TSL2561 Luminosity Sensor

Created by Ladyada
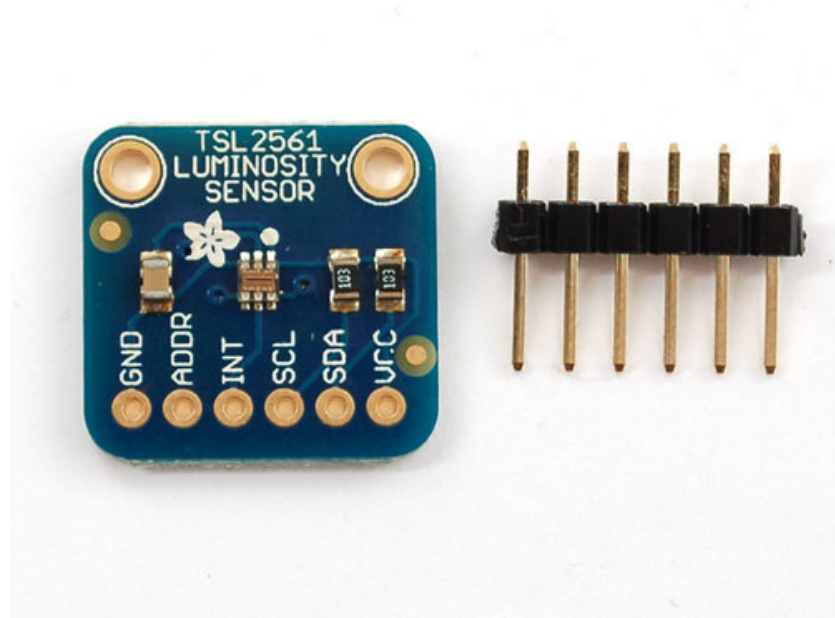
Guide Contents

Overview

The TSL2561 luminosity sensor is an advanced digital light sensor, ideal for use in a wide range of light situations. Compared to low cost CdS cells, this sensor is more precise, allowing for exact Lux calculations and can be configured for different gain/timing ranges to detect light ranges from up to 0.1 - 40,000+ Lux on the fly. The best part of this sensor is that it contains **both infrared and full spectrum diodes**! That means you can seperately measure infrared, full-spectrum or human-visible light. Most sensors can only detect one or the other, which does not accurately represent what human eyes see (since we cannot perceive the IR light that is detected by most photo diodes).
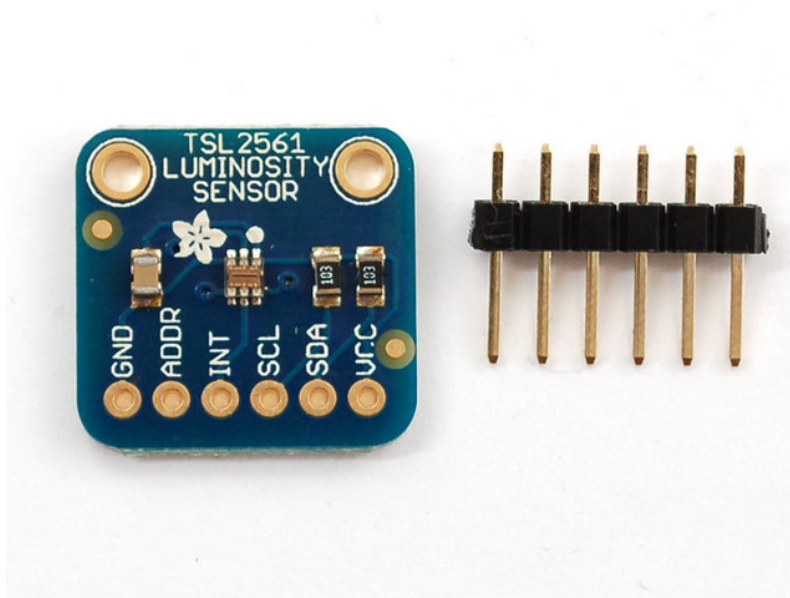
The sensor has a digital (i2c) interface. You can select one of three addresses so you can have up to three sensors on one board - each with a different i2c address. The built in ADC means you can use this with any microcontroller, even if it doesn't have analog inputs. The current draw is extremely low, so its great for low power data-logging systems. about 0.5mA when actively sensing, and less than 15 uA when in powerdown mode.
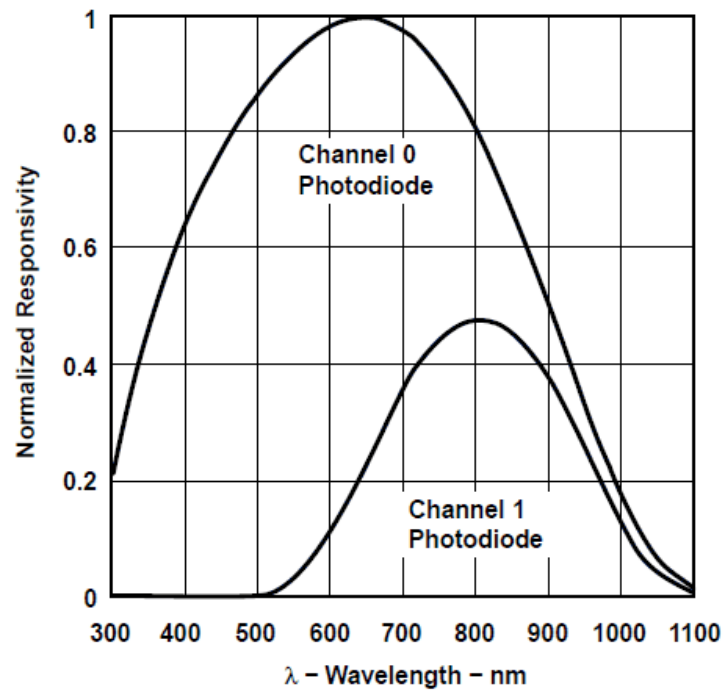
SPECTRAL RESPONSIVITY

