

PIR (Passive Infra-Red) Sensor

The PIR (Passive Infra-Red) Sensor is a pyroelectric device that detects motion by measuring changes in the infrared levels emitted by surrounding objects. This motion can be detected by checking for a high signal on a single I/O pin. PIR have low power and low cost, pretty rugged, have a wide lens range, and are easy to interface with. The product features include:

- Single bit output
- Small size makes it easy to conceal
- Compatible with all types of microcontrollers

The PIR sensor itself has two slots in it; each slot is made of a special material that is sensitive to IR. When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated. When a warm body like human or animal passing by, firstly it will 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. These change pulses are what is detected. For this reason the sensors are actually Fresnel lenses which focus the infrared signals onto the element.

Applications

- 1) Alarms system
- 2) Automatically-activated lighting systems
- 3) Automatic Door Opening System
- 4) Lift Lobby
- 5) For Basement or Covered Parking Area
- 6) Shopping Malls

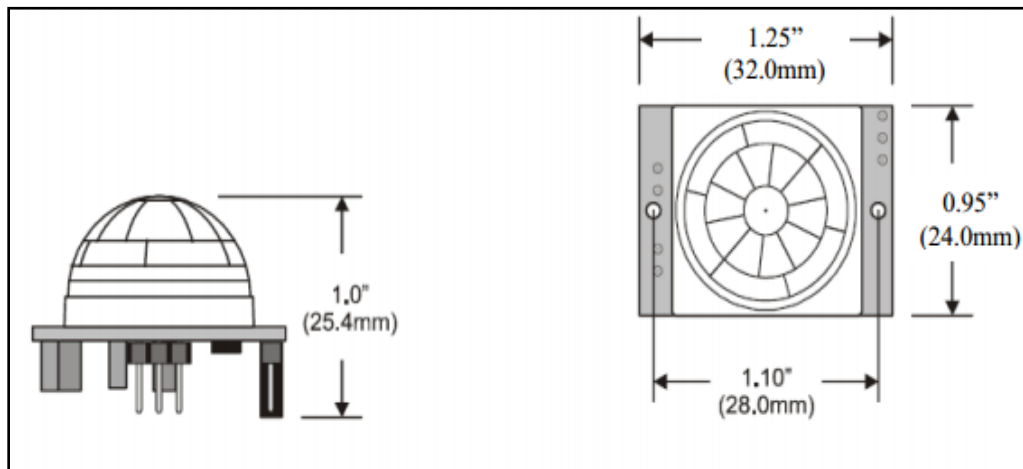
Technical Specifications

Operating voltage: 5–20 V with $< 100\mu\text{A}$

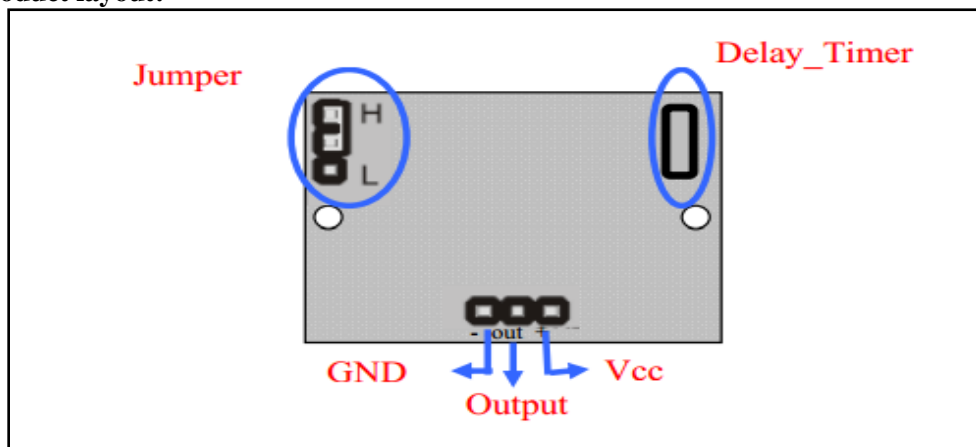
Operating temperature: -15°C to $+70^{\circ}\text{C}$

