

Project 2: Sensor Light

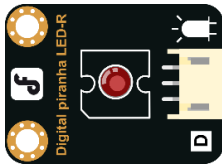
In this session, we will create a sensor light. The behavior we want to implement is as follows:

- When the sensor detects human motion, the LED light will be on
- When no human motion is detected, the LED light will be off.

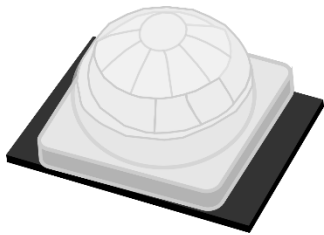
To achieve this, we will need to use an IR sensor that detects infra-red light.

COMPONENT LIST:

1× Digital Piranha LED-R



1× PIR Sensor



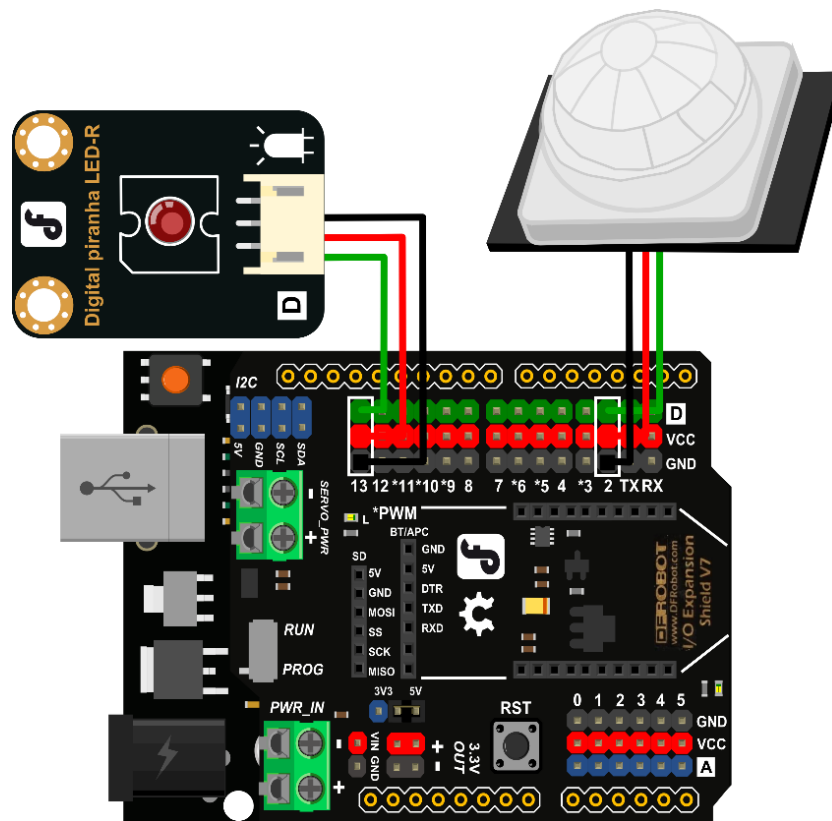
HARDWARE CONNECTION

We need to make the following connections:

- Connect the IR sensor to digital pin 2
- Connect the Digital Piranha LED-R to digital pin 13

Refer to the diagram below for further details. Be sure your power, ground and signal connections are correct or you risk damaging your components.

When your connections are complete, you can move to the Arduino IDE to upload code to the Arduino.



CODE INPUT

Sample Code 2-1:

```
// Item Two-Sensor Light

int sensorPin = 2; // IR sensor to be attached to digital pin NO.2

int ledPin = 13; // Digital Piranha LED-R to be attached to digital pin NO.13

// variable sensorState for storing statistics about the status of the sensor

int sensorState = 0;

void setup() {
    pinMode(ledPin, OUTPUT); // LED is the output unit
    pinMode(sensorPin, INPUT); // Sensor is the input unit
}

void loop(){
    // Read statistics collected from the sensor
    sensorState = digitalRead(sensorPin);
```

```

// If it is HIGH, LED light will be turned on
if (sensorState == HIGH) {
  digitalWrite(ledPin, HIGH);
}

else {          // If it is LOW, LED light will be turned off
  digitalWrite(ledPin, LOW);
}
}

```

Use this sample code to implement the behavior we want.

You can copy and paste it in to the Arduino IDE, but if you want to develop your skills we recommend typing it out.

When you have finished, click “Verify” to check the code for syntax errors. If the code verifies successfully, you can upload it to your Arduino.

When the sensor detects motion, the LED should turn on.

When no motion is detected, the LED should turn off.

HARDWARE ANALYSIS

The whole device is composed of three parts: an input unit, a control unit and an output unit. The IR sensor is the input unit; the Arduino is the control unit and the Digital Piranha LED-R is the output unit.

The IR sensor and LED use digital values, so both should be attached to a digital pin.



CODE ANALYSIS

The IR sensor is the input unit; the LED is the output unit. Initialize them in the setup.

```
pinMode(ledPin, OUTPUT);          // Define LED as the output unit

pinMode(sensorPin, INPUT);        // Define sensor as the input unit
```

We can read values from digital pins using the function "digitalRead()" . We write it in the loop() function.

```
sensorState = digitalRead(sensorPin);
```

The function is as follows:

```
digitalRead(pin)
```

This function is used to read the states of digital pins, either HIGH or LOW. When the IR sensor detects motion, the state is HIGH.

The next part of the code will determine actions based on the state of the sensor. (HIGH stands for 1; LOW represents 0).

The digital sensor can only read two values (HIGH or LOW).

We' ll use the "if" statement here.

The "if" statement is usually used with a comparison operator, such as "==" (equal), "!=" (not equal), "<" (less than) or ">" (greater than). It allows the program to evaluate an expression, and then make a logical decision. Usually we use it to test if a particular condition has been reached.

Let' s examine the application of the "if" statement in our code:

```
if (sensorState == HIGH) {

    digitalWrite(ledPin, HIGH);

}

else{

    digitalWrite(ledPin, LOW);

}

}
```

In this case, the if statement evaluates whether the sensor state is HIGH or LOW. If it is HIGH, it will turn the LED HIGH (on) as well.

“else” is used to give an alternative output if the output condition in the statement is not met.

If the sensor is not HIGH, the LED is LOW (off).

We briefly touched upon comparison operators above. Let’s examine them in more detail:

$x == y$ (x is equal to y)

$x != y$ (x is not equal to y)

$x < y$ (x is less than y)

$x > y$ (x is greater than y)

$x <= y$ (x is less than or equal to y)

$x >= y$ (x is greater than or equal to y)

When we use these in code, the Arduino will make an evaluation between values and then make a decision.

TRY IT YOURSELF

(1) You can make a model of a ghostly face with flashing LEDs inside it. You can change the interval of the LED flashing to make them quicker or slower for different effects. It would be great for a haunted house!

(2) You can make a decorating light with the IR sensor. Checkout the tutorial:

<http://www.dfrobot.com.cn/community/forum.php?mod=viewthread&tid=1983&highlight=IQ%E7%81%AF>