Figure 4

Some Stats

- **Approximates Human eye Response**
- **Precisely Measures Illuminance in Diverse Lighting Conditions**
- **Temperature range:** -30 to 80 *C
- **Dynamic range (Lux):** 0.1 to 40,000 Lux
- **Voltage range:** 2.7-3.6V
- **Interface:** I2C

Pick one up today from the Adafruit shop! (http://adafru.it/439)

Wiring the TSL2561 Sensor

This is an easy sensor to get started with. The breakout board comes with a 6-pin header strip that you can use to plug the sensor into a breadboard or perfboard. Simply plug the header into a solderless breadboard with the long pins down and short pins up. Place the sensor on top so each pad has a header pin in it and solder the two together.
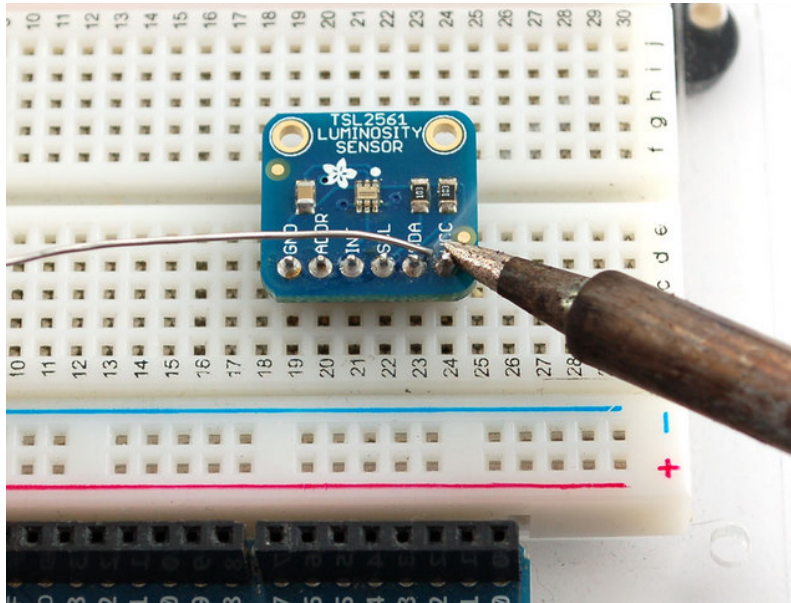

Next we will connect it to our microcontroller. In this case we'll be using an Arduino but nearly any microcontroller can be used by adapting our code

Connect the **VCC** pin to a **3.3V** power source. The sensor cannot be used with anything higher than 3.3V so don't use a 5V supply! Connect **GND** to the ground pin.

