Light Sensor - OPT3002 - Trēo™ Module

Module Features

- Texas Instruments OPT3002
- RoHS Compliant
- Software Library
- NightShade Trēo™ Compatible
- Breakout Headers

OPT3002 Features

(from Texas Instruments)

- Wide Optical Spectrum: 300 nm to 1000 nm
- Automatic Full-Scale Setting Feature Simplifies Software and Configuration
- Measurement Levels: 1.2 nW/cm2 to 10 mW/cm2
- 23-Bit Effective Dynamic Range with Automatic Gain Ranging
- 12 Binary-Weighted, Full-Scale Range Settings: < 0.2% (typ) Matching Between Ranges

Applications

- Intrusion and Door-Open Detection Systems
- Medical and Scientific Instrumentation
- Display Backlight Controls
- Lighting Control Systems
- Thermostats and Home Automation

Trēo™ Compatibility

Electrical

Communication	I2C
Max Current, 3.3V	1mA
Max Current, 5V	0mA

Mechanical

- 25mm x 25mm Outline
- 20mm x 20mm Hole Pattern
- M2.5 Mounting Holes

Description

The OPT3002 Trēo™ Module is a Light Sensor module that that features Texas Instruments' OPT3002 Light Sensor. It responds to a wide spectral range of 300-1000nm and takes measurements over a 23-bit effective range. This module can also generate interrupts based on the luminous intensity. This module is a part of the NightShade Treo system, patent pending.

Table of Contents

1 Summary 2
2 What is Trēo™? 2
3 Electrical Characteristics 3
4 Electrical Schematic 4
5 Mechanical Outline 5
6 Example Arduino Program 6
7 Library Overview (C++ & Python) 7

1 Summary

The OPT3002 is initialized with the begin() method and the results are retrieved with the readLightLevel() method. Other measurement parameters can be changed with the remaining methods.

