeat trigger, H enable repeat trigger

● Sensing range: less than 120 degree, within 7 meters

● Temperature: – 15 ~ +70



There are also 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 that 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:

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

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

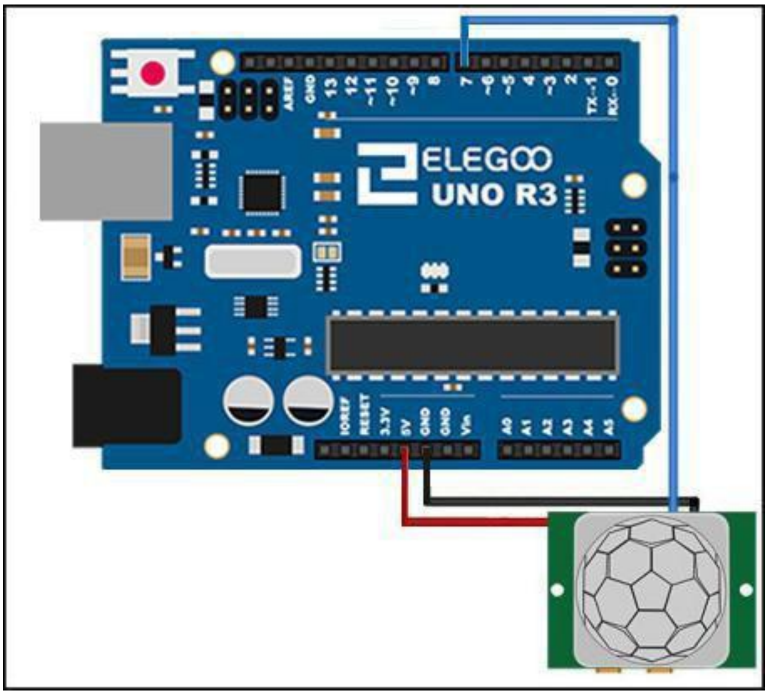
PIN CONFIGURATION:

●**VCC** – This is a positive DC voltage input from 4.5 to 20 VDC.

●**OUTPUT** – This is a 3.3 volt logic output. LOW indicates no detection,
HIGH means someone has been detected.

●**GND** – This is the Ground connection.

Example:



Code:

```
int ledPin = 13;                    // choose the pin for the LED
int inputPin = 7;                   // choose the input pin (for PIR sensor)
int pirState = LOW;                 // we start, assuming no motion detected
int val = 0;                        // variable for reading the pin status

void setup() {
  pinMode(ledPin, OUTPUT);           // declare LED as output
  pinMode(inputPin, INPUT);        // declare sensor as input

  Serial.begin(9600);
}

void loop(){
  val = digitalRead(inputPin);    // read input value
  if (val == HIGH) {                      // check if the input is HIGH
    digitalWrite(ledPin, HIGH);    // turn LED ON
    if (pirState == LOW) {
      // we have just turned on
      Serial.println("Motion detected!");
      // We only want to print on the output change, not state
      pirState = HIGH;
    }
  } else {
    digitalWrite(ledPin, LOW); // turn LED OFF
    if (pirState == HIGH){
      // we have just turned of
      Serial.println("Motion ended!");
      // We only want to print on the output change, not state
      pirState = LOW;
    }
  }
}
```