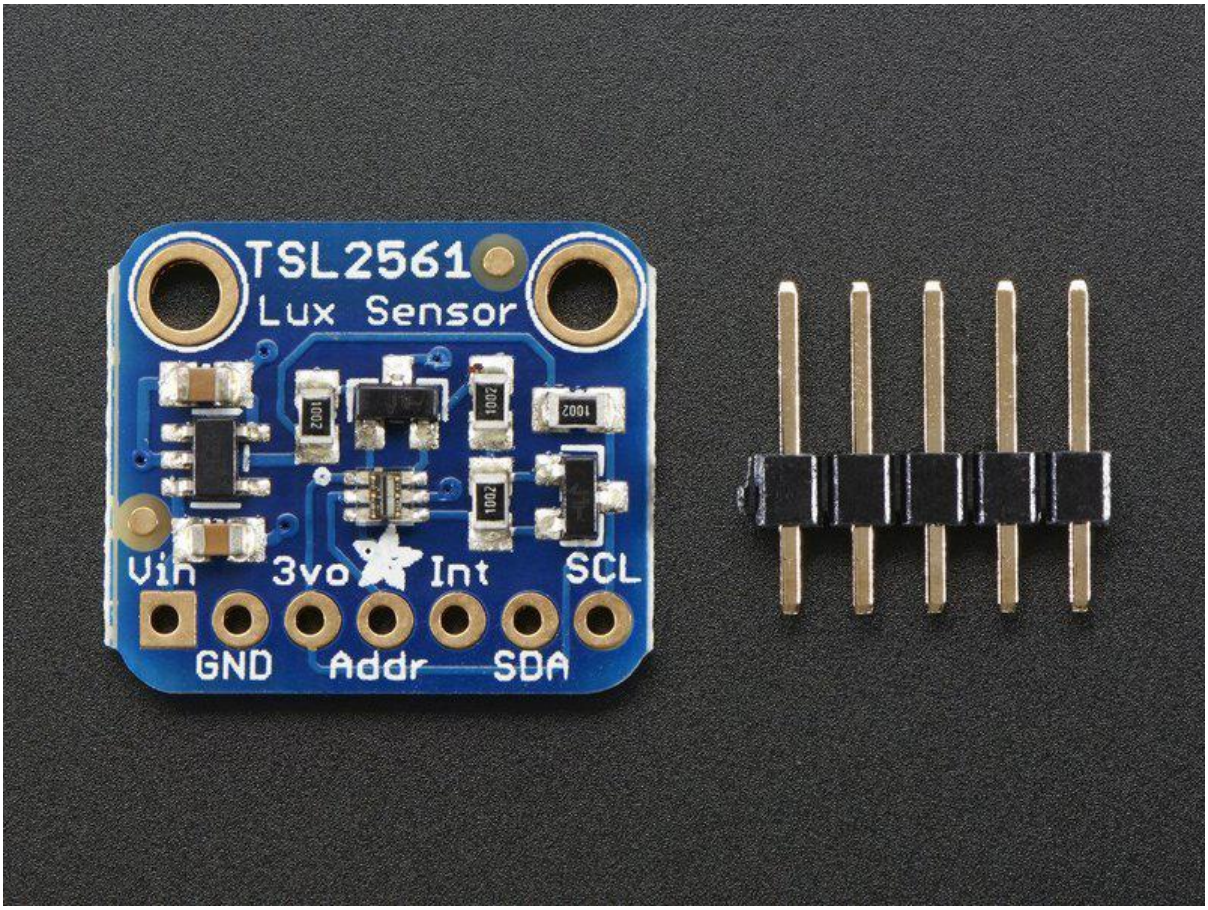
TSL2561 Luminosity Sensor

Created by lady ada

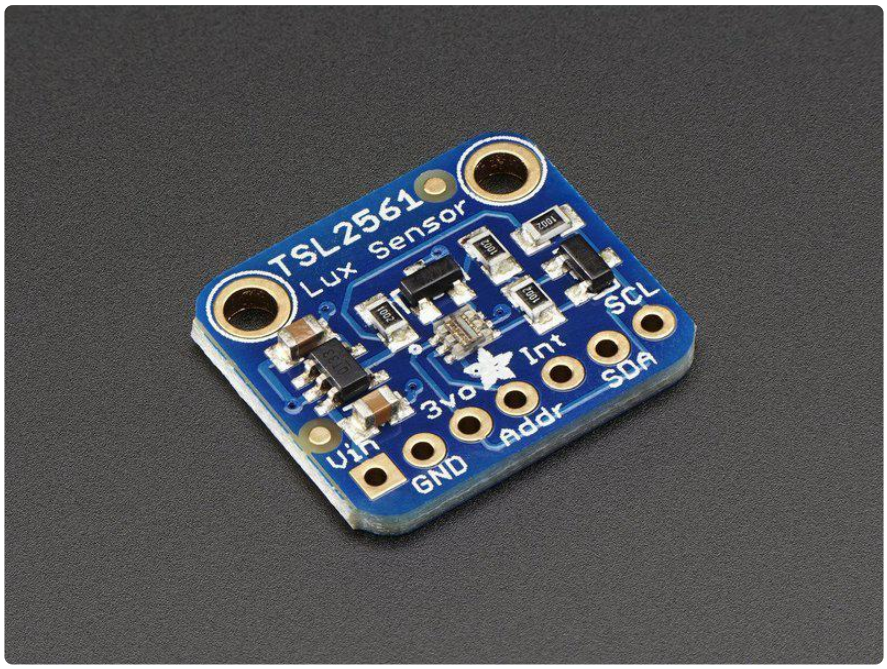


https://learn.adafruit.com/tsl2561

