

# Adafruit TSL2591 High Dynamic Range Digital Light Sensor

Created by lady ada



Last updated on 2019-07-26 07:29:34 PM UTC

### Overview





When the future is dazzlingly-bright, this ultra-high-range luminosity sensor will help you measure it. The TSL2591 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 188uLux up to 88,000 Lux on the fly.

The best part of this sensor is that it **contains both infrared and full spectrum diodes**! That means you can separately 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)



This sensor is much like the TSL2561 but with a wider range (and the interface code is different). This sensor has a massive 600,000,000:1 dynamic range! Unlike the TSL2561 you cannot change the I2C address either, so keep that in mind.



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 0.4mA when actively sensing, and less than 5 uA when in power-down mode.



## Pinouts

The TSL2591 is a I2C sensor. That means it uses the two I2C data/clock wires available on most microcontrollers, and can share those pins with other sensors as long as they don't have an address collision. For future reference, the I2C address is **0x29** and you *can't* change it!



### Power Pins:

- Vin this is the power pin. Since the chip uses 3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller e.g. for a 5V micro like Arduino, use 5V
- 3vo this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- GND common ground for power and logic

### (https://adafru.it/dGy)I2C Logic pins:

- SCL I2C clock pin, connect to your microcontrollers I2C clock line.
- SDA I2C data pin, connect to your microcontrollers I2C data line.

#### Other Pins:

• INT - this is the INTerrupt pin from the sensor. It can be programmed to do a couple different things by noodling with the i2c registers. For example trigger when a conversion is done, or when the light level has changed a lot, etc. We don't have library support for this pin

## Assembly







Prepare the header strip: Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down** 



Add the breakout board:

Place the breakout board over the pins so that the short pins poke through the breakout pads



### And Solder! Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our Guide to *Excellent Soldering* (https://adafru.it/aTk)).





You're done! Check your solder joints visually and continue onto the next steps



### Wiring & Test

You can easily wire this breakout to any microcontroller, we'll be using an Arduino. For another kind of microcontroller, just make sure it has I2C, then port the code - its pretty simple stuff!



#### (https://adafru.it/dBn)

- Connect Vin to the power supply, 3-5V is fine. Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V
- Connect **GND** to common power/data ground
- Connect the SCL pin to the I2C clock SCL pin on your Arduino. On an UNO & '328 based Arduino, this is also known as A5, on a Mega it is also known as digital 21 and on a Leonardo/Micro, digital 3
- Connect the SDA pin to the I2C data SDA pin on your Arduino. On an UNO & '328 based Arduino, this is also known as A4, on a Mega it is also known as digital 20 and on a Leonardo/Micro, digital 2

The TSL2591 has a default I2C address of BOTH **0x29** and **0x28** (which we don't use in code but it definitely is taken) and these cannot be changed!

### Install Adafruit\_TSL2591 library

To begin reading sensor data, you will need to install the Adafruit\_TSL2591 library (code on our github repository) (https://adafru.it/dGz). 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 tsl2591 to locate the library. Click Install

Type All  Topic All  Tsl2591	
Adafruit TSL2591 Library by Adafruit Version 1.0.3 INST Library for the TSL2591 digital luminosity (light) sensors More info	ALLED . Library for the TSL2591 digital luminosity (light) sensors.

We also have a great tutorial on Arduino library installation at: http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use (https://adafru.it/aYM)

### Install Adafruit\_Sensor

The TSL2591 library uses the Adafruit\_Sensor support backend (https://adafru.it/aZm) 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)



### Load Demo

Open up File->Examples->Adafruit\_TSL2591->tsl2591 and upload to your Arduino wired up to the sensor