Connect the **i2c SCL clock** pin to your i2c clock pin. On the classic Arduino Uno/Duemilanove/Diecimila/etc this is **Analog pin #5**

Connect the **i2c SDA data** pin to your i2c data pin. On the classic Arduino Uno/Duemilanove/Diecimila/etc this is **Analog pin #4**

Unfortunately, the i2c lines on most microcontrollers are fixed so you're going to have to stick with those pins.
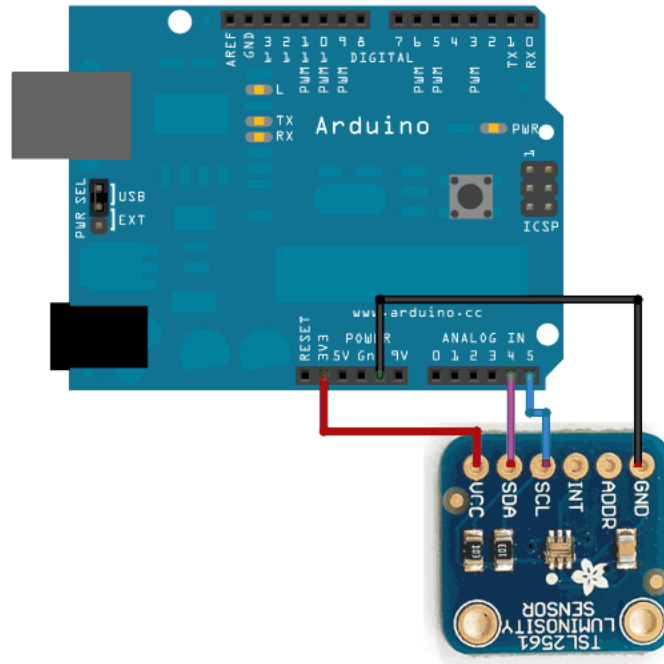
You may be wondering, how is it OK to connect a 3.3V chip like the TSL2561 to 5.0V data pins like the Arduino? Isn't that bad? Well, in this specific case its OK. I2c uses pullup lines to the 3.3V power pin, so the data is actually being sent at 3.3V. As long as all the sensors/device on the i2c bus are running on 3.3V power, we're fine. However, don't use a 5.0v powered i2c device (like the DS1307) with pullups at the same time as a 3.3V device like the TSL2561! If you want to use this sensor with a datalogger that uses the DS1307, remove any/all of the pullup resistors from the DS1307 SDA/SCL pins. The pullups built into the TSL2561 will then be active and keep the voltage at 3.3V which is safe for both the RTC and the sensor.

You don't need to connect the **ADDR** (i2c address change) or **INT** (interrupt output) pins.

The **ADDR** pin can be used if you have an i2c address conflict, to change the address. Connect it to ground to set the address to **0x29**, connect it to 3.3V (vcc) to se t the address to **0x49** or leave it floating (unconnected) to use address **0x39**.

The **INT** pin is an ouput *from* the sensor used when you have the sensor configured to signal when the light level has changed. We don't have that code written in this tutorial so you don't have to use it. If you do end up using it, use a 10K-100K pullup from **INT** to 3.3V (vcc)

Using the TSL2561 Sensor

To use this sensor and calculate Lux, there's a lot of very hairy and unpleasant math. You can check out the math in the datasheet (http://adafru.it/aJR) but really, its not intuitive or educational - its just how the sensor works. So we took care of all the icky math and wrapped it up into a nice Arduino library.
 (http://adafru.it/aZ9)

You can find the Arduino library repository on github (http://adafru.it/aZ9) To install it, click the button below to download the compressed ZIP file then install it. This guide (http://adafru.it/aYM) will help you with the install process if you have never installed an Arduino library.