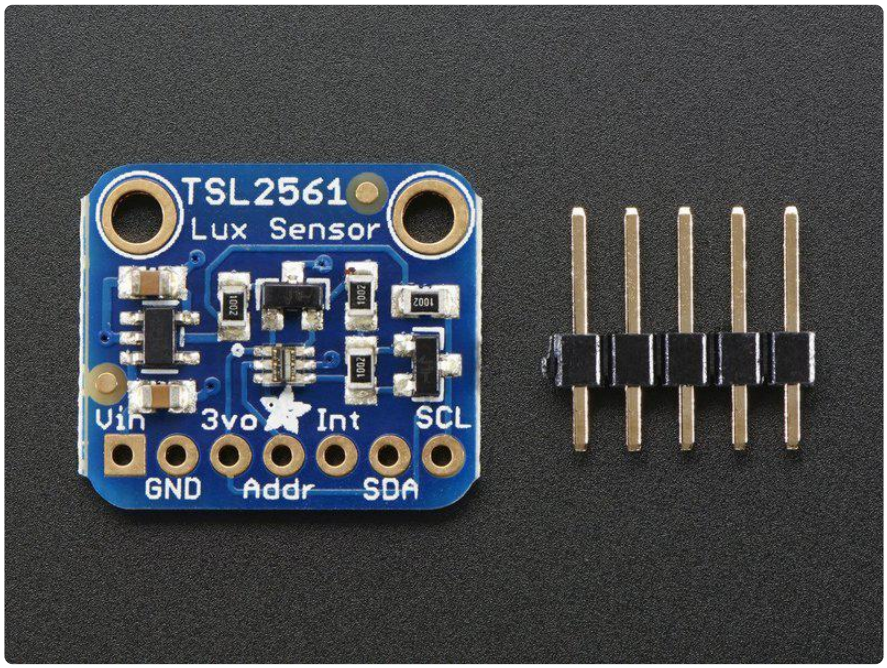
Last updated on 2022-12-01 01:49:22 PM EST

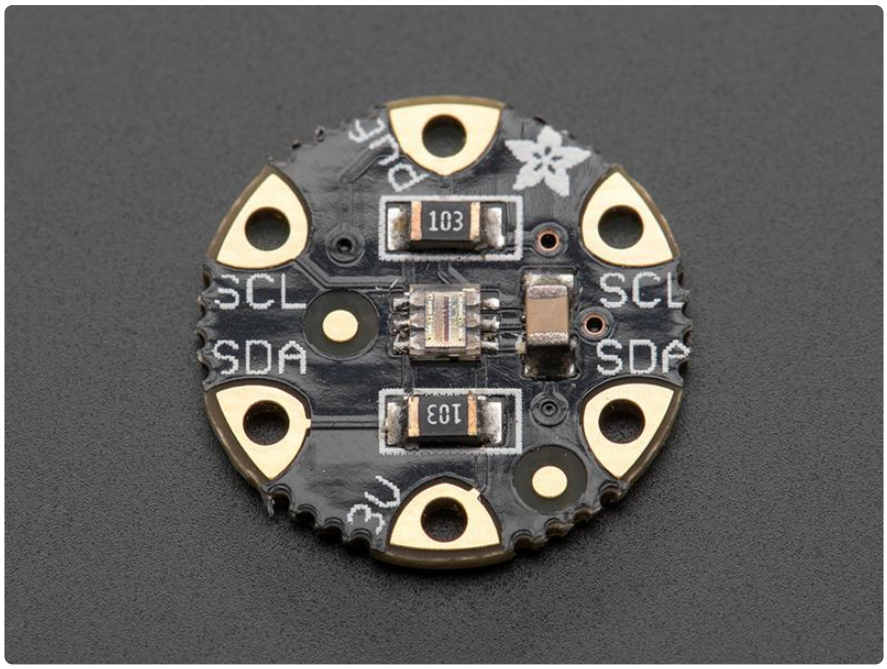
Table of Contents

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.