Image: State in the state			-		
File       Edit       Sketch       Tools       Help       Adafruit_TEA5767       Adafruit_TEA5767         New       Ctrl+N       Adafruit_TFTLCD       Adafruit_TFTLCD       Adafruit_TC5947         Open       Sketchbook       Adafruit_TLC5947       Adafruit_TLC5947         Examples       Adafruit_TLC59711       Adafruit_TC59711         Close       Ctrl+W       Adafruit_TMP006       Adafruit_TMP007         Save       Ctrl+Shift+S       Adafruit_TRA2016D2       Adafruit_Trellis         Upload       Ctrl+U       Adafruit_TSL2591       tsl2591         Page Setup       Ctrl+Shift+P       Adafruit_VC0706       !         Preferences       Ctrl+Comma       Adafruit_WS2801       AdafruitMGC3130         Quit       Ctrl+Q       AttRGBMatrixPanel       AttRGBMatrixPanel       Attroutward         /*       CapSense       total       total       total       total	💿 tsl2591   Arduino 1.0.5	-	Adafruit_TCS34725	+	
New       Ctrl+ N       Adafruit_TFTLCD       Adafruit_TFTLCD         Open       Ctrl+ O       Adafruit_Thermal       Adafruit_TLC5947         Examples       Adafruit_TLC5911       Adafruit_TLC59711         Close       Ctrl+ W       Adafruit_TMP006       Adafruit_TMP007         Save       Ctrl+ S       Adafruit_TPA2016D2       Adafruit_TSL2591         Upload       Ctrl+ Shift+S       Adafruit_TSL2591       tsl2591         Vpload Using Programmer       Ctrl+ Shift+P       Adafruit_VS1053       Adafruit_VS1053         Page Setup       Ctrl+ Comma       Adafruit_MGC3130       adxl345         Quit       Ctrl+Q       AttRGBMatrixPanel       As3935         /*       Kackup       Kackup       Kackup       Kackup	File Edit Sketch Tools Help		Adafruit_TEA5767	•	
Open       Ctrl+O       Adafruit_Thermal       Adafruit_TLC5947         Examples       Adafruit_TLC5947       Adafruit_TLC5947         Examples       Adafruit_TLC5911       Adafruit_TLC5911         Close       Ctrl+W       Adafruit_TMP006         Save       Ctrl+S       Adafruit_TPA2016D2         Save As       Ctrl+Shift+S       Adafruit_Trellis         Upload       Ctrl+Shift+U       Adafruit_VC0706         Page Setup       Ctrl+Shift+P       Adafruit_VS1053         Print       Ctrl+Q       AdafruitMGC3130         Adafruit_TSL2591       tsl >1 = Adafruit_TSL23       AtlrGBMatrixPanel         Adafruit_TSL2591       tsl > + + + + + + + + + + + + + + + + + +	New	Ctrl+N	Adafruit_TFTLCD	•	
Sketchbook       Adafruit_TLC5947         Examples       Adafruit_TLC59711         Close       Ctrl+W         Save       Ctrl+S         Save As       Ctrl+Shift+S         Upload       Ctrl+U         Upload Using Programmer       Ctrl+Shift+U         Page Setup       Ctrl+Shift+P         Print       Ctrl+P         Preferences       Ctrl+Comma         Quit       Ctrl+Q         Adafruit_TSL2591       tsl2591         Adafruit_TSL2591       tsl2591         Adafruit_MGC3130       adxl345         Adafruit_TSL2591       AltrGBMatrixPanel         AS3935       backup         /*       CapSense	Open	Ctrl+O	Adafruit_Thermal	•	
Examples       Adafruit_TLC59711         Close       Ctrl+W         Save       Ctrl+S         Save As       Ctrl+Shift+S         Upload       Ctrl+U         Upload Using Programmer       Ctrl+Shift+P         Page Setup       Ctrl+Shift+P         Print       Ctrl+P         Preferences       Ctrl+Q         Adafruit_TSL2591       tsl2591         Adafruit_MGC3130       adxl345         Adafruit_TSL2591       tsl = Adafruit_TSL25         /*       Ctrl+Q	Sketchbook	•	Adafruit_TLC5947	• •	
Close       Ctrl+W       Adafruit_TMP006       >         Save       Ctrl+S       Adafruit_TMP007       >         Save As       Ctrl+Shift+S       Adafruit_TPA2016D2       >         Upload       Ctrl+U       Adafruit_Trellis       >         Upload Using Programmer       Ctrl+Shift+U       Adafruit_VC0706       >         Page Setup       Ctrl+P       Adafruit_VS1053       >         Preferences       Ctrl+Q       AdafruitMGC3130       >         Adafruit_TSL2591       tsl = Adafruit_TSL25       Afafruit_VPanel       >         Adafruit_Ctrl+Q       Atrogrammer       >       >         Adafruit_Ctrl+Q       Adafruit_MGC3130       >       >         Adafruit_TSL2591       tsl = Adafruit_TSL25       >       >         /*       CapSense       >       >	Examples	Þ	Adafruit_TLC59711	•	
