

PIR Motion Sensor Module:DYP-ME003

From Wiki

Contents

- 1 Introduction
- 2 Feature
- 3 How does it work
 - 3.1 The PIR sensor itself
 - 3.2 Lenses
- 4 Schematic
- 5 Pin definition and Rating
- 6 Usage
 - 6.1 Testing your PIR
 - 6.2 Retriggering
 - 6.3 Changing pulse time and timeout length
 - 6.4 Programming
 - 6.5 Example
- 7 Version Tracker
- 8 Resources
- 9 How to buy
- 10 Licensing
- 11 External Links

Introduction

What is a PIR sensor?

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

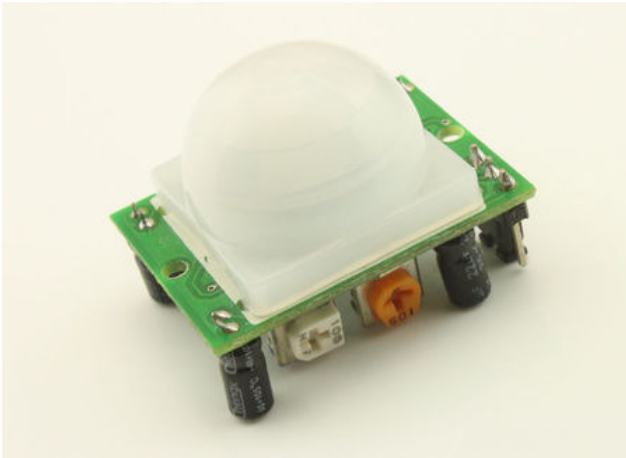
PIRs are basically made of a pyroelectric sensor (which you can see above as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves 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.

Along with the pyroelectric sensor (<http://en.wikipedia.org/wiki/Pyroelectric>) is a bunch of supporting circuitry, resistors and capacitors. It seems that most small hobbyist sensors use the BISS0001 ("Micro Power PIR Motion Detector IC"), undoubtedly a very inexpensive chip. This chip takes the output of the sensor and does some minor processing on it to emit a digital output pulse from the analog sensor.

For many basic projects or products that need to detect when a person has left or entered the area, or has approached, PIR sensors are great. They are low power and low cost, pretty rugged, have a wide lens range, and are easy to interface with. Note that PIRs won't tell you how many people are around or how close they are to the sensor, the lens is often fixed to a certain sweep and distance (although it can be hacked somewhere) and they are also sometimes set off by housepets. Experimentation is key!

DYP-ME003 PIR Motion Sensor Module is base on BISS0001(datasheet),it can automatically and quickly open various types of incandescent, fluorescent lamps, buzzer, automatic doors, electric fans, automatic washing machine and dryer Machines and other devices, is a high-tech products. Especially suitable for enterprises, hotels, shopping malls, warehouses and family aisles, corridors and other sensitive.

Model: SEN005 (<http://www.elec Freaks.com/store/pir-motion-sensor-moduledypme003-p-156.html>)



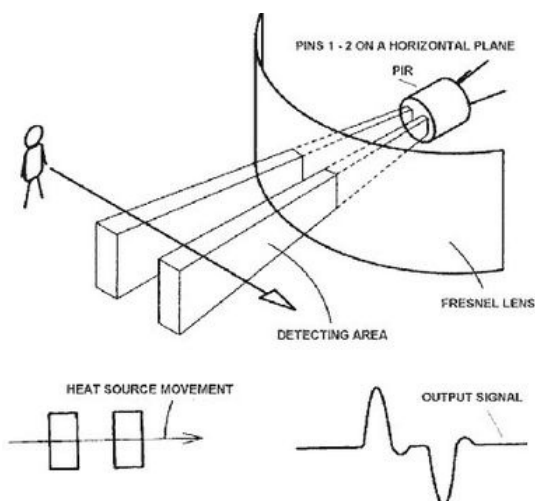
Feature

- Input Voltage: DC 4.5-20V
- Static current: 50uA
- Trigger: H-Yes, L-No
- Block time: 2.5 S(default)
- Delay time: 5 S(default)
- Sentry Angle: < 110 degree
- Sentry Distance: 3 m(default) - max 7 m
- Lens Size: Diameter: 23mm(Default)
- Dimensions: 32mm * 24mm