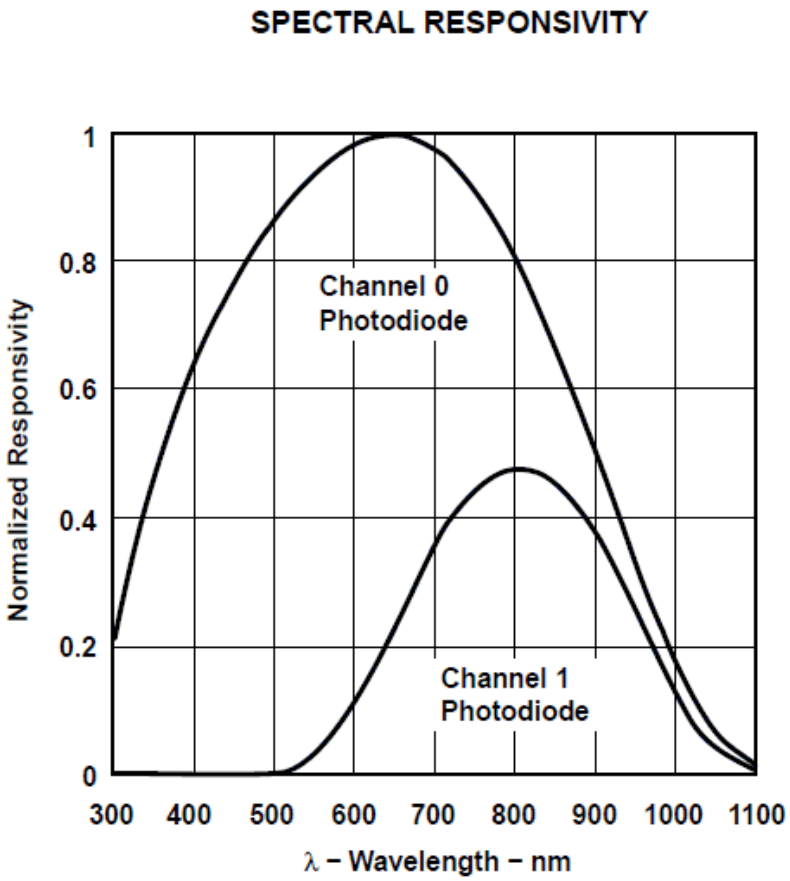


Figure 4

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

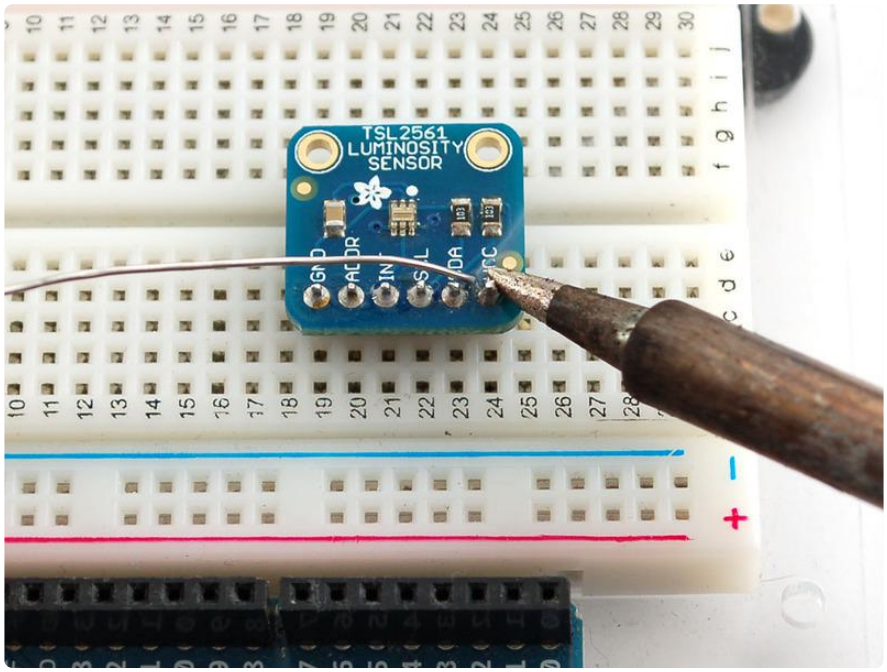
Pick one up today from the Adafruit shop! (http://adafru.it/439)

Wiring the TSL2561 Sensor

Breakout Board Prep

This is an easy sensor to get started with. If you have the Breakout board version, it 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.