Save       Ctrl+S       Adafruit_TMP007          Save As       Ctrl+Shift+S       Adafruit_TPA2016D2          Upload       Ctrl+U       Adafruit_Trellis          Upload Using Programmer       Ctrl+Shift+U       Adafruit_TSL2591       tsl2591         Page Setup       Ctrl+Shift+P       Adafruit_VC0706       !         Print       Ctrl+P       Adafruit_VS1053          Preferences       Ctrl+Comma       AdafruitMGC3130          Quit       Ctrl+Q       AltrGBMatrixPanel          Adafruit_TSL2591 ts1 = Adafruit_TSL25       backup       *******	Close	Ctrl+W	Adafruit_TMP006	•	
Save As       Ctrl+Shift+S       Adafruit_TPA2016D2       Adafruit_Trellis         Upload       Ctrl+U       Adafruit_Trellis       Itsl2591         Upload Using Programmer       Ctrl+Shift+U       Adafruit_TSL2591       Itsl2591         Page Setup       Ctrl+Shift+P       Adafruit_VS1053       Itsl2591         Print       Ctrl+P       Adafruit_WS2801       Adafruit_WS2801         Quit       Ctrl+Q       Adafruit_TSL2591       tsl2591         Adafruit_TSL2591       tsl1 = Adafruit_TSL25       AltRGBMatrixPanel       As3935         /*       CapSense       Itsl2591       Itsl2591	Save	Ctrl+S	Adafruit_TMP007	•	
Upload       Ctrl+U       Adafruit_Trellis       Image: Ctrl+Shift+U         Upload Using Programmer       Ctrl+Shift+U       Adafruit_TSL2591       tsl2591         Page Setup       Ctrl+Shift+P       Adafruit_VC0706       !         Print       Ctrl+P       Adafruit_WS2801       Image: Ctrl+Q         Adafruit_TSL2591       tsl2       Adafruit_MGC3130       Image: Ctrl+Q         Adafruit_TSL2591       tsl       Adafruit_TSL25       Image: Ctrl+Q         Adafruit_TSL2591       tsl       Adafruit_TSL25       Image: Ctrl+Q         Adafruit_TSL2591       tsl       = Adafruit_TSL25       Image: Ctrl+Q         Adafruit_TSL25       Image: Ctrl+Q       Image: Ctrl+Q       Image: Ctrl+Q         Adafruit_TSL25       Image: Ctrl+Q       Image: Ctrl+Q       Image: Ctrl+Q         Adafruit_TSL25       Image: Ctrl+Q	Save As	Ctrl+Shift+S	Adafruit_TPA2016D2	•	
Upload Using Programmer       Ctrl+Shift+U       Adafruit_TSL2591       tsl2591         Page Setup       Ctrl+Shift+P       Adafruit_VC0706       !         Print       Ctrl+P       Adafruit_VS1053       >         Preferences       Ctrl+Comma       Adafruit_MGC3130       >         Quit       Ctrl+Q       Adafruit_TSL2591 ts1 = Adafruit_TSL25       >         Adafruit_CS130       >       >       >         Adafruit_CS2591       ts1 = Adafruit_TSL25       >       >         /*       CapSense       >       >	Upload	Ctrl+U	Adafruit_Trellis	•	
Page Setup       Ctrl+ Shift+P         Print       Ctrl+P         Preferences       Ctrl+Comma         Quit       Ctrl+Q         Adafruit_VS1053       >         Adafruit_WS2801       >         Adafruit_WS2801       >         Adafruit_WS2801       >         Adafruit_MGC3130       >         adxl345       >         Adafruit_TSL2591       ts1 = Adafruit_TSL25         /*       CapSense	Upload Using Programmer	Ctrl+Shift+U	Adafruit_TSL2591	•	tsl2591
Page setup       Ctrl+Snint+P       Adafruit_VS1053       >         Print       Ctrl+P       Adafruit_WS2801       >         Preferences       Ctrl+Comma       AdafruitMGC3130       >         Quit       Ctrl+Q       Adafruit_MGC3130       >         Adafruit_TSL2591       ts1 = Adafruit_TSL25       AltRGBMatrixPanel       >         /*       CapSense       >       >	David Cature	Chill, Chiffe, D	Adafruit_VC0706	► 1.	
Preferences     Ctrl+Comma       Quit     Ctrl+Q       Adafruit_TSL2591 ts1 = Adafruit_TSL23     Adafruit_MGC3130       /*     Ctrl+Q       Adafruit_CTSL2591 ts1 = Adafruit_TSL23     As3935       /*     CapSense	Page Setup	Ctrl+Shift+P	Adafruit_VS1053	•	
Preferences     Ctrl+Comma     AdafruitMGC3130       Quit     Ctrl+Q       Adafruit_TSL2591 ts1 = Adafruit_TSL25     AltRGBMatrixPanel       /*     backup       /*     CapSense	Print	Ctri+P	Adafruit_WS2801	•	
Quit     Ctrl+Q     adxl345     AltRGBMatrixPanel       Adafruit_TSL2591 ts1 = Adafruit_TSL2     AltRGBMatrixPanel     AS3935       /**     backup     ************************************	Preferences	Ctrl+Comma	AdafruitMGC3130	•	
Quit     CuitQ       Adafruit_TSL2591 ts1 = Adafruit_TSL2     AltRGBMatrixPanel       /*     backup       /*     CapSense	Quit	Ctrl+0	adxl345	•	
AS3935	Adafruit TSL2591 tsl = Ad	afruit TSL29	AltRGBMatrixPanel	•	
/*************************************	Addituro_ISESSII (SI - Ad	diraro_1005	AS3935	•	
/* CapSense	/*****	**********	backup	• ****	*******/
	/*	I	CapSense	•	