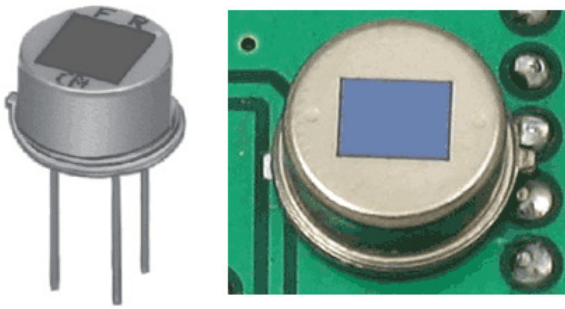
How does it work

PIR sensors are more complicated than many of the other sensors explained in these tutorials (like photocells, FSRs and tilt switches) because there are multiple variables that affect the sensors input and output. To begin explaining how a basic sensor works, we'll use this rather nice diagram (if anyone knows where it originates plz let me know).

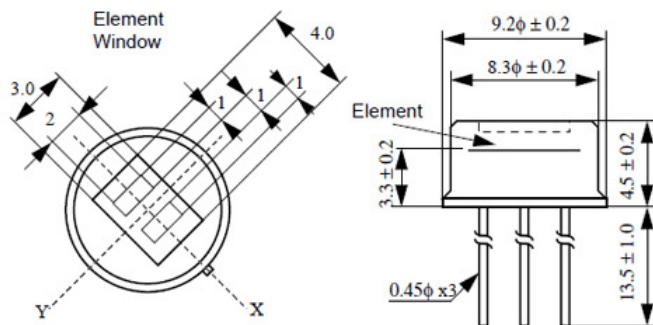
The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. The lens used here is not really doing much and so we see that the two slots can 'see' out past some distance (basically the sensitivity of the sensor). When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. 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. These change pulses are what is detected



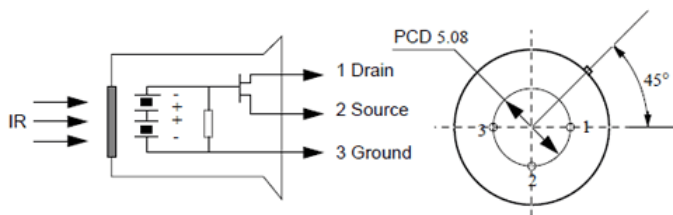
The PIR sensor itself



The IR sensor itself is housed in a hermetically sealed metal can to improve noise/temperature/humidity immunity. There is a window made of IR-transmissive material (typically coated silicon since that is very easy to come by) that protects the sensing element. Behind the window are the two balanced sensors.



You can see above the diagram showing the element window, the two pieces of sensing material

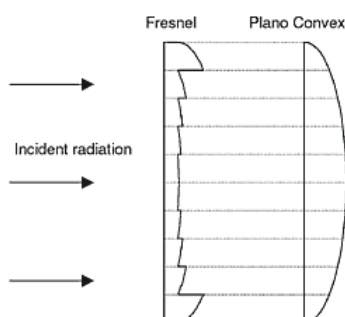


This image shows the internal schematic. There is actually a JFET inside (a type of transistor) which is very low-noise and buffers the extremely high impedance of the sensors into something a low-cost chip (like the BIS0001) can sense.