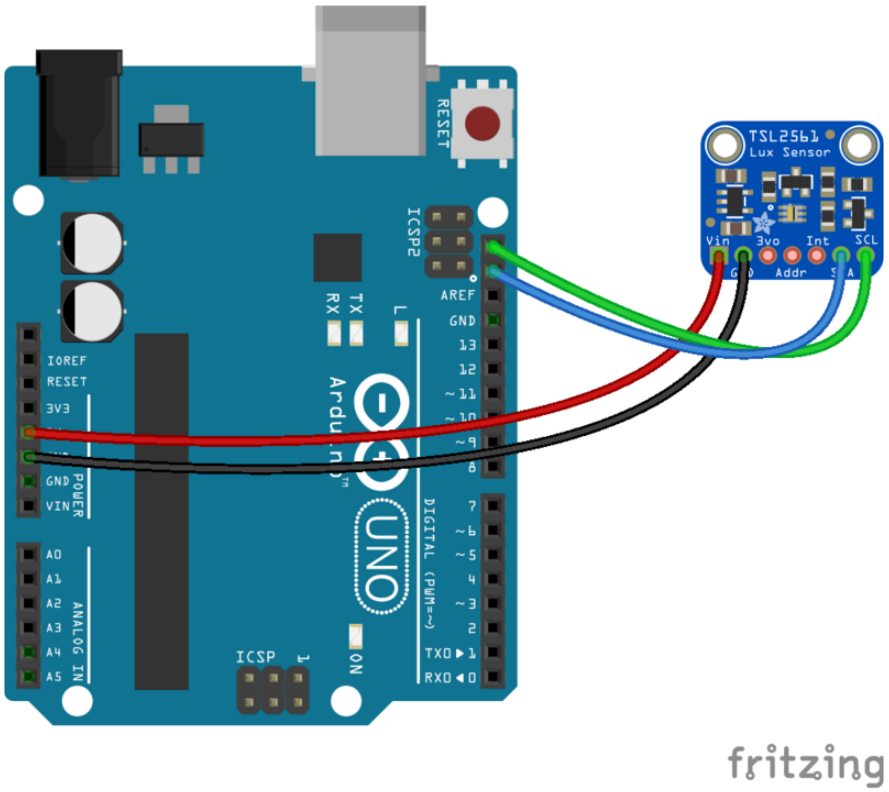


Wiring up the sensor

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 or 5v power source (Whatever the logic level of your microcontroller is!)
- 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

The i2c lines on most microcontrollers are fixed so you're going to have to stick with those pins.



uno + tsl2561 Fritzing diagram

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)

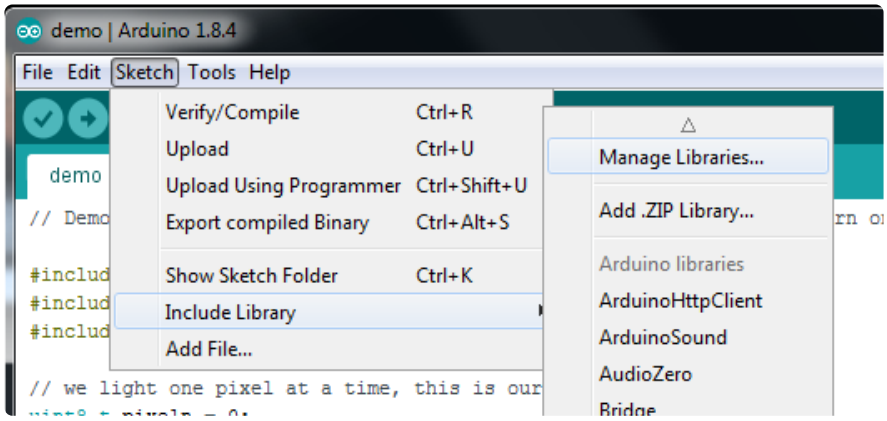
Arduino Code

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 () 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.

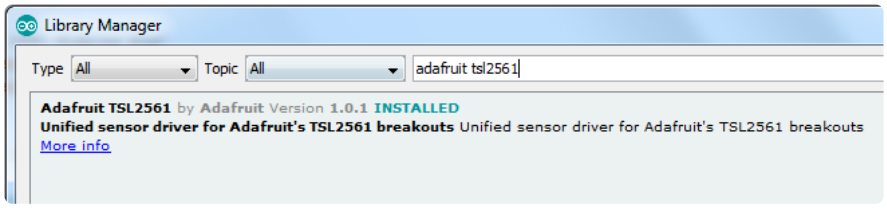
Install Adafruit_TSL2561 library

To begin reading sensor data, you will need to install the Adafruit_TSL2561 library (code on our github repository) (). It is available from the Arduino library manager so we recommend using that.