> **Download Adafruit_TSL2561 V2 Library**
>
> http://adafru.it/d7d

You will also need the Adafruit Sensor Library (http://adafru.it/aZm) if you do not already have it installed. You can download it by clicking this button to get the compressed Zip. Then install as you did above.
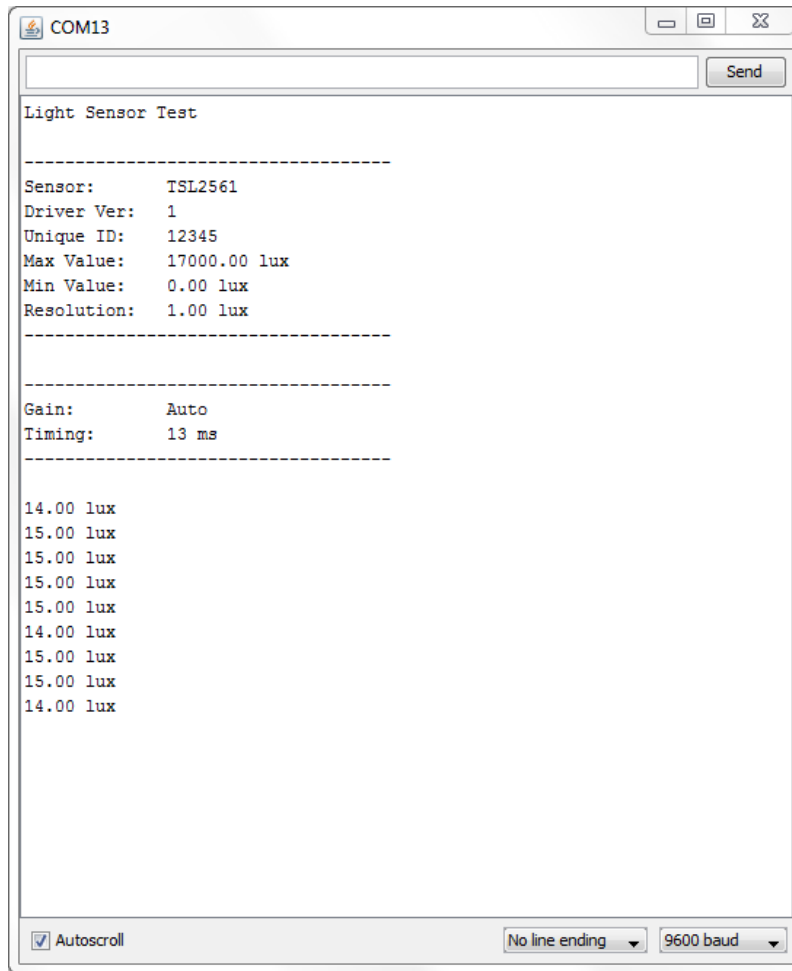
> **Download Adafruit_Sensor Library**
>
> http://adafru.it/cMO

Restart the IDE

Now you can run the **File->Examples->Adafruit_TSL2561->sensorapi** example program which will read and calculate the lux readings for you.

Open up the serial monitor at 9600 baud to see the measurements. Use a lamp or your hand to illuminate/shade the sensor to see the values change.

```
COM13                                          _ □ ⊠
                                                   Send

Light Sensor Test

------------------------------------
Sensor:       TSL2561
Driver Ver:   1
Unique ID:    12345
Max Value:    17000.00 lux
Min Value:    0.00 lux
Resolution:   1.00 lux
------------------------------------


------------------------------------
Gain:         Auto
Timing:       13 ms
------------------------------------

14.00 lux
15.00 lux
15.00 lux
15.00 lux
15.00 lux
14.00 lux
15.00 lux
15.00 lux
14.00 lux




☑ Autoscroll              No line ending ▾   9600 baud ▾
```

The library is fairly simple to use. The first line of code in the example is the 'constructor' where you can supply the **I2C ADDR** (in case you want to change it), and a unique ID to attach to this sensor (you can just leave this to the default value of 12345 for now). By modifying the I2C address we can have up to three TSL2561 sensors connected on the same board:

```
// The address will be different depending on whether you leave
// the ADDR pin float (addr 0x39), or tie it to ground or vcc. In those cases
// use TSL2561_ADDR_LOW (0x29) or TSL2561_ADDR_HIGH (0x49) respectively
Adafruit_TSL2561 tsl = Adafruit_TSL2561(TSL2561_ADDR_FLOAT, 12345);
```

Next up, you will want to configure the sensor with the **gain** and **integration time**.

You can have either a gain of 0 (no extra gain, good in low light situations) or a gain of 16 which will boost the light considerably in dim situations.

You can also change the integration time, which is how long it will collect light data for. The longer the integration time, the more precision the sensor has when collecting light samples.