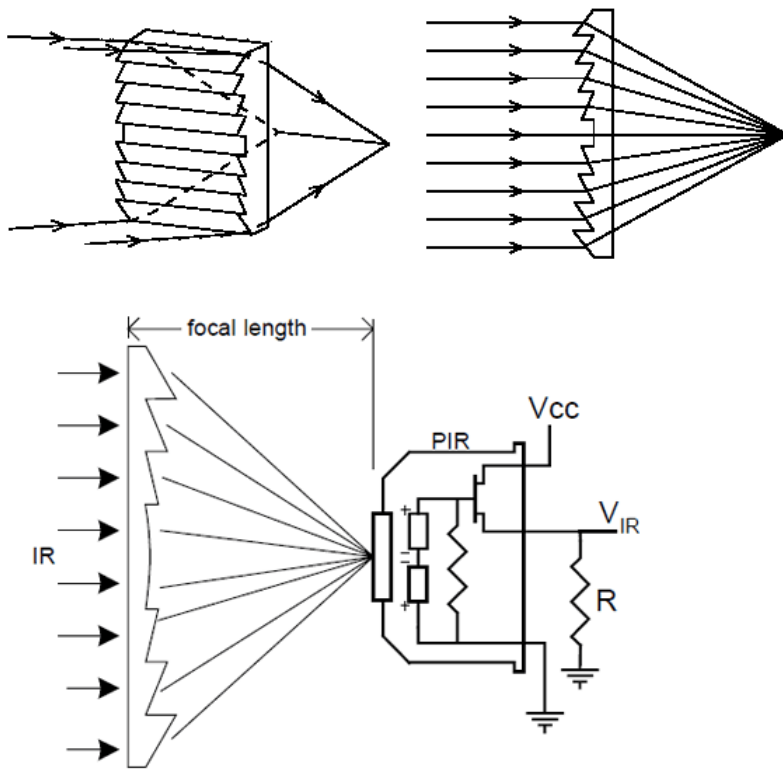
Lenses

PIR sensors are rather generic and for the most part vary only in price and sensitivity. Most of the real magic happens with the optics. This is a pretty good idea for manufacturing: the PIR sensor and circuitry is fixed and costs a few dollars. The lens costs only a few cents and can change the breadth, range, sensing pattern, very easily.

In the diagram up top, the lens is just a piece of plastic, but that means that the detection area is just two rectangles. Usually we'd like to have a detection area that is much larger. To do that, we use a simple lens ([http://en.wikipedia.org/wiki/Lens_\(optics\)](http://en.wikipedia.org/wiki/Lens_(optics))) such as those found in a camera: they condense a large area (such as a landscape) into a small one (on film or a CCD sensor). For reasons that will be apparent soon, we would like to make the PIR lenses small and thin and moldable from cheap plastic, even though it may add distortion. For this reason the sensors are actually Fresnel lenses (http://en.wikipedia.org/wiki/Fresnel_lens) :



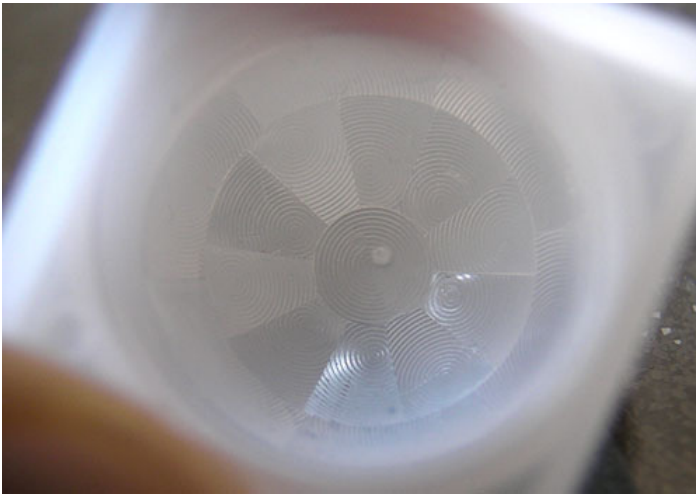
The Fresnel lens condenses light, providing a larger range of IR to the sensor.



OK, so now we have a much larger range. However, remember that we actually have two sensors, and more importantly we don't want two really big sensing-area rectangles, but rather a scattering of multiple small areas. So what we do is split up the lens into multiple sections, each section of which is a Fresnel lens

