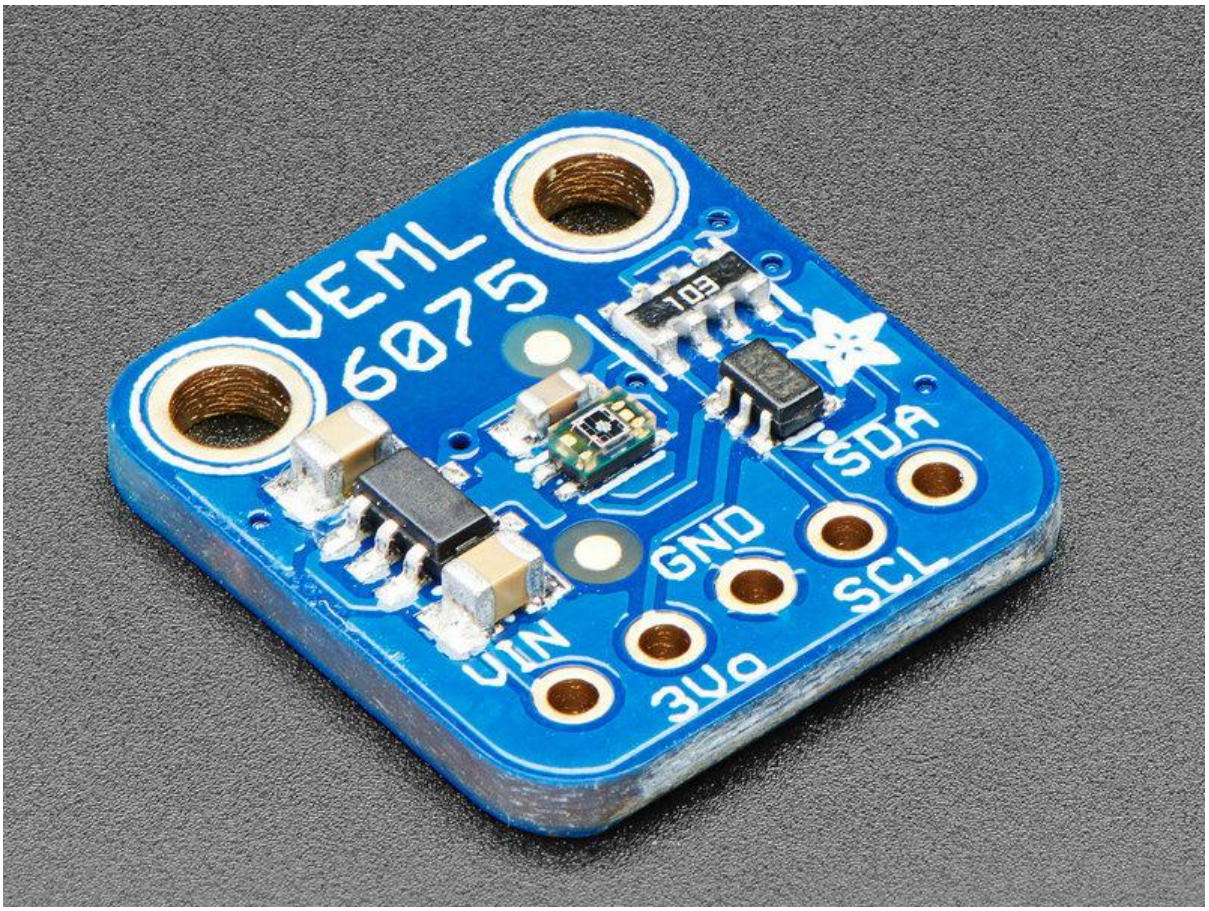
Adafruit VEML6075 UVA / UVB / UV Index Sensor