From the IDE open up the library manager...



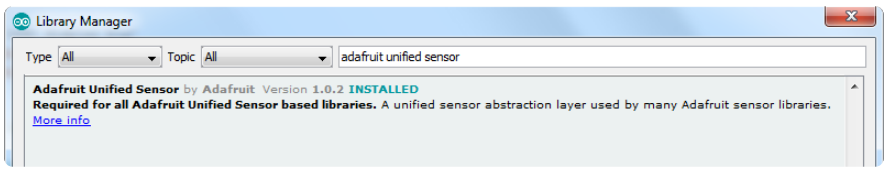
And type in adafruit tsl2561 to locate the library. Click Install



Install Adafruit_Sensor

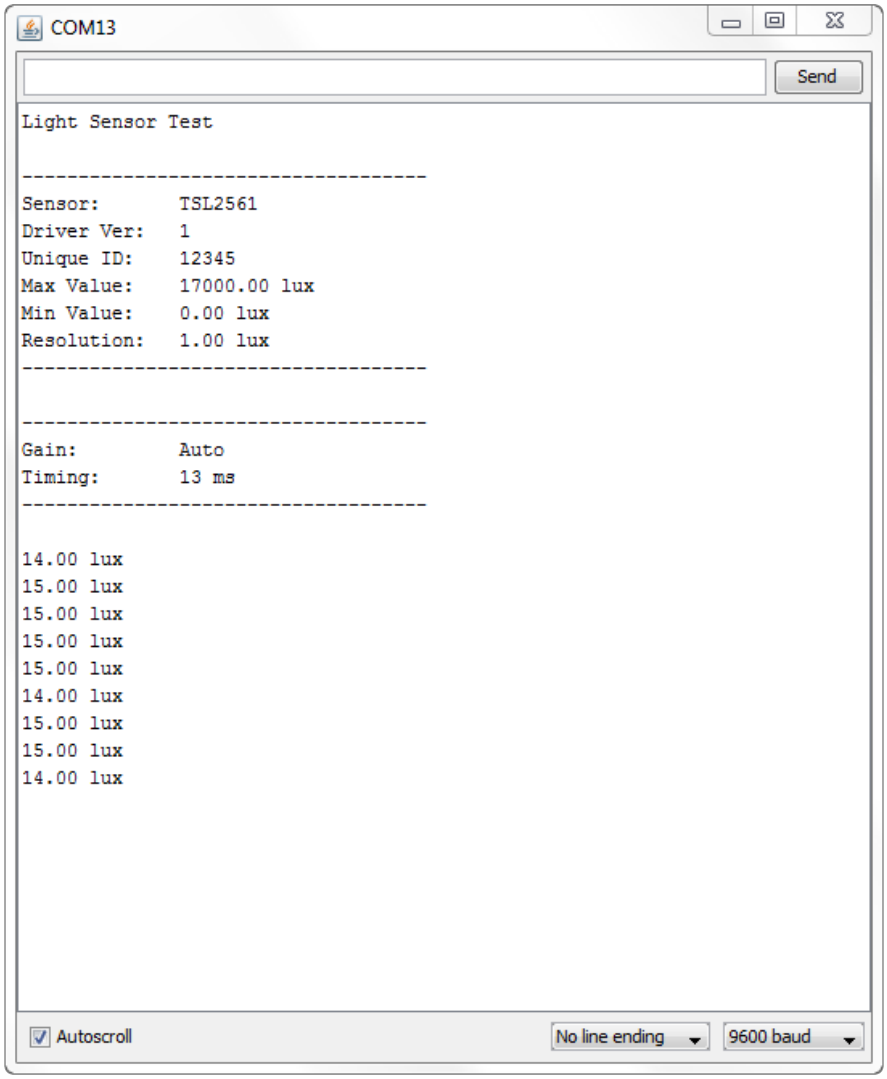
The TSL2561 library uses the Adafruit_Sensor support backend () so that readings can be normalized between sensors.

Search the library manager for Adafruit Unified Sensor and install that too (you may have to scroll a bit)



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.