Thats it! Now open up the serial terminal window at 9600 speed to begin the test.

Found a TSL2	591 sensor		
Sensor:	TSI.2591		
Driver Ver:	1		
Unique ID:	2591		
Max Value:	88000.00 lux		
Min Value:	0.00 lux		
Resolution:	1.00 lux		
Gain:	Medium (25x)		
<b>m</b> :	100 ms		
liming:	100 mb		
liming:			
[ 892 mg ] 4	86 00 1uw		
[ 892 ms ] 4	86.00 lux		
[ 892 ms ] 4 [ 1256 ms ] [ 1618 ms ]	86.00 lux 539.00 lux 676.00 lux		
[ 892 ms ] 4 [ 1256 ms ] 4 [ 1618 ms ] [ 1981 ms ]	200 mB 86.00 lux 539.00 lux 676.00 lux 859.00 lux		
[ 892 ms ] 4 [ 1256 ms ] [ 1618 ms ] [ 1981 ms ] [ 2344 ms ]	86.00 lux 539.00 lux 676.00 lux 859.00 lux 1359.00 lux		
[ 892 ms ] 4 [ 1256 ms ] [ 1618 ms ] [ 1981 ms ] [ 2344 ms ] [ 2707 ms ]	86.00 lux 539.00 lux 676.00 lux 859.00 lux 1359.00 lux 2132.00 lux		
[ 892 ms ] 4 [ 1256 ms ] [ 1618 ms ] [ 1981 ms ] [ 2344 ms ] [ 2707 ms ] [ 3069 ms ]	86.00 lux 539.00 lux 676.00 lux 859.00 lux 1359.00 lux 2132.00 lux 3400.00 lux		
[ 892 ms ] 4 [ 1256 ms ] [ 1618 ms ] [ 1981 ms ] [ 2344 ms ] [ 2707 ms ] [ 3069 ms ] [ 3433 ms ]	86.00 lux 539.00 lux 676.00 lux 859.00 lux 1359.00 lux 2132.00 lux 3400.00 lux 4862.00 lux		
[ 892 ms ] 4 [ 1256 ms ] [ 1618 ms ] [ 1981 ms ] [ 2344 ms ] [ 2707 ms ] [ 3069 ms ] [ 3433 ms ] [ 3795 ms ]	86.00 lux 539.00 lux 676.00 lux 859.00 lux 1359.00 lux 2132.00 lux 3400.00 lux 4862.00 lux 4081.00 lux		
[ 892 ms ] 4 [ 1256 ms ] [ 1618 ms ] [ 1981 ms ] [ 2344 ms ] [ 2707 ms ] [ 3069 ms ] [ 3433 ms ] [ 3795 ms ] [ 4158 ms ]	86.00 lux 539.00 lux 676.00 lux 859.00 lux 1359.00 lux 2132.00 lux 3400.00 lux 4862.00 lux 4081.00 lux 2658.00 lux		
[ 892 ms ] 4 [ 1256 ms ] [ 1618 ms ] [ 1981 ms ] [ 2344 ms ] [ 2707 ms ] [ 3069 ms ] [ 3433 ms ] [ 3795 ms ] [ 4158 ms ] [ 4521 ms ]	86.00 lux 539.00 lux 676.00 lux 859.00 lux 1359.00 lux 2132.00 lux 3400.00 lux 4862.00 lux 4081.00 lux 2658.00 lux 1639.00 lux		

Try covering with your hand or shining a lamp onto the sensor to experiment with the light levels!