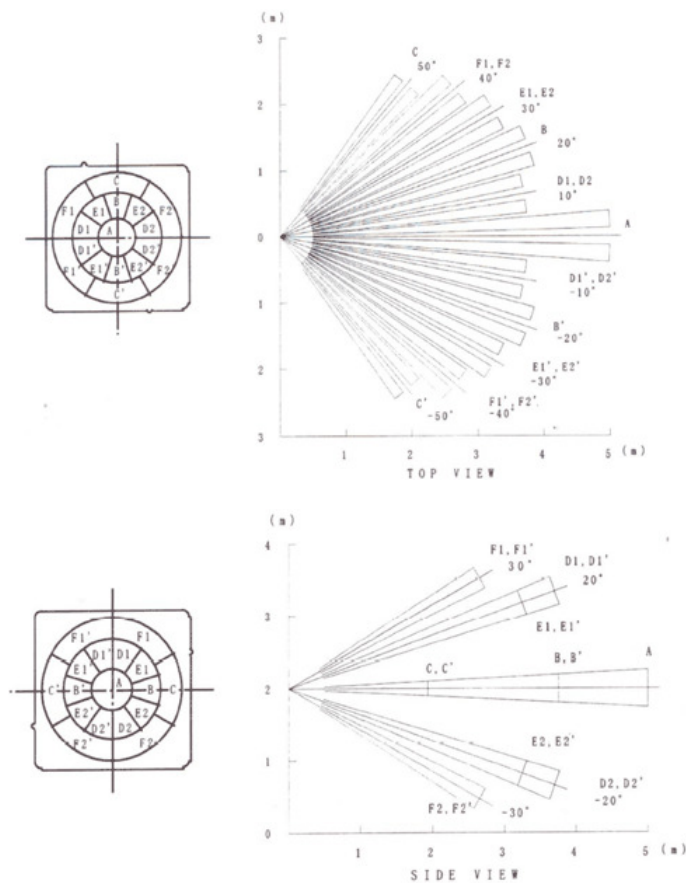


Here you can see the multiple facet-sections

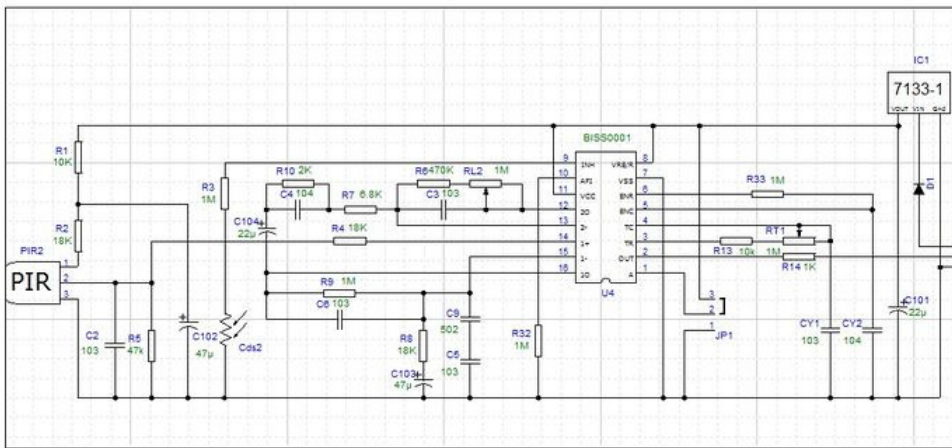


This macro shot shows the different Fresnel lenses in each facet!

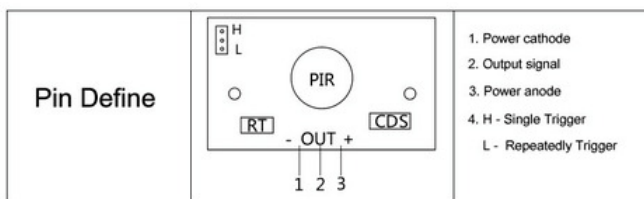
The different faceting and sub-lenses create a range of detection areas, interleaved with each other. That's why the lens centers in the facets above are 'inconsistent' - every other one points to a different half of the PIR sensing element