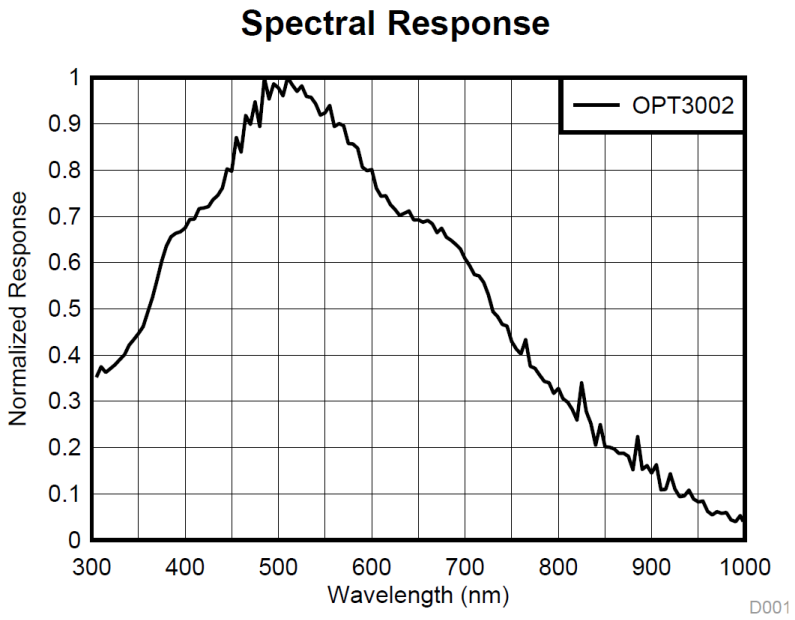


Figure 1. OPT3002 Spectral Response from TI OPT3002 Datasheet SBOS745A

2 What is Trēo™?

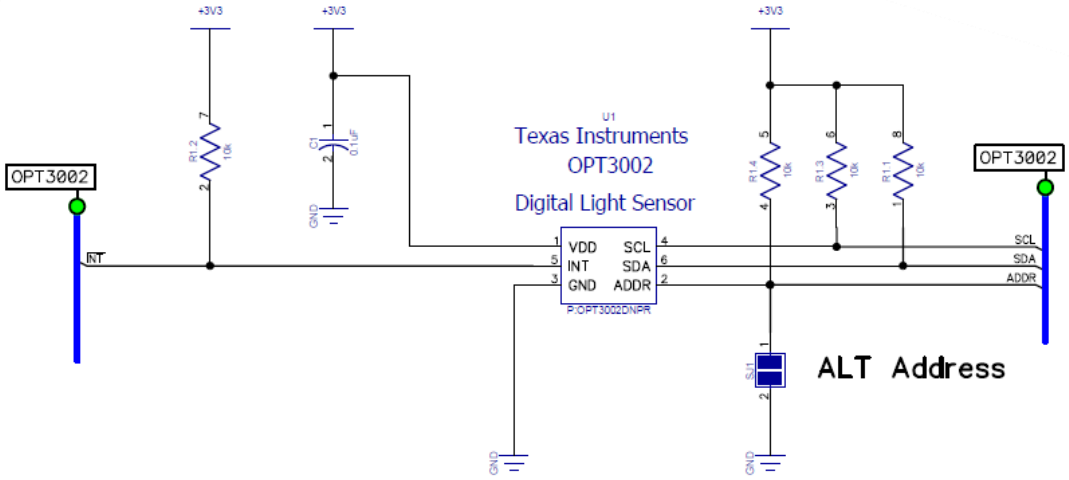
NightShade Trēo is a system of electronic modules that have standardized mechanical, electrical, and software interfaces. It provides you with a way to quickly develop electronic systems around microprocessor development boards. The grid attachment system, common connector/cabling, and extensive cross-platform software library allow you more time to focus on your application. Trēo is supported with detailed documentation and CAD models for each device.

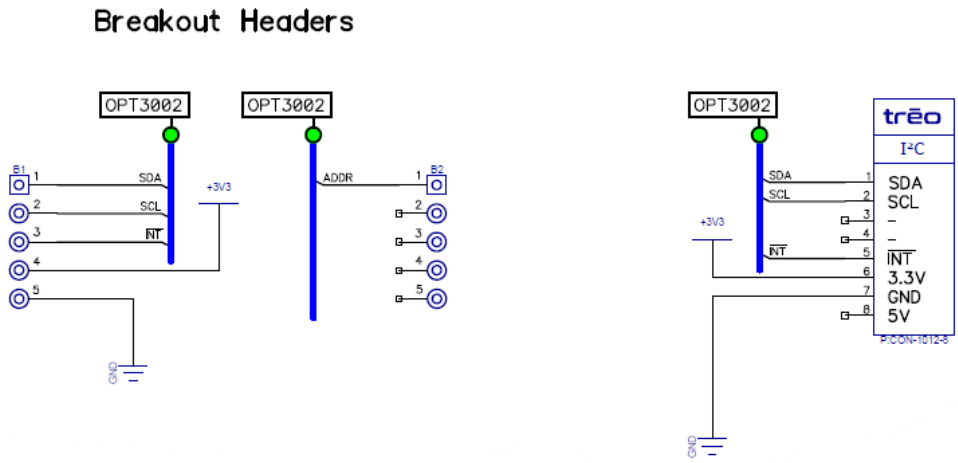
Learn more about Trēo here.

3 Electrical Characteristics

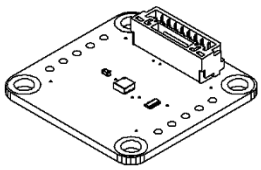
	Minimum	Nominal	Maximum
Voltages			
$V_{i/o}$ (SDA, SCL, INT)	-0.3V	-	3.6V
$V_{3.3V}$	3.1V	3.3V	3.5V
Measurement			
Peak Irradiance Spectral Responsivity	-	505nm	-
Sampling Rate	10Hz	-	1.25Hz
Range	1.2nW/cm^2	-	10063872mW/cm^2
Precision	1.2nW/cm^2	-	2457.6nW/cm^2
Error	2% (>5000nW/cm^2)	-	5% (<5000nW/cm^2)
I2C Slave Address			
SJ1 Open (Default)		0x45	
SJ1 Closed (Soldered)		0x44	
Operating Temperature	-25°C	-	+85°C