## Library Reference

The Adafruit\_TSL2591 library contains a number of public functions to help you get started with this sensor.

### Constructor

To create an instance of the Adafruit\_TSL2591 driver, simple declare an appropriate object, along with a 32-bit numeric value to identify this sensor (in case you have several TSL2591s and want to track them separately in a logging system).

Adafruit\_TSL2591 tsl = Adafruit\_TSL2591(2591);

### Gain and Timing

You can adjust the gain settings and integration time of the sensor to make it more or less sensitive to light, depending on the environment where the sensor is being used.

The gain can be set to one of the following values (though the last value, MAX, has limited use in the real world given the extreme amount of gain applied):

- TSL2591\_GAIN\_LOW: Sets the gain to 1x (bright light)
- TSL2591\_GAIN\_MEDIUM: Sets the gain to 25x (general purpose)
- TSL2591\_GAIN\_HIGH: Sets the gain to 428x (low light)
- TSL2591\_GAIN\_MAX: Sets the gain to 9876x (extremely low light)

Gain can be read or set via the following functions:

- void setGain(tsl2591Gain\_t gain);
- tsl2591Gain\_t getGain();

The integration time can be set between 100 and 600ms, and the longer the integration time the more light the sensor is able to integrate, making it more sensitive in low light the longer the integration time. The following values can be used:

- TSL2591\_INTEGRATIONTIME\_100MS
- TSL2591\_INTEGRATIONTIME\_200MS
- TSL2591\_INTEGRATIONTIME\_300MS
- TSL2591\_INTEGRATIONTIME\_400MS
- TSL2591\_INTEGRATIONTIME\_500MS
- TSL2591\_INTEGRATIONTIME\_600MS

The integration time can be read or set via the following functions:

- void setTiming (tsl2591IntegrationTime\_t integration);
- tsl2591IntegrationTime\_t getTiming();

An example showing how these functions are used can be seen in the code below:

```
/*
   Configures the gain and integration time for the TSL2561
*/
void configureSensor(void)
{
 // You can change the gain on the fly, to adapt to brighter/dimmer light situations
 //tsl.setGain(TSL2591_GAIN_LOW); // 1x gain (bright light)
 tsl.setGain(TSL2591 GAIN MED);
                                // 25x gain
 //tsl.setGain(TSL2591 GAIN HIGH); // 428x gain
 // Changing the integration time gives you a longer time over which to sense light
 // longer timelines are slower, but are good in very low light situtations!
 tsl.setTiming(TSL2591 INTEGRATIONTIME 100MS); // shortest integration time (bright light)
 //tsl.setTiming(TSL2591 INTEGRATIONTIME 200MS);
 //tsl.setTiming(TSL2591 INTEGRATIONTIME 300MS);
 //tsl.setTiming(TSL2591 INTEGRATIONTIME 400MS);
 //tsl.setTiming(TSL2591 INTEGRATIONTIME 500MS);
 //tsl.setTiming(TSL2591 INTEGRATIONTIME 600MS); // longest integration time (dim light)
 /* Display the gain and integration time for reference sake */
 Serial.println("-----");
 Serial.print ("Gain:
                      ");
 tsl2591Gain t gain = tsl.getGain();
 switch(gain)
 {
   case TSL2591_GAIN_LOW:
     Serial.println("1x (Low)");
     break;
   case TSL2591 GAIN MED:
     Serial.println("25x (Medium)");
     break;
   case TSL2591 GAIN HIGH:
     Serial.println("428x (High)");
     break:
   case TSL2591 GAIN MAX:
     Serial.println("9876x (Max)");
     break;
 }
 Serial.print ("Timing:
                        ");
 Serial.print((tsl.getTiming() + 1) * 100, DEC);
 Serial.println(" ms");
 Serial.println("-----");
 Serial.println("");
}
```