Schematic



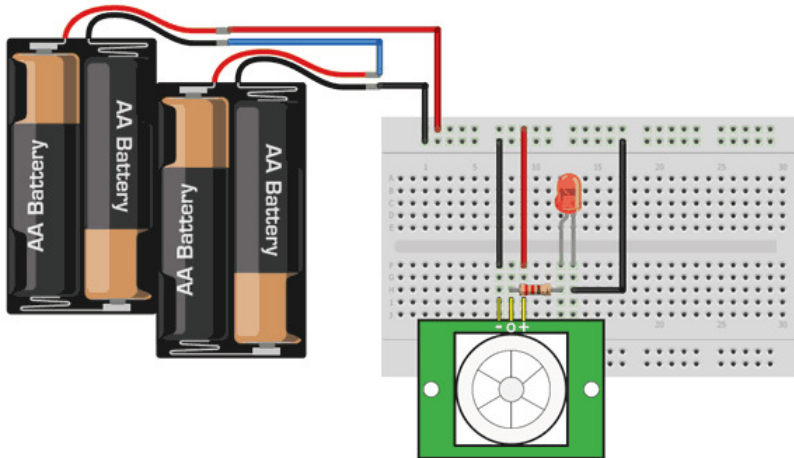
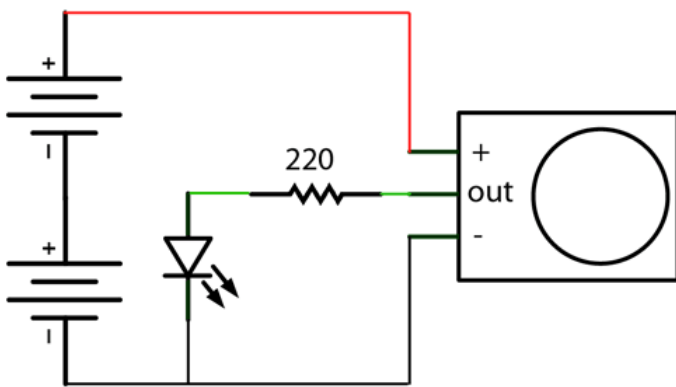
Pin definition and Rating



Usage

Testing your PIR

Once you have your PIR wired up its a good idea to do a simple test to verify that it works the way you expect. This test is also good for range testing. Simply connect 3-4 alkaline batteries (make sure you have more than 3.5VDC out but less than 6V by checking with your multimeter!) and connect ground to the - pin on your PIR. Power goes to the + pin. Then connect a basic red LED (red LEDs have lower forward voltages than green or blue so they work better with only the 3.3v output) and a 220Ω resistor (any value from 100Ω to 1.0KΩ will do fine) to the out pin as shown. Of course, the LED and resistor can swap locations as long as the LED is oriented connection and connects between out and ground



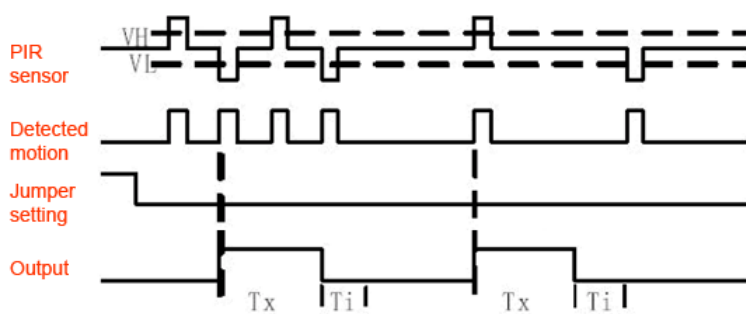
Now when the PIR detects motion, the output pin will go "high" to 3.3V and light up the LED!

Once you have the breadboard wired up, insert batteries and wait 30-60 seconds for the PIR to 'stabilize'. During that time the LED may blink a little. Wait until the LED is off and then move around in front of it, waving a hand, etc, to see the LED light up!

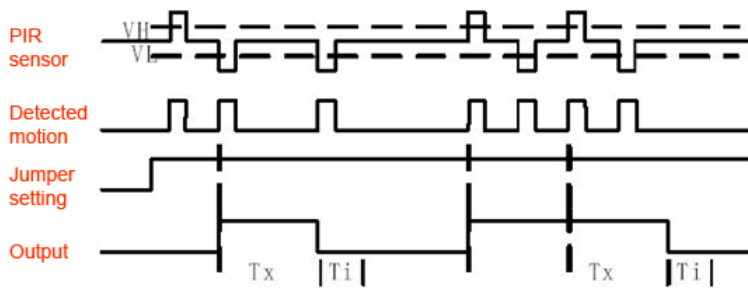
Retriggering

Once you have the LED blinking, look on the back of the PIR sensor and make sure that the jumper is placed in the L position as shown above.

Now set up the testing board again. You may notice that when connecting up the PIR sensor as above, 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. If you set up the test, you will notice that now the LED does stay on the entire time that something is moving. That is called "retriggering".



