IR (Infrared) sensor with an Arduino Uno

Aim:

an IR (Infrared) sensor with an Arduino Uno is a popular project that can be implemented for various applications like object detection, remote control, or even basic robotics. Below is a guide on how to set up an IR sensor with an Arduino Uno.

**Components Needed**

1. **Arduino Uno** - The main microcontroller board.
2. **IR Sensor Module** - This could be a simple IR receiver (like TSOP1738) or an IR proximity sensor (like HC-SR501).
3. **Breadboard and Jumper Wires** - For making connections.
4. **LED** (optional) - To visualize the sensor output.
5. **220-ohm Resistor** (if using an LED).

**Wiring Diagram**

Here's a simple wiring setup for an IR receiver module like the TSOP1738:

* **IR Receiver (TSOP1738):**
  + **GND** to Arduino GND
  + **VCC** to Arduino 5V
  + **OUT** to Arduino Digital Pin (e.g., Pin 2)

**Code:**

// Define pin numbers

const int ledPin = 3;        // LED connected to digital pin 5

const int irSensorPin = 2;   // IR Sensor connected to digital pin 2

void setup() {

  pinMode(ledPin, OUTPUT);      // Set the LED pin as output

  pinMode(irSensorPin, INPUT);  // Set the IR sensor pin as input

}

void loop() {

  int sensorState = digitalRead(irSensorPin); // Read the state from the IR sensor

  // The sensor outputs LOW when it detects an obstacle

  if (sensorState == LOW) {

    digitalWrite(ledPin, HIGH);  // Turn the LED on

  } else {

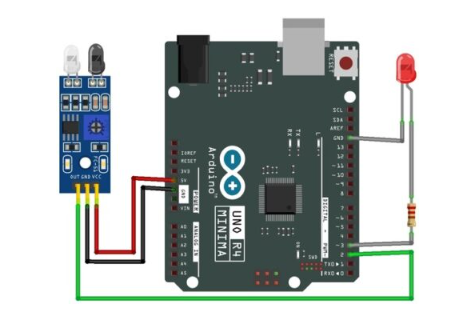
    digitalWrite(ledPin, LOW);   // Turn the LED off

  }

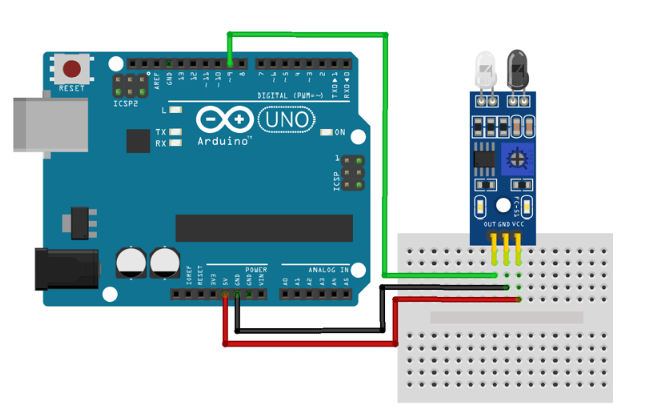
  delay(100);  // Wait for 100 milliseconds

}

Circuit Diagram:(With led)



Circuit Diagram:(Serial monitor)



Code for serial monitor:

int IRSensor = 9; // connect IR sensor module to Arduino pin D9

int LED = 13; // connect LED to Arduino pin 13

Next, we have our setup function. In the setup function, we initialize the serial with 115200 baud. Next, we print a statement to check if the serial monitor window is properly working or not, and then we initialize the IRSensor pin as input and the LED pin as output.

void setup(){

Serial.begin(115200); // Init Serial at 115200 Baud Rate.

Serial.println("Serial Working"); // Test to check if serial is working or not

pinMode(IRSensor, INPUT); // IR Sensor pin INPUT

pinMode(LED, OUTPUT); // LED Pin Output

}

Next, we have our infinite loop. In the infinite loop, we first read the sensor pin with the **digitalRead()** function and store the value to **sensorStatus**variable. Then we check to see if the output of the sensor is high or low, if the output of the sensor is high that means no motion is detected, else motion is detected, we also print this status in the serial monitor window.

void loop(){

int sensorStatus = digitalRead(IRSensor); // Set the GPIO as Input

if (sensorStatus == 1) // Check if the pin high or not

{

// if the pin is high turn off the onboard Led

digitalWrite(LED, LOW); // LED LOW

Serial.println("Motion Detected!"); // print Motion Detected! on the serial monitor window

}

else {

//else turn on the onboard LED

digitalWrite(LED, HIGH); // LED High

Serial.println("Motion Ended!"); // print Motion Ended! on the serial monitor window

}

}