### Unified Sensor API

The Adafruit\_TSL2591 library makes use of the Adafruit unified sensor framework (https://adafru.it/dGB) to provide sensor data in a standardized format and scale. If you wish to make use of this framweork, the two key functions that you need to work with are **getEvent** and **getSensor**, as described below:

### void getEvent(sensors\_event\_t\*)

This function will read a single sample from the sensor and return it in a generic sensors\_event\_t object. To use this function, you simply pass in a sensors\_event\_t reference, which will be populated by the function, and then read the

```
/*
   Performs a read using the Adafruit Unified Sensor API.
*/
void unifiedSensorAPIRead(void)
{
 /* Get a new sensor event */
 sensors event t event;
 tsl.getEvent(&event);
 /* Display the results (light is measured in lux) */
 Serial.print("[ "); Serial.print(event.timestamp); Serial.print(" ms ] ");
 if ((event.light == 0) |
     (event.light > 4294966000.0) |
     (event.light <-4294966000.0))
 {
   /* If event.light = 0 lux the sensor is probably saturated */
   /* and no reliable data could be generated! */
   /* if event.light is +/- 4294967040 there was a float over/underflow */
   Serial.println("Invalid data (adjust gain or timing)");
 }
 else
 {
   Serial.print(event.light); Serial.println(" lux");
 }
}
```

Note that some checks need to be performed on the sensor data in case the sensor saturated. If saturation happens, please adjust the gain and integration time up or down to change the sensor's sensitivity and output range.

### void getSensor(sensor\_t\*)

This function returns some basic information about the sensor, and operates in a similar fashion to getEvent. You pass in an empty sensor\_t reference, which will be populated by this function, and we can then read the results and retrieve some key details about the sensor and driver, as shown in the code below:

```
/*
   Displays some basic information on this sensor from the unified
   sensor API sensor t type (see Adafruit Sensor for more information)
*/
void displaySensorDetails(void)
{
 sensor_t sensor;
 tsl.getSensor(&sensor);
 Serial.println("-----");
 Serial.print ("Sensor: "); Serial.println(sensor.name);
 Serial.print ("Driver Ver: "); Serial.println(sensor.version);
 Serial.print ("Unique ID: "); Serial.println(sensor.sensor id);
 Serial.print ("Max Value: "); Serial.print(sensor.max value); Serial.println(" lux");
 Serial.print ("Min Value: "); Serial.print(sensor.min value); Serial.println(" lux");
 Serial.print ("Resolution: "); Serial.print(sensor.resolution); Serial.println(" lux");
 Serial.println("-----");
 Serial.println("");
 delay(500);
}
```

#### Raw Data Access API

If you don't wish to use the Unified Sensor API, you can access the raw data for this sensor via the following three functions:

- uint16\_t getLuminosity (uint8\_t channel );
- uint32\_t getFullLuminosity ();
- uint32\_t calculateLux ( uint16\_t ch0, uint16\_t ch1 );

getLuminosity can be used to read either the visible spectrum light sensor, or the infrared light sensor. It will return the raw 16-bit sensor value for the specified channel, as shown in the code below:

```
/*
   Shows how to perform a basic read on visible, full spectrum or
   infrared light (returns raw 16-bit ADC values)
*/
void simpleRead(void)
{
 // Simple data read example. Just read the infrared, fullspecrtrum diode
 // or 'visible' (difference between the two) channels.
 // This can take 100-600 milliseconds! Uncomment whichever of the following you want to read
 uint16 t x = tsl.getLuminosity(TSL2591 VISIBLE);
 //uint16 t x = tsl.getLuminosity(TSL2561 FULLSPECTRUM);
 //uint16 t x = tsl.getLuminosity(TSL2561 INFRARED);
 Serial.print("[ "); Serial.print(millis()); Serial.print(" ms ] ");
 Serial.print("Luminosity: ");
 Serial.println(x, DEC);
}
```