Dimensions, Layout and Pin-out

Product dimensions:



Product layout:



Pin definitions and ratings:

Pin	Name	Function
–	GND	Connects to Ground
OUT	Output	Connects to I/O pin set to INPUT mode
+	V _{CC}	Connects to V _{CC}

Jumper setting:

Position	Mode	Description
H	Retrigger	Output remains HIGH when sensor is triggered repeatedly. Output is LOW when idle (not triggered).
L	Normal	Output goes HIGH then LOW when triggered. Continuous motion results in repeated HIGH/LOW pulses. Output is LOW when idle.

Calibration

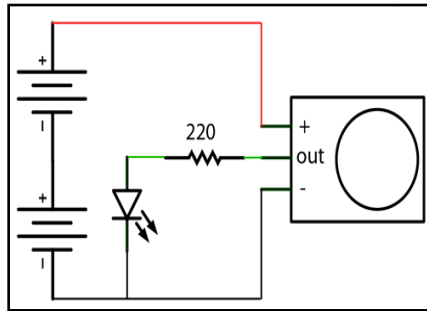
The PIR Sensor requires a ‘warm-up’ time in order to function properly. This is due to the settling time involved in ‘learning’ its environment. This could be anywhere from 10-60 seconds. During this time there should be as little motion as possible in the sensors field of view. There is a variable resistor (Delay Time) on the PIR sensor to control the ‘ON’ delay time for the sensor. Turning the variable resistor clockwise will give longer ‘ON’ delay time while turning anticlockwise will reduce the ‘ON’ delay time.

Sensitivity

The PIR Sensor has a range of approximately 5 meters. The PIR sensor can sense object up to 120° within 1 meter range. The sensitivity can vary with environmental conditions. The sensor is designed to adjust to slowly changing conditions that would happen normally as the day progresses and the environmental conditions change, but responds by making its output high when sudden changes occur, such as when there is motion.

Testing your PIR

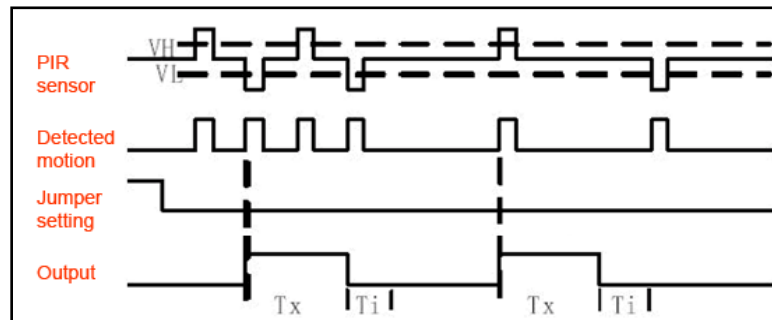
A simple test is done to verify that PIR Module works to its expectation and for range testing. Connect power supply between 3.5–6V to the + pin and connect Ground to the –pin on your PIR. Then connect a basic red LED and a 220 Ω resistor (value from 100 Ω to 1.0 k Ω) to the OUT pin as shown. The LED and resistor can swap locations as long as the LED is oriented connection and connects between OUT and Ground.



Now when the PIR detects motion, the Output pin will go "high" to and light up the LED. Once LED light up, wait for 10-60 seconds for the PIR to 'stabilize'. Wait until the LED is off and LED will light up when PIR detects motion in the infrared levels emitted by surrounding objects.

Retriggering

Make sure that the jumper is placed in the **L** position. The LED does not stay on when moving in front of it but actually turns on and off every second or so. That is called "non-retriggering".



Now change the jumper so that it is in the **H** position. The LED now does stay on the entire time that something is moving. That is called "retriggering".