Created by lady ada

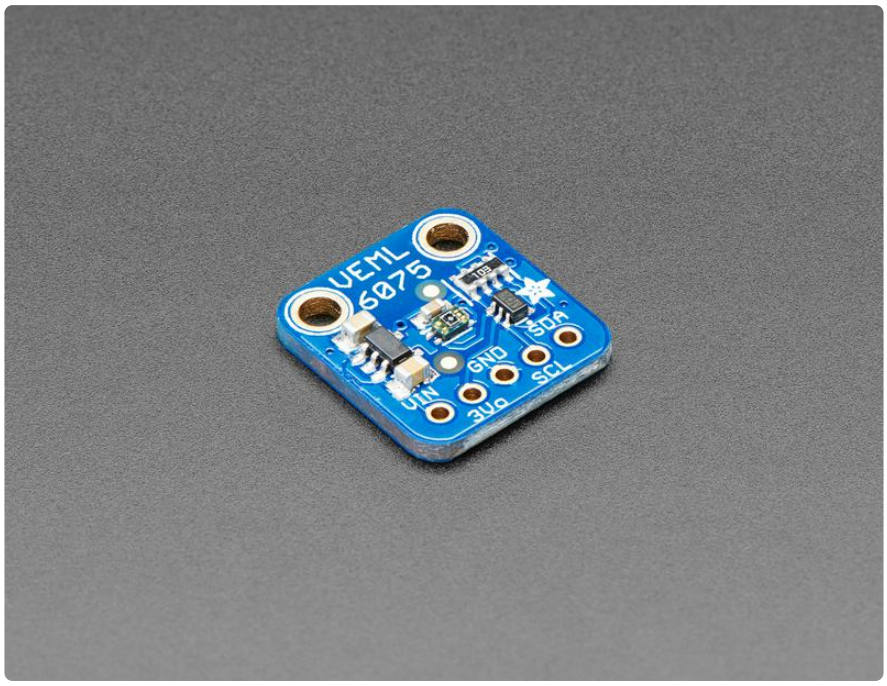


https://learn.adafruit.com/adafruit-veml6075-uva-uvb-uv-index-sensor

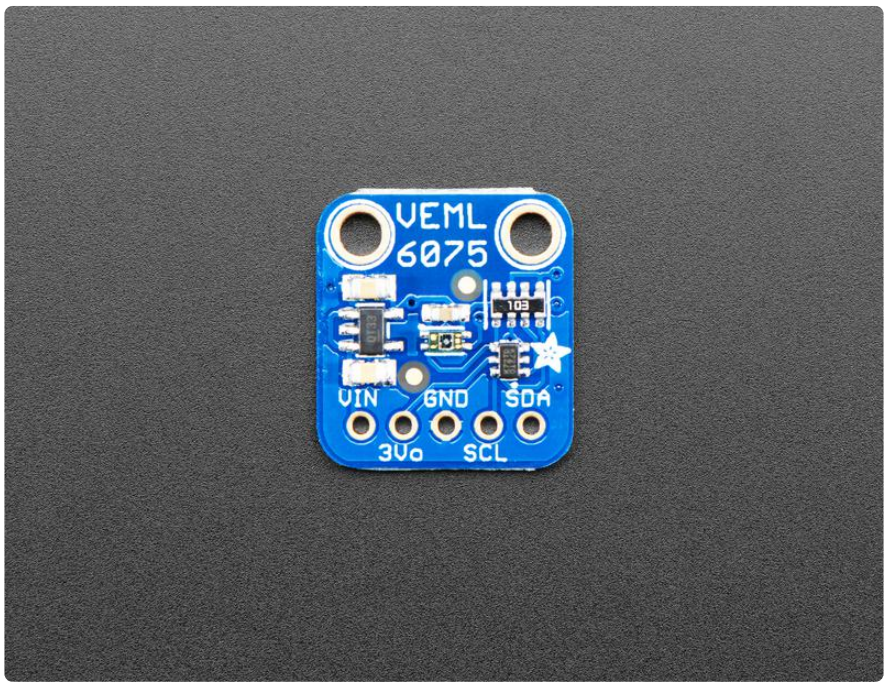
Last updated on 2022-12-01 03:21:40 PM EST

