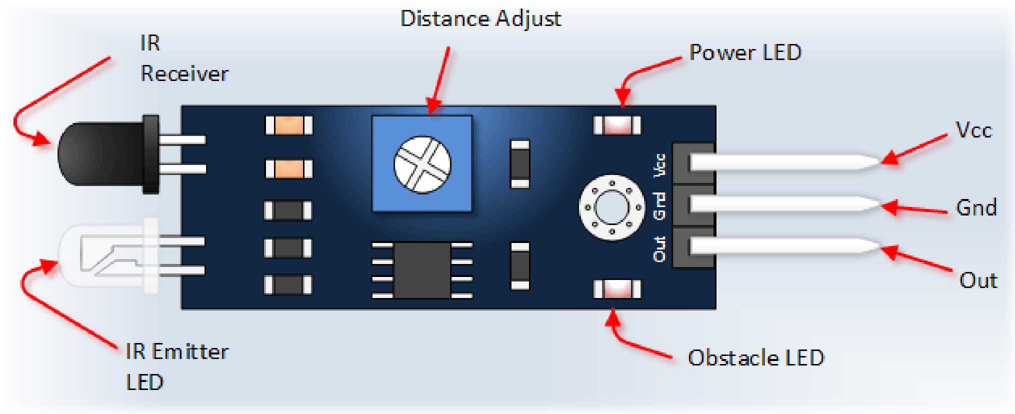
Arduino Infrared Collision Avoidance

This is yet another one of those modules with cool possibilities. You could for example, sound an alarm when something got too close or you could change the direction of a robot or vehicle.

The device consists of an Infrared Transmitter, an Infrared Detector, and support circuitry. It only requires three connections. When it detects an obstacle within range it will send an output low.

IR Obstacle Detection Module Pin Outs

The drawing and table below identify the function of module pin outs, controls and indicators.

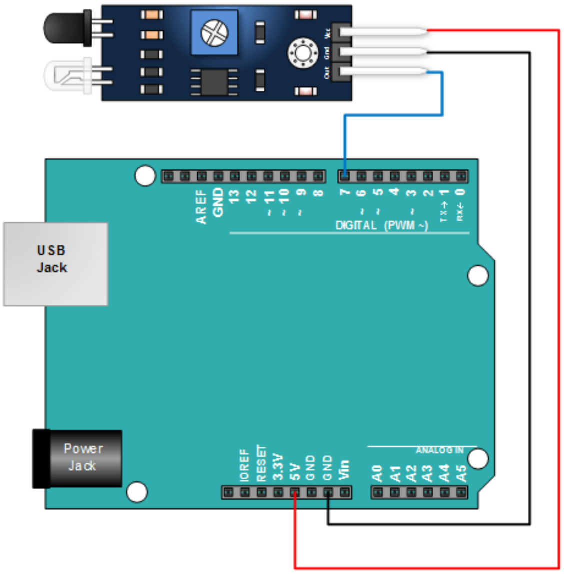


Pin, Control Indicator	Description
Vcc	3.3 to 5 Vdc Supply Input
Gnd	Ground Input
Out	Output that goes low when obstacle is in range
Power LED	Illuminates when power is applied
Obstacle LED	Illuminates when obstacle is detected
Distance Adjust	Adjust detection distance. CCW decreases distance. CW increases distance.
IR Emitter	Infrared emitter LED
IR Receiver	Infrared receiver that receives signal transmitted by Infrared emitter.

Arduino IR Obstacle Collision Module Tutorial

Connect the Arduino to the Detection Module

Use the picture below. It only requires three wires.

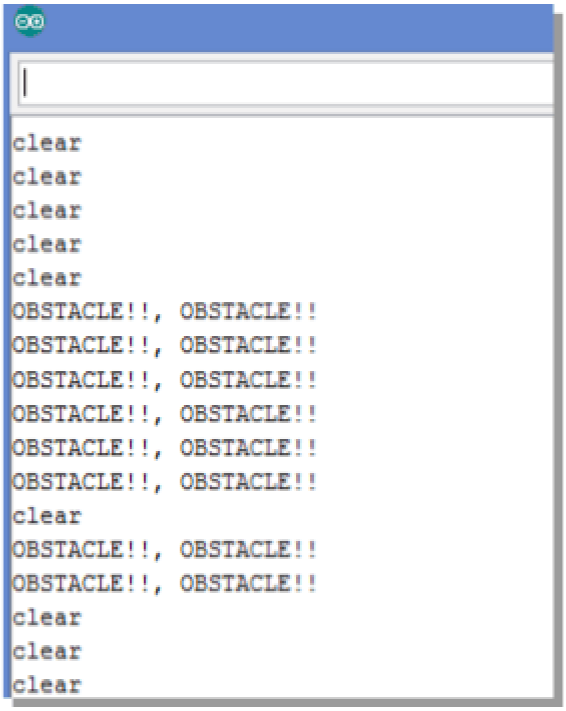


Copy, Paste and Upload the Sample Sketch

```
// IR Obstacle Collision Detection Module
// Henry's Bench

int LED = 13; // Use the onboard Uno LED
int isObstaclePin = 7;  // This is our input pin
int isObstacle = HIGH;  // HIGH MEANS NO OBSTACLE

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(isObstaclePin, INPUT);
  Serial.begin(9600);

}

void loop() {
  isObstacle = digitalRead(isObstaclePin);
  if (isObstacle == LOW)
  {
    Serial.println("OBSTACLE!!, OBSTACLE!!");
    digitalWrite(LED, HIGH);
  }
  else
  {
    Serial.println("clear");
    digitalWrite(LED, LOW);
  }
  delay(200);
}
```

Test the Tutorial Sketch

Move your hand towards the IR LEDs. As you near them, the Output LED on the module and the LED for pin 13 on your Arduino will illuminate. Open your serial monitor and vary the distance of your hand while viewing the serial monitor.   The output should look like the picture below:



```
clear
clear
clear
clear
clear
OBSTACLE!!, OBSTACLE!!
OBSTACLE!!, OBSTACLE!!
OBSTACLE!!, OBSTACLE!!
OBSTACLE!!, OBSTACLE!!
OBSTACLE!!, OBSTACLE!!
OBSTACLE!!, OBSTACLE!!
clear
OBSTACLE!!, OBSTACLE!!
OBSTACLE!!, OBSTACLE!!
clear
clear
clear
```