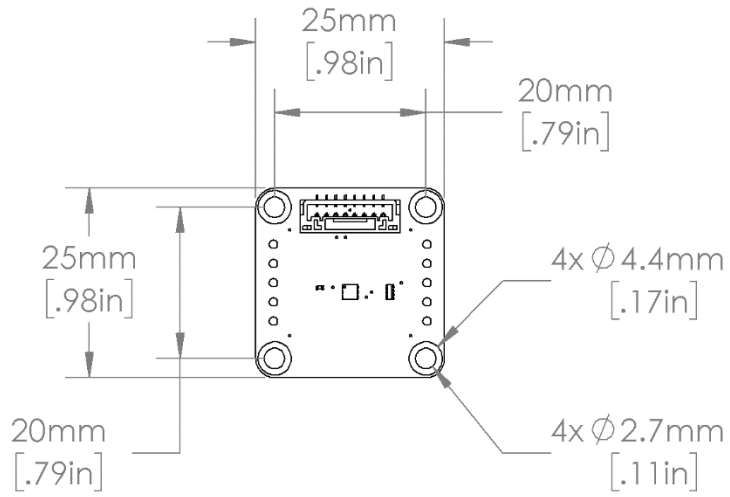
4 Electrical Schematic

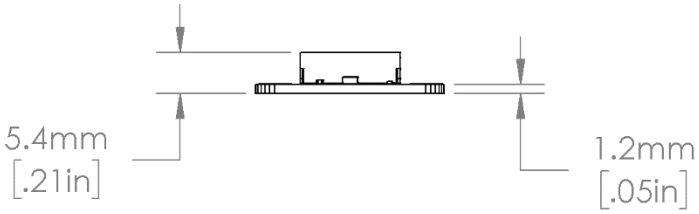




5 Mechanical Outline







6 Example Arduino Program

```
/*********************************************************
  OPT3002_LightSensor - NightShade_Treo by NightShade Electronics

  This sketch demonstrates the functionality of the
  NightShade Trēo OPT3002 light sensing module.
  (NSE-1131-1) It prints the measured light level to
  Serial at 115200 baudrate.

  Created by Aaron D. Liebold
  on February 15, 2021

  Links:
  NightShade Trēo System: https://nightshade.net/treo
  Product Page: https://nightshade.net/product/treo-light-sensor-opt3002/

  Distributed under the MIT license
  Copyright (C) 2021 NightShade Electronics
  https://opensource.org/licenses/MIT
*********************************************************/

// Include NightShade Treo Library
#include <NightShade_Treo.h>

// Declare Objects
NightShade_Treo_OPT3002 sensor(1);

void setup() {
  sensor.begin();
  Serial.begin(115200);
}

void loop() {
  float lightValue = sensor.readLightLevel();
  Serial.print("Light Level = ");
  Serial.print(lightValue, 1);
  Serial.println("nW/cm2");
  delay(500);
}
```

7 Library Overview (C++ & Python)

C++ Class

NightShade_Treo_OPT3002 <classObject>();

Python Module

<classObject> = NightShade_Treo.OPT3002()

7.1 Constructors

NightShade_Treo_OPT3002(int port, uint8_t slaveAddress, uint32_t clockSpeed)

Creates a OPT3002 object.

Arguments:

port	Integer of the I2C port used (e.g. 0 = "/dev/i2c_0")
slaveAddress	7-bit slave address
clockSpeed	Desired clock speed for the bus

Returns:

Nothing

NightShade_Treo_OPT3002(int port)

Creates a OPT3002 object assuming the default slave address and clock speed.

Arguments:

port	Integer of the I2C port used. (e.g. 0 = "/dev/i2c_0")

Returns:

Nothing

7.2 Methods

writeConfigReg(uint16_t regSetting)

Writes a 16-bit value to the configuration register.

Arguments:

register value

Returns:

Error	0 = Success

readConfigReg()

Reads the 16-bit configuration register value.

Arguments:
None

Returns:
register value (uint16_t)

readLightLevel()

Returns the current light level measurement in nW/cm^2 regardless of the current FSR setting.