Table of Contents

Overview

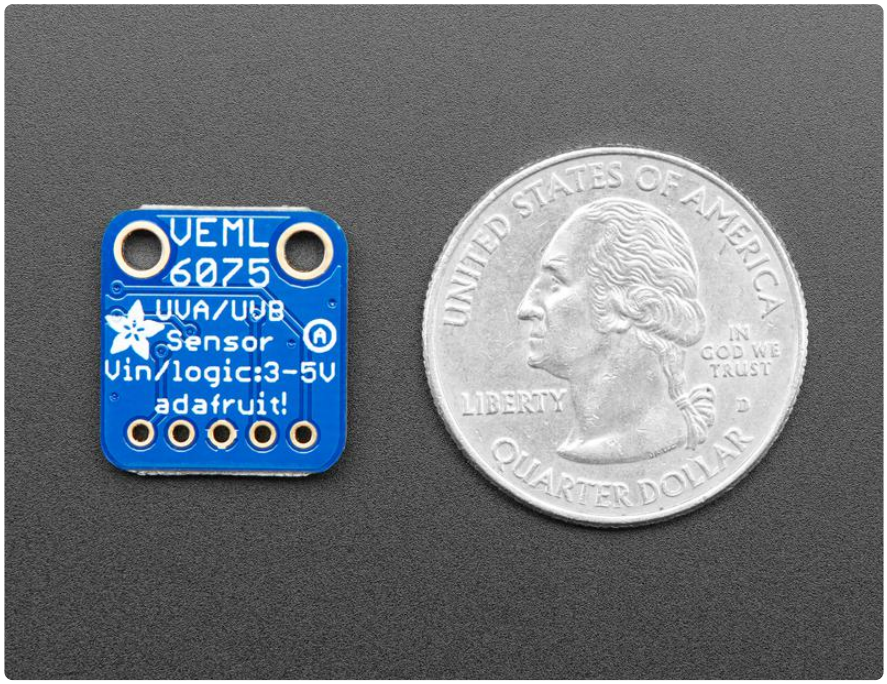


This little sensor is a great way to add UVA and UVB light sensing to any microcontroller project. The VEML6070 from Vishay has both true UVA and UVB band light sensors and an I2C-controlled ADC that will take readings and integrate them. The sensor also comes with calibration registers so you can easily convert the UVA/UVB readings into the UV Index.

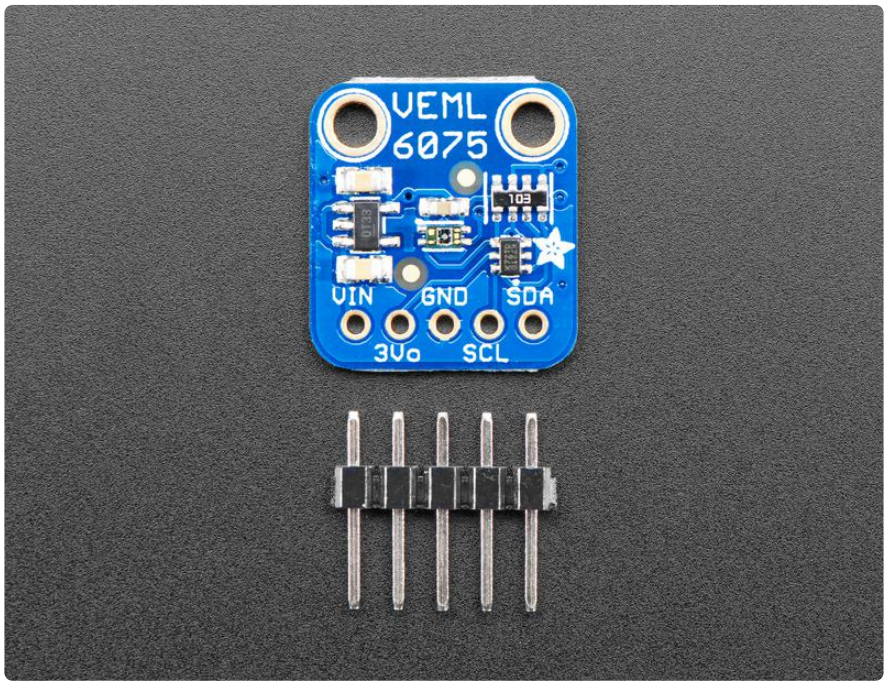


Compared to our other UV sensors, this one actually does a pretty good job of getting accurate UV data. Unlike the Si1145, it has a real UV sensor, and in contrast to the

VEML6070, it has dual band senors and an Index calculation algorithm. So far this is the best UV sensor we've got!



