

Passive Infrared (PIR) Sensors

Overview

PIR or Passive Infrared Sensors are commonly used in projects where students want a device capable of motion detection. These sensors detect movement by measuring changes in infrared radiation emitted by warm objects, such as people or animals, within their field of view. These sensors are known as "passive" sensors, as they merely listen to changes in energy, and do not emit any signals themselves. This passive characteristic makes them ideal for low power applications.

Staff at the ITLL tested the detection range of these sensors along with doing some research and found it could sense motion up to approximately 23 feet or 7 meters (depending on model and conditions). You may have seen these sensors before as they are commonly used for automatic lights, interactive exhibits, or security alarms. At the ITLL, you can find them controlling the lights within our lab spaces to save energy.

Students are welcome to come by and check out sensors from the Electronics Center. These are the sensors you can check out from the ITLL: [link to inventory](#) . If you do not see a sensor listed feel free to come in and chat with an Engineer about your project.

What is a PIR Sensor?

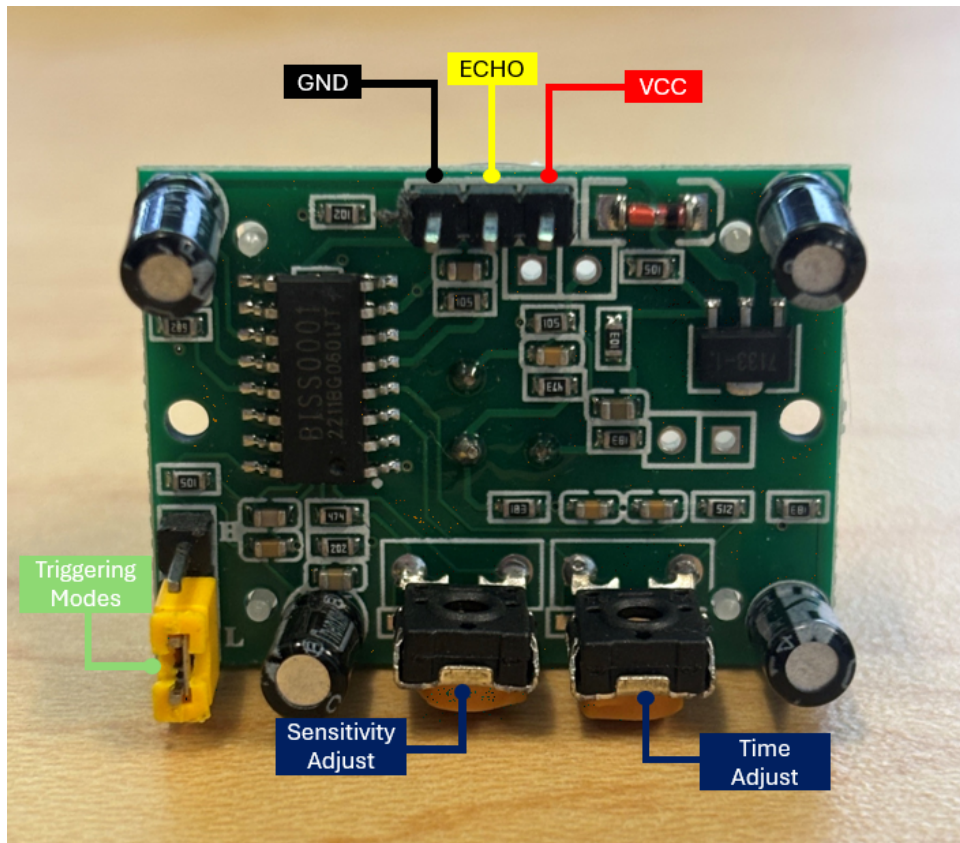


A PIR sensor comes with a small white dome shaped lens on top of the PCB. This lens, the Fresnel lens, helps focus light from any direction to the small pyroelectric sensor underneath. When motion occurs, half of this sensor detects a change in IR before the other, and the difference causes a

signal change (from LOW to HIGH). These sensors also include a sensitivity and time delay adjustment potentiometer. Which allow you to change the range or distance the sensor can detect, and how long the sensor remains active (before switching back from HIGH to LOW). Below you can find an image labeling the potentiometers.