(The graphs above are from the BISS0001 datasheet, they kinda suck)

For most applications, "retriggering" (jumper in H position) mode is a little nicer. If you need to connect the sensor to something edge-triggered, you'll want to set it to "non-retriggering" (jumper in L position).

Changing pulse time and timeout length

There are two 'timeouts' associated with the PIR sensor. One is the "Tx" timeout: how long the LED is lit after it detects movement. The second is the "Ti" timeout which is how long the LED is guaranteed to be off when there is no movement. These are not easily changed but if you're handy with a soldering iron it is within reason.

First, lets take a look at the BISS datasheet again

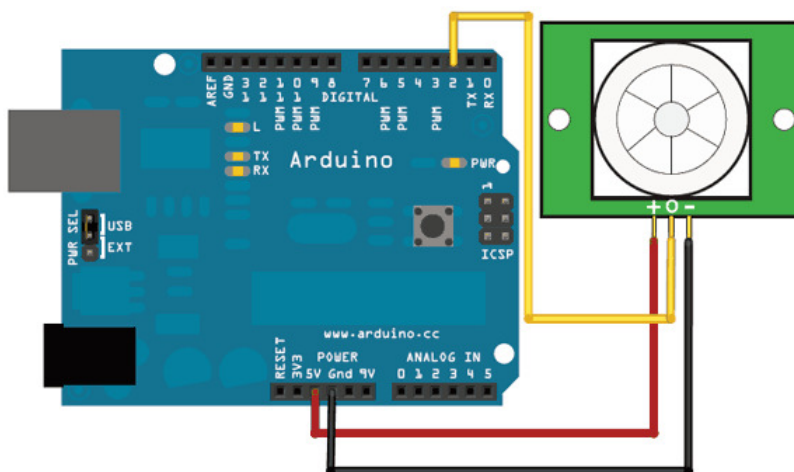
Tx = The time duration during which the output pin (Vo) remains high after triggering.
Ti = During this time period, triggering is inhibited. See timing charts for details.

$Tx \approx 24576 \times R10 \times C6$; $Ti \approx 24 \times R9 \times C7$. (ref to schematic)

Determining R10 and R9 isnt too tough. Unfortunately this PIR sensor is mislabeled (it looks like they swapped R9 R17). You can trace the pins by looking at the BISS001 datasheet and figuring out what pins they are - R10 connects to pin 3 and R9 connects to pin 7. the capacitors are a little tougher to determine, but you can 'reverse engineer' them from timing the sensor and solving!

You can change the timing by swapping different resistors or capacitors. For a nice tutorial on this, see Keith's PIR hacking page (<http://www.neufeld.newton.ks.us/electronics/?p=208>)

Programming



Includes important code snippet. Demo code like :

```

/*
 * PIR sensor tester
 */

int ledPin = 13;           // choose the pin for the LED
int inputPin = 2;          // 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

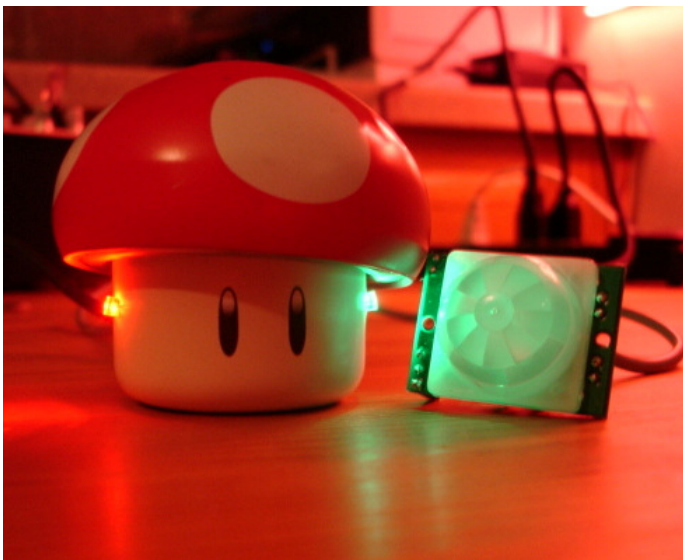
  Serial.begin(9600);
}

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;
    }
  }
}
```

Don't forget that there are some times when you don't need a microcontroller. A PIR sensor can be connected to a relay (perhaps with a transistor buffer) without a micro!