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 tweaked, 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 conversions */

  /* 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:

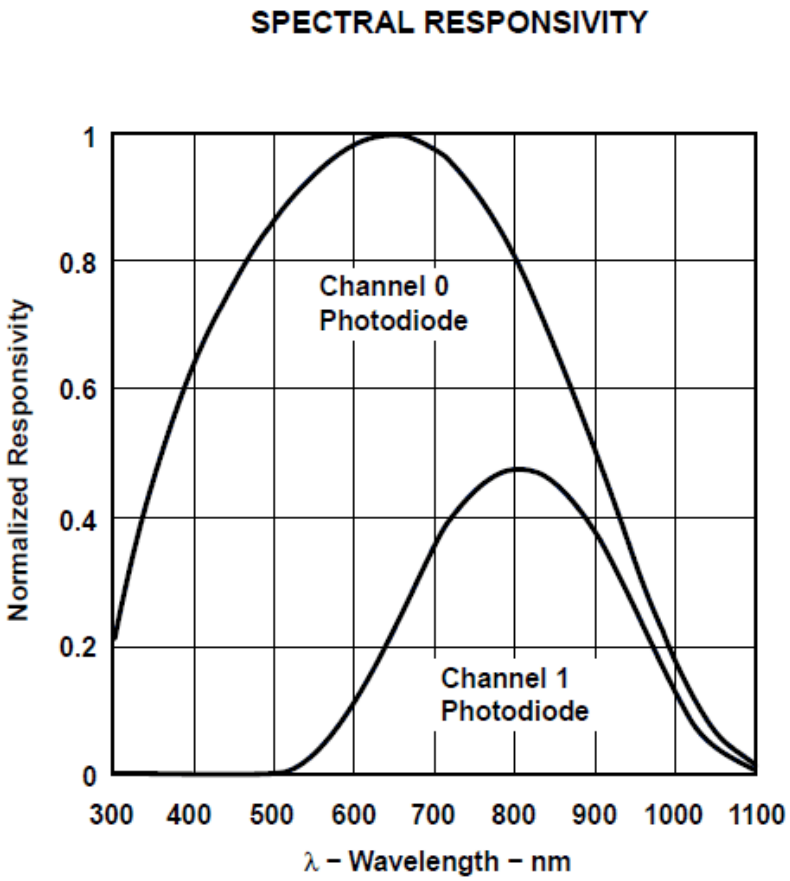


Figure 4

When you're ready to get your measurement in standard SI lux units, simply call getEv ent 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(&amp;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 (&amp;broadband, &amp;infrared);
```

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

Arduino Library Docs

Arduino Library Docs ()

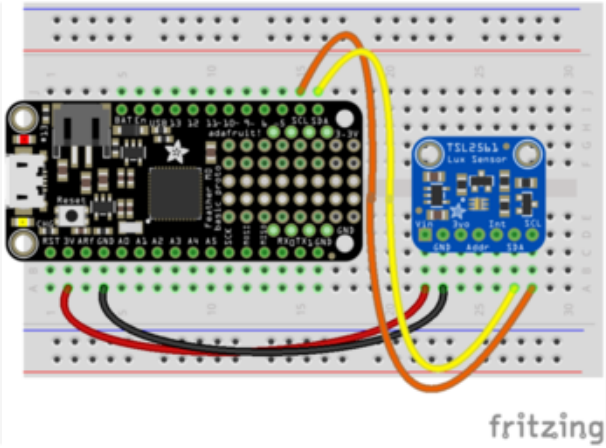
Python & CircuitPython

It's easy to use the TSL2561 sensor with Python and CircuitPython, and the Adafruit CircuitPython TSL2561 () module. This module allows you to easily write Python code that reads the luminosity and more from the sensor.

You can use this sensor with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library ().

CircuitPython Microcontroller Wiring

First wire up a TSL2561 to your board exactly as shown on the previous pages for Arduino using an I2C connection. Here's an example of wiring a Feather M0 to the sensor with I2C:

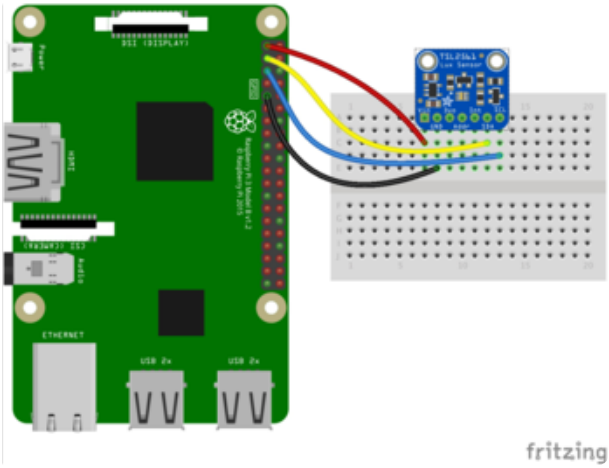


Board 3V to sensor VIN
Board GND to sensor GND
Board SCL to sensor SCL
Board SDA to sensor SDA

Python Computer Wiring

Since there's dozens of Linux computers/boards you can use we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported ().

Here's the Raspberry Pi wired with I2C:



Pi 3V3 to sensor VIN
Pi GND to sensor GND
Pi SCL to sensor SCK
Pi SDA to sensor SDA

CircuitPython Installation of TSL2561 Library

Next you'll need to install the Adafruit CircuitPython TSL2561 () library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython () for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (). For example the Circuit Playground Express guide has a great page on how to install the library bundle () for both express and non-express boards.

Remember for non-express boards like the Trinket M0, Gemma M0, and Feather/Metro M0 basic you'll need to manually install the necessary libraries from the bundle:

- adafruit_tsl2561.mpy
- adafruit_bus_device

You can also download the adafruit_tsl2561.mpy from its releases page on Github ().

Before continuing make sure your board's lib folder or root filesystem has the adafruit_tsl2561.mpy, and adafruit_bus_device files and folders copied over.

Next connect to the board's serial REPL ()so you are at the CircuitPython >>> prompt.

Python Installation of TSL2561 Library

You'll need to install the Adafruit_Blinka library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready ()!

Once that's done, from your command line run the following command:

- `sudo pip3 install adafruit-circuitpython-tsl2561`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython and Python Usage

To demonstrate the usage of the sensor we'll initialize it and read the luminosity from the board's Python REPL. Run the following code to import the necessary modules and initialize the I2C connection with the sensor:

```
import board
import busio
import adafruit_tsl2561
i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_tsl2561.TSL2561(i2c)
```

Now you're ready to read values from the sensor using any of these properties:

- lux - The computed light lux value measured by the sensor.
- broadband - The broadband channel value.
- infrared - The infrared channel value.
- luminosity - A 2-tuple of broadband and infrared channel values.

```
print('Lux: {}'.format(sensor.lux))
print('Broadband: {}'.format(sensor.broadband))
print('Infrared: {}'.format(sensor.infrared))
print('Luminosity: {}'.format(sensor.luminosity))
```

```
>>> print('Lux: {}'.format(sensor.lux))
Lux: 3.424
>>> print('Broadband: {}'.format(sensor.broadband))
Broadband: 40
>>> print('Infrared: {}'.format(sensor.infrared))
Infrared: 22
>>> print('Luminosity: {}'.format(sensor.luminosity))
Luminosity: (40, 22)
>>>
```

In addition there are some properties you can both read and write to change how the sensor works:

- gain - Get and set the gain of the light sensor. A value of 0 is low gain mode, and a value of 1 is high gain / 16x mode.
- integration_time - Get and set the integration time of the sensor. A value 0 is 13.7ms, 1 is 101ms, 2 is 402ms, and 3 is manual mode.

```
# Set high gain mode.
sensor.gain = 1
# Set 402ms integration time.
sensor.integration_time = 2
```

```
>>> sensor.gain = 1
>>> sensor.integration_time = 2
>>> print('Lux: {}'.format(sensor.lux))
Lux: 3.4022
>>>
```

That's all there is to using the TSL2561 sensor with CircuitPython!

Here's a complete example of reading the light value every second. Save this as a co de.py on your board and look at the output in the serial monitor:

Full Example Code

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import busio
import adafruit_tsl2561

# Create the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)

# Create the TSL2561 instance, passing in the I2C bus
tsl = adafruit_tsl2561.TSL2561(i2c)

# Print chip info
print("Chip ID = {}".format(tsl.chip_id))
print("Enabled = {}".format(tsl.enabled))
print("Gain = {}".format(tsl.gain))
print("Integration time = {}".format(tsl.integration_time))

print("Configuring TSL2561...")

# Enable the light sensor
tsl.enabled = True
time.sleep(1)

# Set gain 0=1x, 1=16x
tsl.gain = 0

# Set integration time (0=13.7ms, 1=101ms, 2=402ms, or 3=manual)
tsl.integration_time = 1

print("Getting readings...")

# Get raw (luminosity) readings individually
broadband = tsl.broadband
infrared = tsl.infrared

# Get raw (luminosity) readings using tuple unpacking
# broadband, infrared = tsl.luminosity

# Get computed lux value (tsl.lux can return None or a float)
lux = tsl.lux

# Print results
print("Enabled = {}".format(tsl.enabled))
print("Gain = {}".format(tsl.gain))
print("Integration time = {}".format(tsl.integration_time))
print("Broadband = {}".format(broadband))
print("Infrared = {}".format(infrared))
if lux is not None:
    print("Lux = {}".format(lux))
else:
    print("Lux value is None. Possible sensor underrange or overrange.")

# Disble the light sensor (to save power)
tsl.enabled = False
```