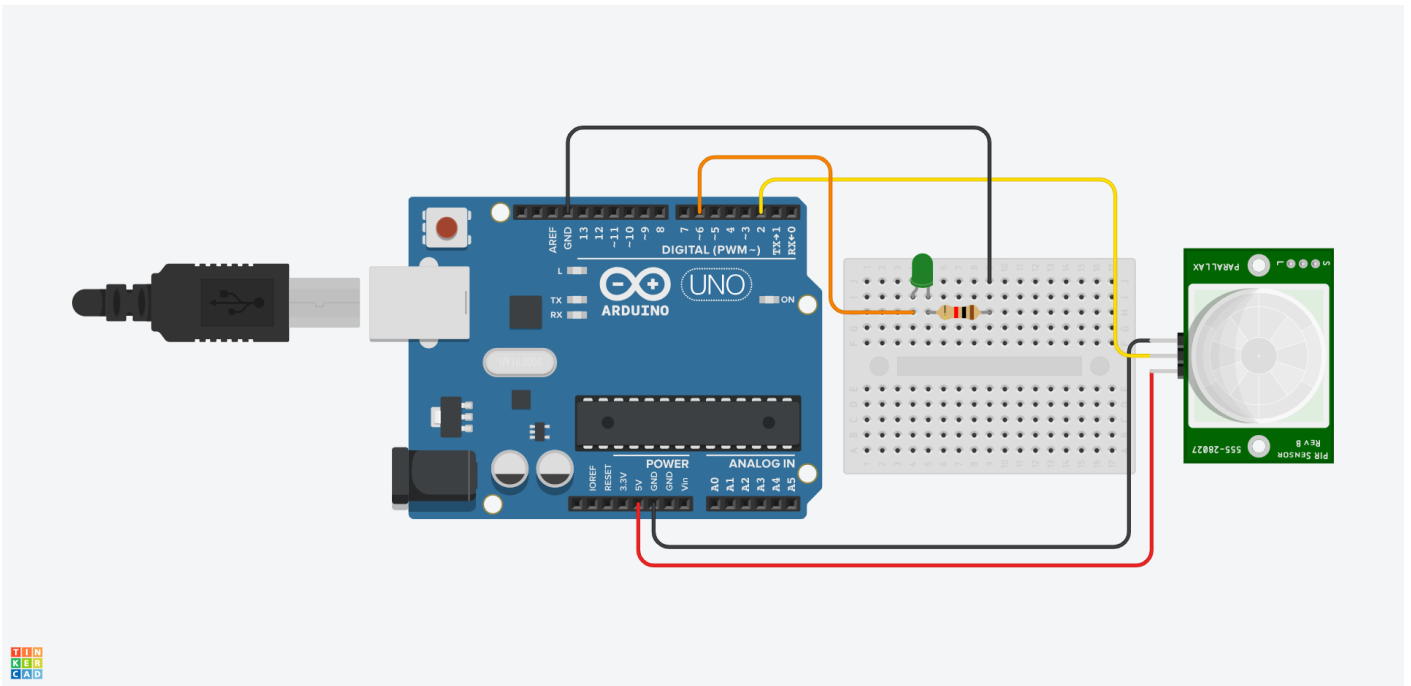
Wiring the Sensor

Important: Look up your specific sensors pinout and input voltage as they can differ. The pinout for the sensor we used in this demo was...



Example circuit setup:

- PIR sensor output connected to Arduino digital pin 2, VCC connected to 5V, and GND connected to GND.
- LED connected to Arduino digital pin 6 and 330 ohm resistor (resistor band color is different on picture).



Troubleshooting

1. If sensor reading always HIGH or ON
 - Sensor may be warming up (10-60 seconds after being turned on)
 - Reduce sensitivity with potentiometer
 - Ensure stable power supply
2. If no output changes
 - Verify correct wiring
 - Use Serial Monitor to debug readings
3. If false triggering
 - Avoid pointing sensor at heat source (like a vent)
 - Ensure no vibration of sensor mount

Simple Code



PIR_Sensor\$

```
int LED = 6;           // LED connected to digital pin 6
int PIR = 2;           // PIR motion sensor connected to digital pin 2

int state = LOW;       // Tracks previous motion state:
                        // LOW  = no motion previously detected
                        // HIGH = motion previously detected

int value = 0;         // Stores current reading from PIR sensor (HIGH or LOW)

void setup() {
  pinMode(LED, OUTPUT); // Set LED pin as output (Arduino controls LED)
  pinMode(PIR, INPUT);  // Set PIR pin as input (Arduino reads sensor signal)

  Serial.begin(9600);    // Start serial communication for debugging/output
}

void loop() {
  // Read the current value from the PIR sensor
  value = digitalRead(PIR); // Will be HIGH (motion) or LOW (no motion)

  // If motion is detected
  if (value == HIGH) {
    digitalWrite(LED, HIGH); // Turn LED ON
    delay(100);              // Keep LED on briefly (100 ms)

    // Only print when motion FIRST starts (prevents spamming Serial Monitor)
    if (state == LOW) {
      Serial.println("Motion Detected");
      state = HIGH;        // Update state to indicate motion is active
    }
  }
  else { // If NO motion is detected
    digitalWrite(LED, LOW); // Turn LED OFF
    delay(100);             // Small delay

    // Only print when motion JUST stopped
    if (state == HIGH) {
      Serial.println("Motion has stopped");
      state = LOW;          // Update state to indicate no motion
    }
  }
}
```

```

1. int LED = 6; // LED connected to digital pin 6
2. int PIR = 2; // PIR motion sensor connected to digital pin 2
3.
4. int state = LOW; // Tracks previous motion state:
5. // LOW = no motion previously detected
6. // HIGH = motion previously detected
7.
8. int value = 0; // Stores current reading from PIR sensor (HIGH or LOW)
9.
10. void setup() {
11.   pinMode(LED, OUTPUT); // Set LED pin as output (Arduino controls LED)
12.   pinMode(PIR, INPUT); // Set PIR pin as input (Arduino reads sensor signal)
13.
14.   Serial.begin(9600); // Start serial communication for debugging/output
15. }
16.
17. void loop() {
18.   // Read the current value from the PIR sensor
19.   value = digitalRead(PIR); // Will be HIGH (motion) or LOW (no motion)
20.
21.   // If motion is detected
22.   if (value == HIGH) {
23.     digitalWrite(LED, HIGH); // Turn LED ON
24.     delay(100); // Keep LED on briefly (100 ms)
25.
26.     // Only print when motion FIRST starts (prevents spamming Serial Monitor)
27.     if (state == LOW) {
28.       Serial.println("Motion Detected");
29.       state = HIGH; // Update state to indicate motion is active
30.     }
31.   }
32.   else { // If NO motion is detected
33.     digitalWrite(LED, LOW); // Turn LED OFF
34.     delay(100); // Small delay
35.
36.     // Only print when motion JUST stopped
37.     if (state == HIGH) {
38.       Serial.println("Motion has stopped");
39.       state = LOW; // Update state to indicate no motion
40.     }
41.   }
42. }

```

Revision #5

Created 2026-05-28 14:14:37 MDT by Tanner

Updated 2026-07-27 14:18:36 MDT by Tanner