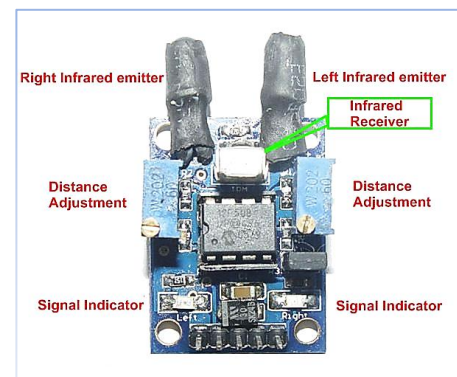

ARDUINO INFRARED OBSTACLE-AVOIDING SENSOR

I am investigating a number of different sensor alternatives to determine cost/benefit as measured against my needs. The device I evaluated in this case is as the title above reads, I purchased it from a distributor in China: www.elechouse.com.

OVERVIEW

The Device (rather than Arduino Infrared...) is designed to send out infrared signals and detect the response to determine the existence of something with the right heat signature within a certain range. The Device has two emitters (left and right) If it finds something, the Device flashes, blinks or turns on the LED on the appropriate.

At the same time, by driving the EN pin HIGH (it latches for you non-Arduino savvy like me), the Left_output and Right_output pins will indicate HIGH if something has been detected. Therefore, by writing HIGH to the EN pin in **setup** and subsequently reading the Left_output or Right_output pins, you can determine if they are HIGH.

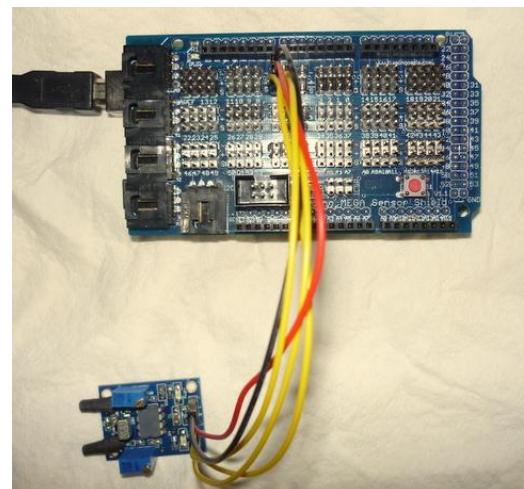


CONNECTING TO ARDUINO

Oddly enough, I used the Arduino to interface to the Device; more specifically, the Arduino Mega 2560. I chose to map the EN pin to PWM/Digital pin 7, the Left_output pin to PWM/Digital pin 6, and the Right_output pin to PWM/Digital pin 5.

[plug] I purchased the Arduino MEGA Special Sensor Shield from elechouse when I bought the Device. Boys and girls, this thing is TOPS! The nice people at elechouse also sell these nifty little wires with single pole connectors on the end. Great for popping on one of the many headers protruding from the board in all their finery! Saves me hours! [\/plug]

That's about it for the hardware. Now on to the tough stuff! Software.



SOFTWARE

I hope this code speaks enough for itself.

```
// Module to control the Arduino Infrared Object Avoiding Sensor
// Created 3/1/2011

int sensorPinEN = 7;    // A5 - A7 respectively are assigned to
int sensorPinLO = 6;    // Right_output, Left_output and NEN
int sensorPinRO = 5;    //

int ledPin = 13;        // select the pin for the LED
int sensorValue = 0;    // variable to store the value coming from the sensor

void setup() {
  // Initialize Arduino Serial which will be used to communicate
  // various status to the developer.
  Serial.begin(9600);

  // Set the pins correctly
  pinMode(sensorPinEN, OUTPUT);
  pinMode(sensorPinLO, INPUT);
  pinMode(sensorPinRO, INPUT);

  // drive the output
  digitalWrite(sensorPinEN, HIGH);
}

void loop() {
  int charIn;

  // drive the output
  digitalWrite(sensorPinEN, HIGH);

  // read the value from the sensor:
  if (digitalRead(sensorPinRO) == HIGH)
    // sensors sees something on the right side
    Serial.println("Sensing Right");

  // drive the output
  // digitalWrite(sensorPinEN, HIGH);

  if (digitalRead(sensorPinLO) == HIGH)
    // sensors sees something on the right side
    Serial.println("Sensing Left");

  delay(1000);
}
```