New to v2.0 of the driver, there is also an **auto-gain** option that is useful when measuring in mixed lighting-situations. This will automatically enable or disable the gain depending on the light level. This is still an experimental feature and the trigger levels to switch may need to be tweeked, but this should be useful to collect light both indoors and outdoors without having to

change the code yourself.

```
/*************************************************************************/
/*
    Configures the gain and integration time for the TSL2561
*/
/*************************************************************************/
void configureSensor(void)
{
  /* You can also manually set the gain or enable auto-gain support */
  // tsl.setGain(TSL2561_GAIN_1X);      /* No gain ... use in bright light to avoid sensor saturation */
  // tsl.setGain(TSL2561_GAIN_16X);     /* 16x gain ... use in low light to boost sensitivity */
  tsl.enableAutoRange(true);           /* Auto-gain ... switches automatically between 1x and 16x */

  /* Changing the integration time gives you better sensor resolution (402ms = 16-bit data) */
  tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_13MS);      /* fast but low resolution */
  // tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_101MS); /* medium resolution and speed   *
  // tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_402MS); /* 16-bit data but slowest conversic

  /* Update these values depending on what you've set above! */
  Serial.println("------------------------------------");
  Serial.print  ("Gain:         "); Serial.println("Auto");
  Serial.print  ("Timing:       "); Serial.println("13 ms");
  Serial.println("------------------------------------");
}
```

By default, the driver will return light in standard SI lux units, which are a result of some complex calculations based on both photo diodes on the TSL2561 (one for full spectrum and one for IR). The sensitivity of the two diodes can be seen in the chart below:
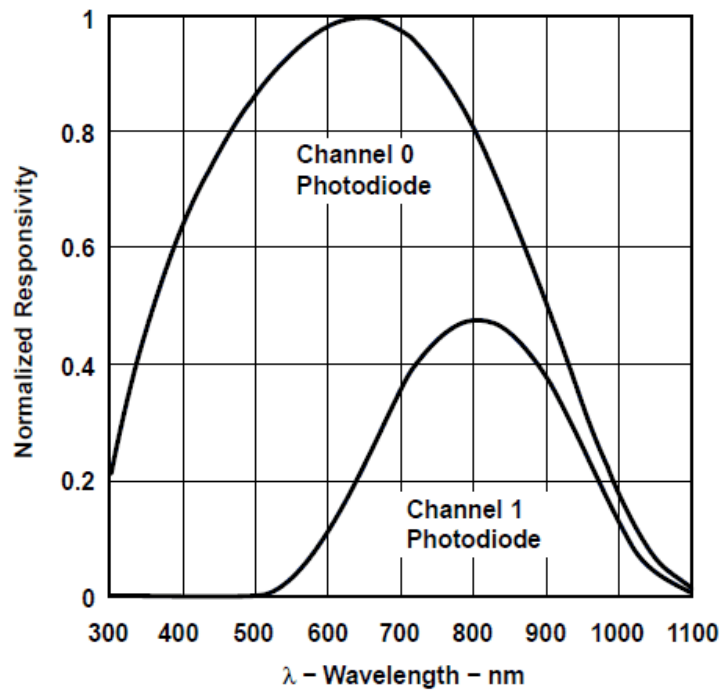
SPECTRAL RESPONSIVITY

Figure 4

When you're ready to get your measurement in standard SI lux units, simply call **getEvent** with a reference to the 'sensors_event_t' object where the sensor data will be stored. This example assumes we are using the 'tsl' instance of Adafruit_TSL2561 at the top of the example code:

```
/* Get a new sensor event */
sensors_event_t event;
tsl.getEvent(&event);

/* Display the results (light is measured in lux) */
if (event.light)
{
  Serial.print(event.light); Serial.println(" lux");
}
else
{
  /* If event.light = 0 lux the sensor is probably saturated
     and no reliable data could be generated! */
  Serial.println("Sensor overload");
}
```

This function will return a reading in SI lux units, which is probably the easiest unit to understand when working with light.

If you wish to manually read the individual photo diodes, though, you can still do this in the latest library by calling the **getLuminosity** function, and passing in two variables where the sensor data will be stored:

```
uint16_t broadband = 0;
uint16_t infrared = 0;

/* Populate broadband and infrared with the latest values */
getLuminosity (&broadband, &infrared);
```

That's it! The example should be easy to understand and work into your own projects from here!

Downloads

- Datasheet (http://adafru.it/aJS)
- TSL2561 Driver v2.0 (Unified Sensor Driver) (http://adafru.it/aZ9) - See http://learn.adafruit.com/tsl2561/use (http://adafru.it/c7O) for installation instructions
- TSL2561 Driver v1.0 on github (obsolete!) (http://adafru.it/aJT)

Buy a TSL2561 Sensor

Buy a TSL2561 Sensor (http://adafru.it/439)