Example

The projects and application examples.

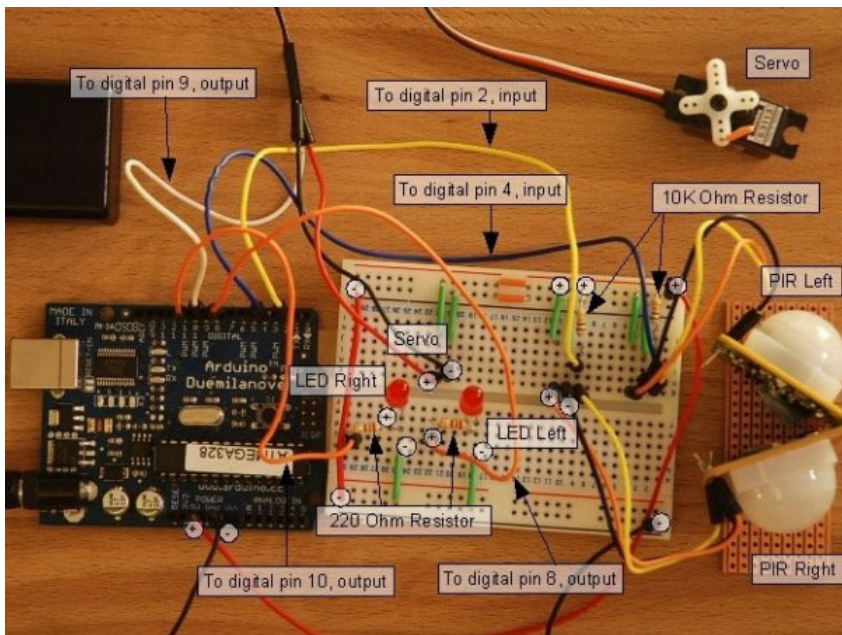


A USB-powered singing and blinking Mario mushroom (there's a video on the site!) (<http://jarv.org/mario.shtml>)



Rain Umbrellas (<http://coopy.sproutlab.com/projects/rain-umbrellas/>) Testing a PIR sensor for interfacing to Max/MSP for an interactive garden

A home-made security system using PIR sensors (which is built into a Start Trek panel!!) PIR sensor + Arduino + Servo = automatic cat door!



A 2-PIR motion tracker (<http://luckylarry.co.uk/2009/07/arduino-very-basic-motion-tracking-with-2-pir-sensors/>) by Lucky Larry



A PIR-based remote camera trigger (also by Lucky Larry!) (<http://luckylarry.co.uk/2009/07/arduino-motion-triggered-camera/>)

An interesting hack whereby the PIR sensor is used 'raw' to track movement

Version Tracker

	Revision	Descriptions	Release
v0.9b		Initial public release	date

Resources

- BISSOOO1 Datasheet. (<http://elecfreaks.com/store/download/datasheet/sensor/DYP-ME003/BISS0001.pdf>)
- Schematic(jpg). (<http://elecfreaks.com/store/download/datasheet/sensor/DYP-ME003/DYP-ME003.zip>)
- Specification. (<http://elecfreaks.com/store/download/datasheet/sensor/DYP-ME003/Specification.pdf>)
- Related Articles (<http://www.elecfreaks.com/398.html>)

How to buy

Click here to buy: <http://www.elecfreaks.com/store/pir-motion-sensor-moduledypme003-p-156.html>

Licensing

This documentation is licensed under the Creative Commons Attribution-ShareAlike License 3.0 (<http://creativecommons.org/licenses/by-sa/3.0/>) Source code and libraries are

licensed under GPL/LGPL (<http://www.gnu.org/licenses/gpl.html>) , see source code files for details.

External Links

Links to external webpages which provide more application ideas, documents/datasheet or software libraries[[Category:senso

Retrieved from "http://www.elecfreaks.com/wiki/index.php?title=PIR_Motion_Sensor_Module:DYP-ME003"

Categories: PIR | All Products

- This page was last modified on 13 September 2011, at 03:46.
- This page has been accessed 53,109 times.

- About Wiki
- Disclaimers