getFullLuminosity reads both the IR and full spectrum sensors at the same time to allow tigher correlation between the values, and then separates them in SW. The function returns a 32-bit value which needs to be split into two 16-bit

```
/*
   Show how to read IR and Full Spectrum at once and convert to lux
*/
void advancedRead(void)
{
 // More advanced data read example. Read 32 bits with top 16 bits IR, bottom 16 bits full spectrum
 // That way you can do whatever math and comparisons you want!
 uint32 t lum = tsl.getFullLuminosity();
 uint16 t ir, full;
 ir = lum >> 16;
 full = lum & 0xFFFF;
 Serial.print("[ "); Serial.print(millis()); Serial.print(" ms ] ");
 Serial.print("IR: "); Serial.print(ir); Serial.print(" ");
 Serial.print("Full: "); Serial.print(full); Serial.print(" ");
 Serial.print("Visible: "); Serial.print(full - ir); Serial.print(" ");
 Serial.print("Lux: "); Serial.println(tsl.calculateLux(full, ir));
}
```

calculateLux can be used to take both the infrared and visible spectrum sensor data and roughly correlate with the equivalent SI lux value, based on a formula from the silicon vendor that takes into account the sensor properties and the integration time and gain settings of the device.

To calculate the lux, simple call **calculateLux(full, ir)**, where 'full' and 'ir' are raw 16-bit values taken from one of the two raw data functions above. See the code sample above for an example of calculating lux.

# Arduino Library Docs

Arduino Library Docs (https://adafru.it/AvJ)





## Python & CircuitPython

It's easy to use the TSL2591 sensor with Python and CircuitPython and the Adafruit CircuitPython TSL2591 (https://adafru.it/C4I) 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 (https://adafru.it/BSN).

### CircuitPython Microcontroller Wiring

First wire up a TSL2591 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 (https://adafru.it/BSN).

Here's the Raspberry Pi wired with I2C:



## CircuitPython Installation of TSL2591

Next you'll need to install the Adafruit CircuitPython TSL2591 (https://adafru.it/C4I) library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython (https://adafru.it/Amd) 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 (https://adafru.it/zdx). Our introduction guide has a great page on how to install the library bundle (https://adafru.it/ABU) for both express and non-express boards.

Remember for non-express boards like the, you'll need to manually install the necessary libraries from the bundle:

- adafruit\_tsl2591.mpy
- adafruit\_bus\_device

Before continuing make sure your board's lib folder or root filesystem has the **adafruit\_tsl2591.mpy**, and **adafruit\_bus\_device** files and folders copied over.

Next connect to the board's serial REPL (https://adafru.it/Awz)so you are at the CircuitPython >>> prompt.

## 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_tsl2591
i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_tsl2591.TSL2591(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.
- **visible** The visible light level measured by the sensor. This is a 32-bit unsigned value with no units, the higher the number the more visible light.
- **infrared** The infrared light level measured by the sensor. This is a 16-bit unsigned value with no units, the higher the number the more infrared light.
- **full\_spectrum** The visible & infrared light level measured by the sensor as a single 32-bit value with no units. The higher the number the more full spectrum light.
- raw\_luminosity A 2-tuple of raw sensor visible+IR and IR only light levels. Each is a 16-bit value with no units where the higher the value the more light.

```
print('Light: {0}lux'.format(sensor.lux))
print('Visible: {0}'.format(sensor.visible))
print('Infrared: {0}'.format(sensor.infrared))
```

```
>>> print('Light: {0}lux'.format(sensor.lux))
Light: 272.146lux
>>> print('Visible: {0}'.format(sensor.visible))
Visible: 24315761
>>> print('Infrared: {0}'.format(sensor.infrared))
Infrared: 370
>>>
```

In addition you can read and write a few properties to change how the sensor behaves and is configured:

- **gain** Set this to a value below to change the gain of the light sensor:
  - adafruit\_tsl2591.GAIN\_LOW 1x gain
  - adafruit\_tsl2591.GAIN\_MED 25x gain (the default)
  - adafruit\_tsl2591.GAIN\_HIGH 428x gain

- adafruit\_tsl2591.GAIN\_MAX 9876x gain
- **integration\_time** Set the integration time of the sensor to a value of:
  - adafruit\_tsl2591.INTEGRATIONTIME\_100MS 100ms (the default)
  - adafruit\_tsl2591.INTEGRATIONTIME\_200MS 200ms
  - adafruit\_tsl2591.INTEGRATIONTIME\_300MS 300ms
  - adafruit\_tsl2591.INTEGRATIONTIME\_400MS 400ms
  - adafruit\_tsl2591.INTEGRATIONTIME\_500MS 500ms
  - adafruit\_tsl2591.INTEGRATIONTIME\_600MS 600ms

```
sensor.gain = adafruit_tsl2591.GAIN_LOW
sensor.integration time = adafruit tsl2591.INTEGRATIONTIME 200MS
```

```
>>> sensor.gain = adafruit_tsl2591.GAIN_LOW
>>> sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_200MS
>>> print('Light: {0}lux'.format(sensor.lux))
Light: 281.112lux
>>> print('Visible: {0}'.format(sensor.visible))
Visible: 1966237
>>> print('Infrared: {0}'.format(sensor.infrared))
Infrared: 30
>>>
```

That's all there is to using the TSL2591 sensor with CircuitPython!