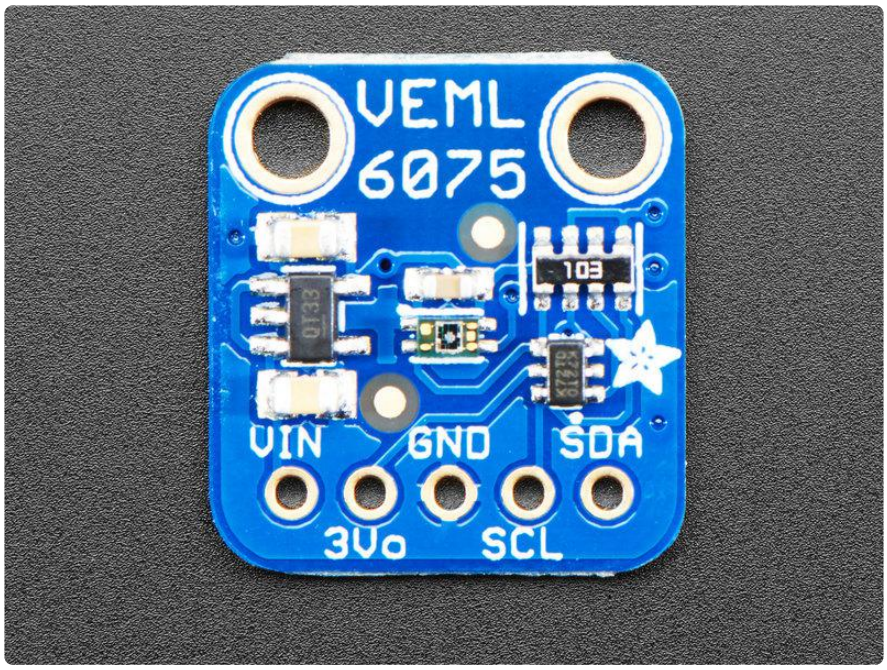
This UV sensor works great with 3 or 5V power or logic, its nice and compact, and its easy to use with any I2C-capable microcontroller. We have example code and libraries for Arduino () and CircuitPython/Python ().



Each order comes with one assembled PCB with a sensor, power regulator, level shifting and a small piece of header. Some light soldering is required to attach the header but its a fast task!

Pinouts

The VEML607 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 0x10 and you can't change it!



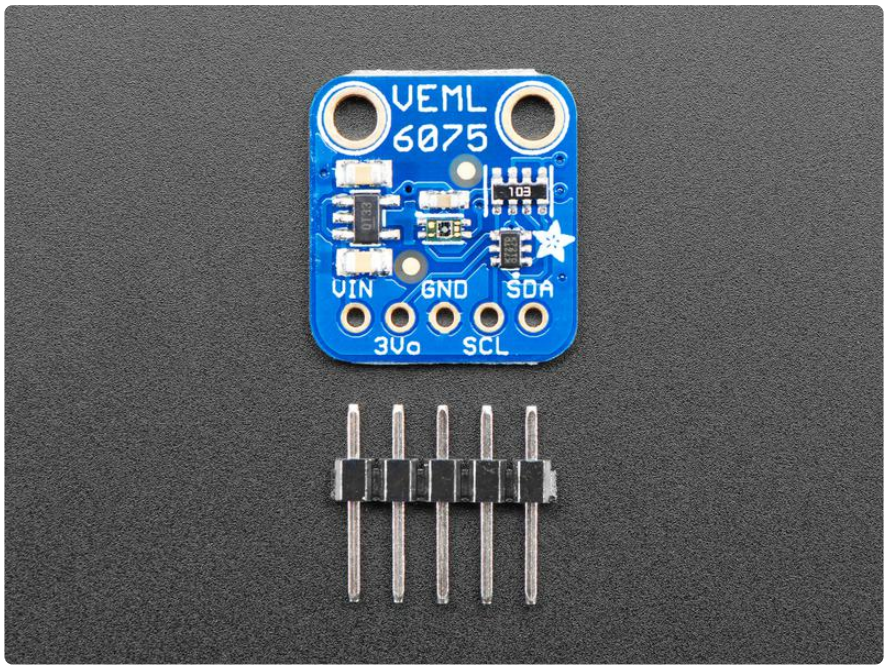
Power Pins:

