**Experiment No. 1a**

**Aim:** Write a programm to test active sensors using smart boards.

Apparatus: PIR Sensor, Arduino Uno .Connecting Wire, LED.

**Theory**:Connecting PIR sensors to a microcontroller . The PIR acts as a digital output, it can be high voltage or low voltage, so all you need to do is listen for the pin to flip high (detected) or low (not detected) by listening on a digital input on your Arduino.Its likely that you'll want reriggering, so be sure to put the jumper in the **H** position!

Power the PIR with 5V and connect ground to ground. Then connect the output to a digital pin. In this example we'll use pin 2.The code is very simple, and is basically just keeps track of whether the input to pin 2 is high or low. It also tracks the state of the pin, so that it prints out a message when motion has started and stopped.

PIR: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.

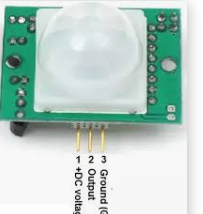


Fig1: PIR Sensor

LED-A **light-emitting diode** (**LED**) is a [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) [light source](https://en.wikipedia.org/wiki/Light_source) that emits light when [current](https://en.wikipedia.org/wiki/Electric_current) flows through it. [Electrons](https://en.wikipedia.org/wiki/Electron) in the semiconductor recombine with [electron holes](https://en.wikipedia.org/wiki/Electron_hole), releasing energy in the form of [photons](https://en.wikipedia.org/wiki/Photon). The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the [band gap](https://en.wikipedia.org/wiki/Band_gap) of the semiconductor.[[5]](https://en.wikipedia.org/wiki/Light-emitting_diode#cite_note-5) White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.[[6]](https://en.wikipedia.org/wiki/Light-emitting_diode#cite_note-6)



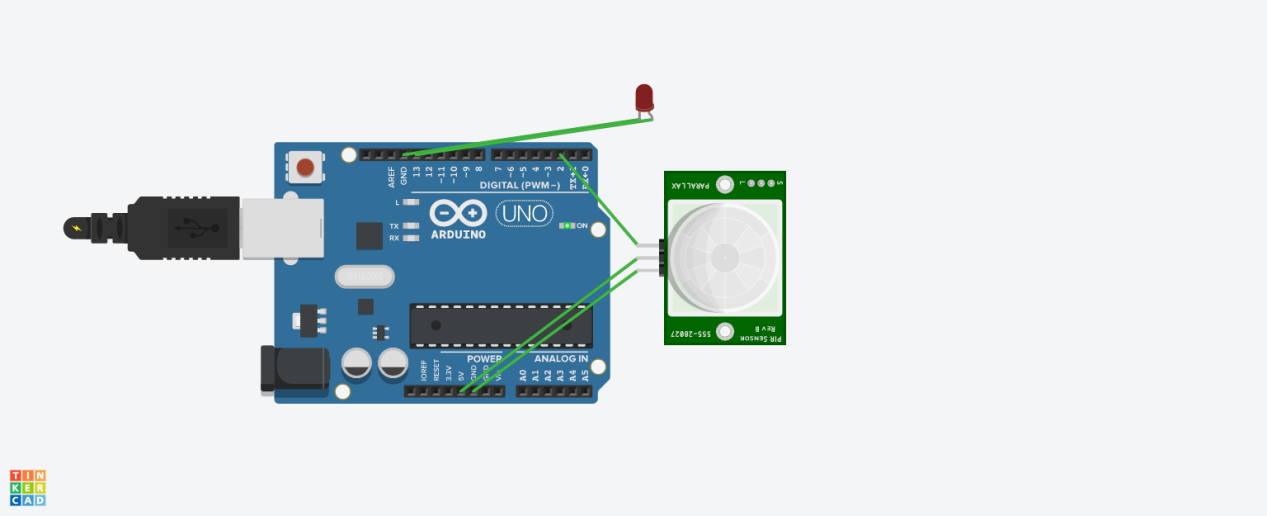
Fig 2: LED

Arduino UNO-The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which six can be used as PWM outputs), six analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Arduino Uno differs from all preceding boards because it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter. Revision 2 of the Arduino Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

|  |  |
| --- | --- |
| **Features** |  |
| * 14 digital I/O pins (six of which provide PWM output) * 3.3 V supply generated by an on-board regulator * Six analog input pins * 32 KB of flash memory | * Can supply 40 mA of DC current per pin * 16 MHz clock speed * Code example from Arduino website to help get started |

**Circuit Diagram:**



**Code**:

int ledPin = 13;

int inputPin = 2;

int pirState = LOW;

int val = 0;

void setup() {

pinMode(ledPin, OUTPUT);

pinMode(inputPin, INPUT);

Serial.begin(9600);

}

void loop(){

val = digitalRead(inputPin);

if (val == HIGH) {

digitalWrite(ledPin, HIGH);

if (pirState == LOW) {

Serial.println("Motion detected!");

pirState = HIGH;

}

} else {

digitalWrite(ledPin, LOW);

if (pirState == HIGH){

Serial.println("Motion ended!");

pirState = LOW;

}

}

}

**Result:**