Below is a complete example that will print the light level every second. Save this as **code.py** on the board and view the output in the REPL.

### Full Example Code

```
# Simple demo of the TSL2591 sensor. Will print the detected light value
# every second.
import time
import board
import busio
import adafruit tsl2591
# Initialize the I2C bus.
i2c = busio.I2C(board.SCL, board.SDA)
# Initialize the sensor.
sensor = adafruit tsl2591.TSL2591(i2c)
# You can optionally change the gain and integration time:
#sensor.gain = adafruit tsl2591.GAIN LOW (1x gain)
#sensor.gain = adafruit tsl2591.GAIN MED (25x gain, the default)
#sensor.gain = adafruit tsl2591.GAIN HIGH (428x gain)
#sensor.gain = adafruit tsl2591.GAIN MAX (9876x gain)
#sensor.integration time = adafruit tsl2591.INTEGRATIONTIME 100MS (100ms, default)
#sensor.integration time = adafruit tsl2591.INTEGRATIONTIME 200MS (200ms)
#sensor.integration time = adafruit tsl2591.INTEGRATIONTIME 300MS (300ms)
#sensor.integration time = adafruit tsl2591.INTEGRATIONTIME 400MS (400ms)
#sensor.integration time = adafruit tsl2591.INTEGRATIONTIME 500MS (500ms)
#sensor.integration time = adafruit tsl2591.INTEGRATIONTIME 600MS (600ms)
# Read the total lux, IR, and visible light levels and print it every second.
while True:
   # Read and calculate the light level in lux.
   lux = sensor.lux
   print('Total light: {0}lux'.format(lux))
   # You can also read the raw infrared and visible light levels.
   # These are unsigned, the higher the number the more light of that type.
   # There are no units like lux.
   # Infrared levels range from 0-65535 (16-bit)
    infrared = sensor.infrared
    print('Infrared light: {0}'.format(infrared))
    # Visible-only levels range from 0-2147483647 (32-bit)
   visible = sensor.visible
    print('Visible light: {0}'.format(visible))
    # Full spectrum (visible + IR) also range from 0-2147483647 (32-bit)
    full spectrum = sensor.full spectrum
    print('Full spectrum (IR + visible) light: {0}'.format(full_spectrum))
    time.sleep(1.0)
```

# Python Docs

Python Docs (https://adafru.it/C4n)





## Downloads

### Datasheets & Files

- EagleCAD PCB files on GitHub (https://adafru.it/pOf)
- Fritzing object in Adafruit Fritzing library (https://adafru.it/c7M)



https://adafru.it/Foz

## Schematic



## Layout

(Dimensions are in Inches)





#### ООО "ЛайфЭлектроникс"

ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 3010181090000000703 БИК 044030703

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- Конкурентоспособные цены и скидки постоянным клиентам.
- Специальные условия для постоянных клиентов.
- Подбор аналогов.
- Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.
- Приемлемые сроки поставки, возможна ускоренная поставка.
- Доставку товара в любую точку России и стран СНГ.
- Комплексную поставку.
- Работу по проектам и поставку образцов.
- Формирование склада под заказчика.
- Сертификаты соответствия на поставляемую продукцию (по желанию клиента).
- Тестирование поставляемой продукции.
- Поставку компонентов, требующих военную и космическую приемку.
- Входной контроль качества.
- Наличие сертификата ISO.

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- Регистрацию проекта у производителя компонентов.
- Техническую поддержку проекта.
- Защиту от снятия компонента с производства.
- Оценку стоимости проекта по компонентам.
- Изготовление тестовой платы монтаж и пусконаладочные работы.



Тел: +7 (812) 336 43 04 (многоканальный) Email: org@lifeelectronics.ru

#### www.lifeelectronics.ru