Arguments:
None

Returns:
light level in nW/cm^2 (float)

readLightLevelRaw()

Returns the raw 16-bit result register value. The light level value is equal to $\frac{1.2nW}{cm^2} x\ Mantissa\ x\ 2^{Exponent}$.

Arguments:
None

Returns:

Result Register	B15-B12: Exponent B11-B0: Mantissa in 1.2nW/cm^2 per LSB

setHighLimit(uint16_t setting)

Sets the interrupt's high-limit value in raw format. The light level value is equal to $\frac{1.2nW}{cm^2} x\ Mantissa\ x\ 2^{Exponent}$.

Arguments:

setting	B15-B12: Exponent B11-B0: Mantissa in 1.2nW/cm^2 per LSB

Returns:

Error	0 = Success

setLowLimit(uint16_t setting)

Sets the interrupt's low-limit value in raw format. The light level value is equal to $\frac{1.2nW}{cm^2} x\ Mantissa\ x\ 2^{Exponent}$.

Arguments:

setting	B15-B12: Exponent B11-B0: Mantissa in 1.2nW/cm^2 per LSB

Returns:

Error	0 = Success

readMfgId()

Reads the manufacture ID number from the OPT3002, which is 0x5449.

Arguments:

None

Returns:

Manufacture ID (uint16_t)

setFullScaleRange(int setting)

Sets the full-scale measurement range.

Arguments:

setting	0:	FSR = 4,914nW/cm^2	LSB = 1.2nW/cm^2
	1:	FSR = 9,828nW/cm^2	LSB = 2.4nW/cm^2
	2:	FSR = 19,656nW/cm^2	LSB = 4.8nW/cm^2
	3:	FSR = 39,312nW/cm^2	LSB = 9.6nW/cm^2
	4:	FSR = 78,624nW/cm^2	LSB = 19.2nW/cm^2
	5:	FSR = 157,248nW/cm^2	LSB = 38.4nW/cm^2
	6:	FSR = 314,496nW/cm^2	LSB = 76.8nW/cm^2
	7:	FSR = 628,992nW/cm^2	LSB = 153.6nW/cm^2
	8:	FSR = 1,257,984nW/cm^2	LSB = 307.2nW/cm^2
	9:	FSR = 2,515,968nW/cm^2	LSB = 614.4nW/cm^2
	10:	FSR = 5,031,936nW/cm^2	LSB = 1,228.8nW/cm^2
	11:	FSR = 10,063,872nW/cm^2	LSB = 2,457.6nW/cm^2

Returns:

Error	0 = Success

enableLongSampleTime(int enable)

The sampling time of the OPT3002 can be set to 100ms (10Hz) or 800ms (1.25Hz). The larger sampling time decreases the sample rate, but it increases the resolution and accuracy of the measurement.

Arguments:

enable	true/false

Returns:

Error	0 = Success

setPowerMode(int setting)

Sets the operating mode of the OPT3002.

Arguments:

setting	0: Shutdown 1: Single-shot 2: Continuous conversions

Returns:

Error	0 = Success

readOverflowFlag()

The overflow flag is set if the converted light value overflows the register. This happens when the light value it greater that the current FSR of the sensor.

Arguments:

None

Returns:

Overflow flag (true/false)

readHighValueFlag()

The high-value flag is set when the result exceeds the high-limit.

Arguments:

None

Returns:

High-value flag (true/false)

readLowValueFlag()

The low-value flag is set when the result is less that the low-limit.

Arguments:

None

Returns:

Low-value flag (true/false)

setIntLatch(int enableLatching)

When enabled, the interrupt state is latched until the user clears the event.

Arguments:

enableLatching	true/false

Returns:

Error	0 = Success

setIntPolarity(int activeHigh)

Sets the polarity of the interrupt output pin. The interrupt must be set Active-Low for use with the Trēo system.

Arguments:

activeHigh	0: Interrupt is active-low 1: Interrupt is active-high

Returns:

Error	0 = Success

enableExponentMask(int enableMask)

When enabled, the exponent field of the raw data will not be given. Only the mantissa of the set FSR will be reported.

Arguments:

enableMask	true/false

Returns:

Error	0 = Success