Python Docs

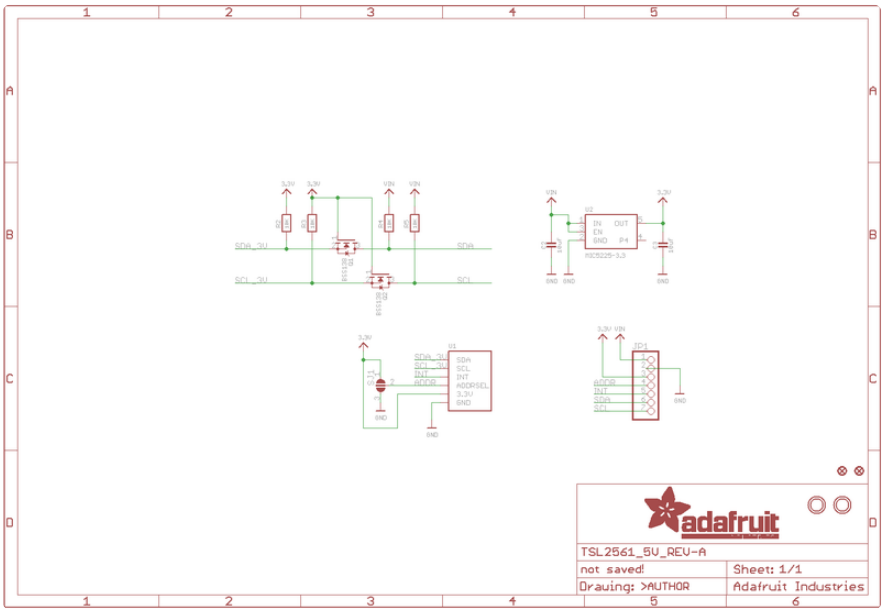
Python Docs ()

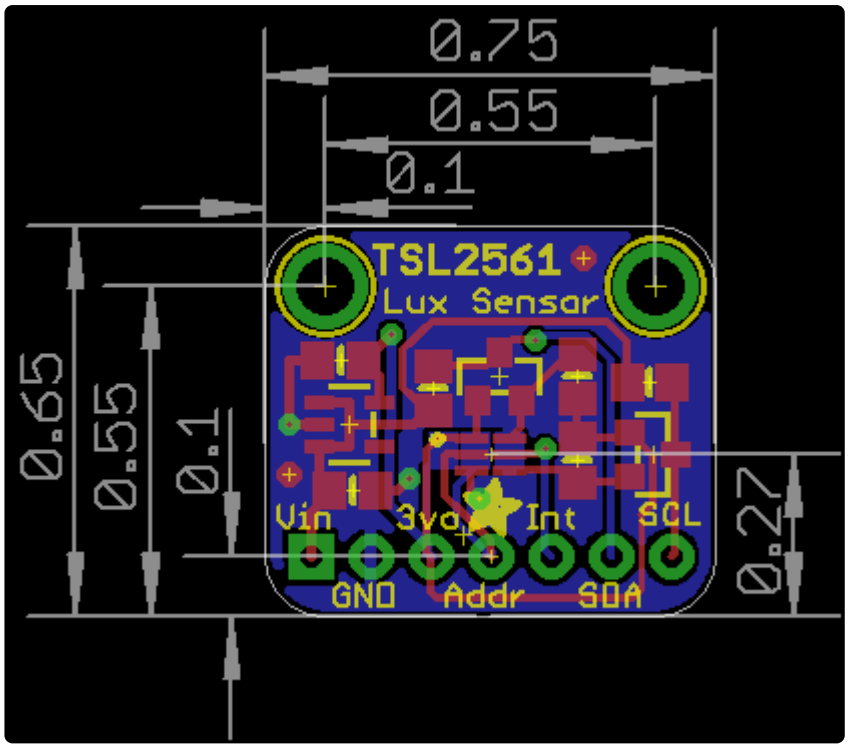
Downloads

Files

- TSL2561 Datasheet ()
- TSL2561 Driver v2.0 (Unified Sensor Driver) () - See http://learn.adafruit.com/tsl2561/use () for installation instructions
- TSL2561 Driver v1.0 on github (obsolete!) ()
- Fritzing objects in the Adafruit Fritzing library ()
- EagleCAD PCB files for breakout version ()

Breakout Board Schematic & Fabrication Print




0.75
0.55
0.1
TSL2561
Lux Sensor
0.65
0.55
0.1
0.27
Vin
3Vo
Int
SCL
GND
Addr
SDA

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Adafruit:
439