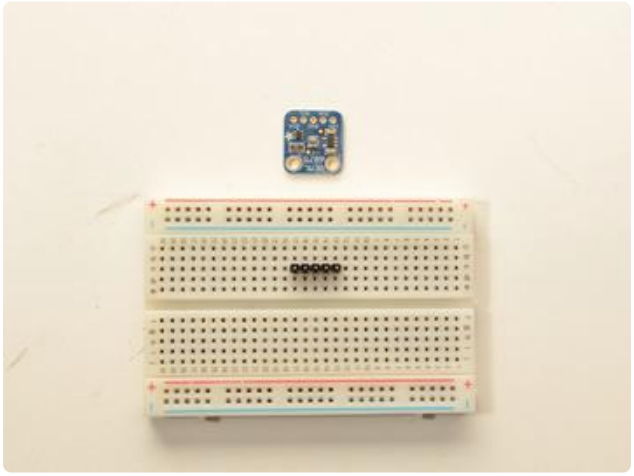
- Vin - this is the power pin. Since the chip uses 3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V
- 3v3 - this is the 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

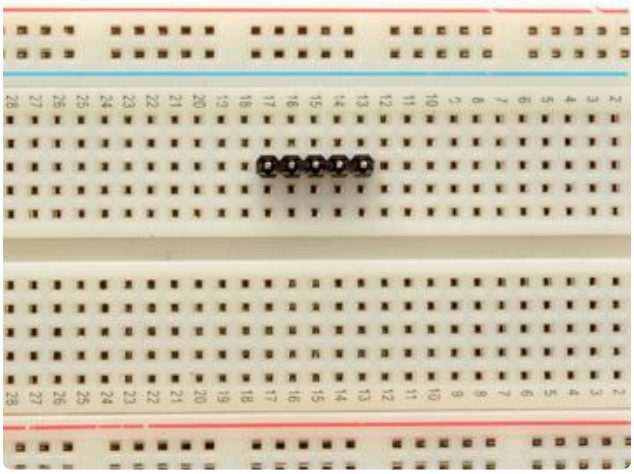
I2C Logic pins:

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

Assembly



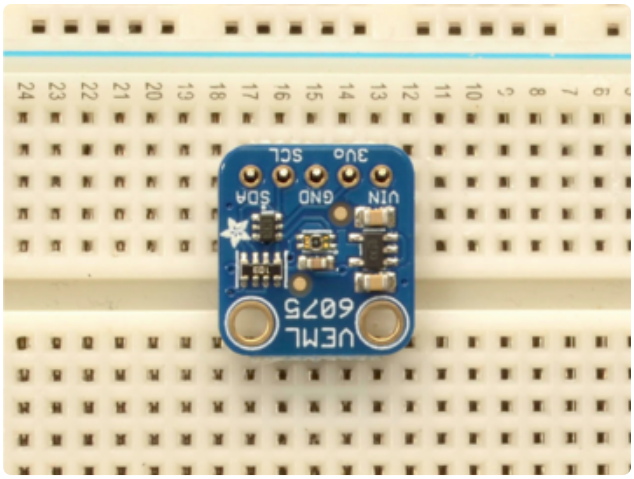




Prepare the header strip:

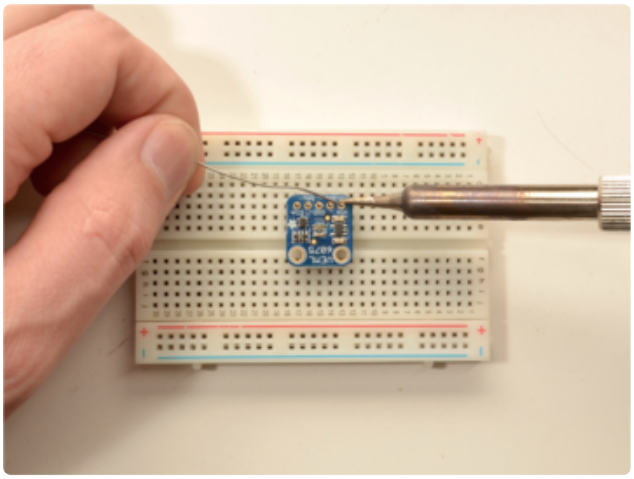
Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - long pins down.

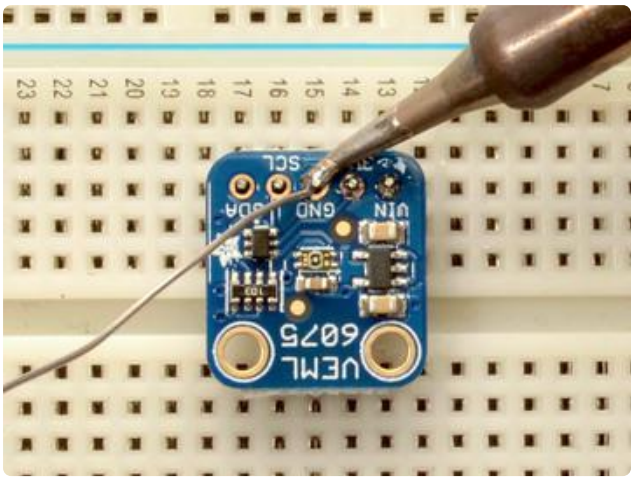
Place the breakout board over the pins so that the short pins poke through the breakout pads.



Add the breakout board:

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

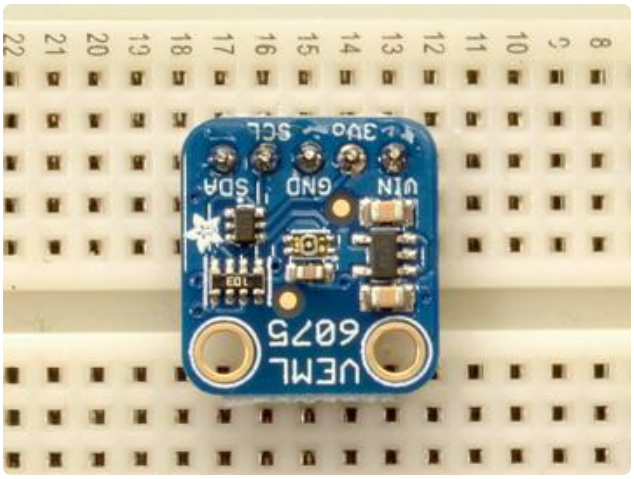




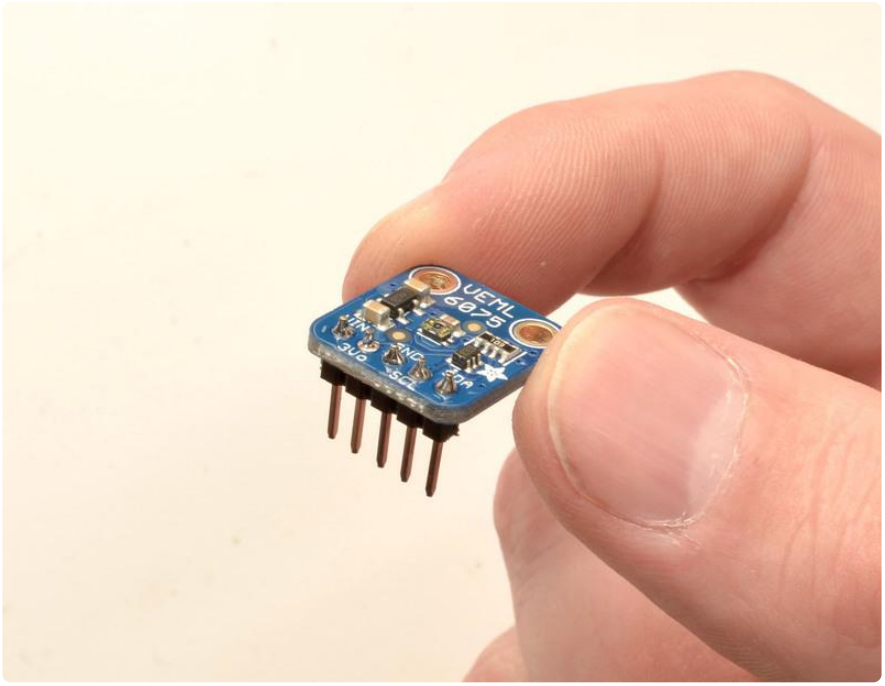
And Solder!

Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our Guide to Excellent Soldering ()).

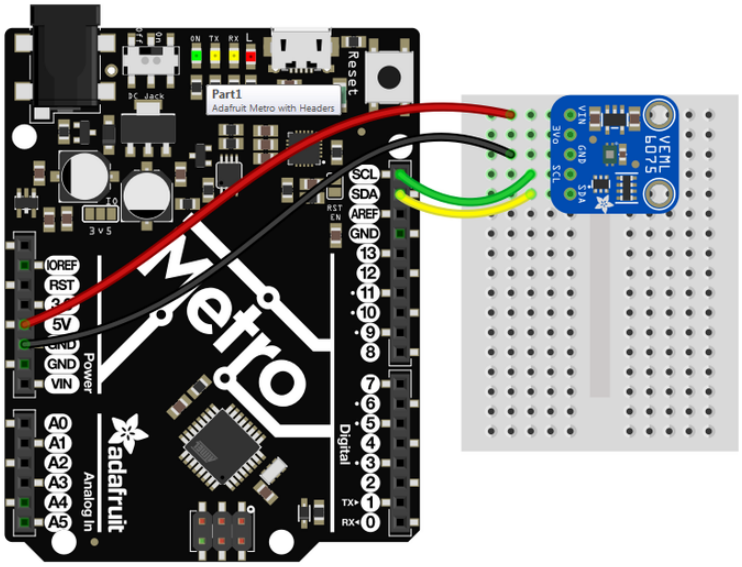


You're done! Check your solder joints visually and continue onto the next steps



Arduino 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!

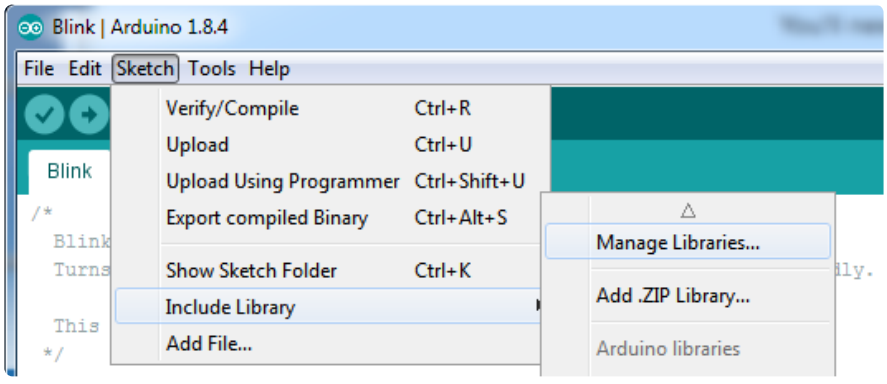


- Connect Vin to the power supply, 3-5V is fine. Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V
- Connect GND to common power/data ground
- Connect the SCL pin to the I2C clock SCL pin on your Arduino. On an UNO & '328 based Arduino, this is also known 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

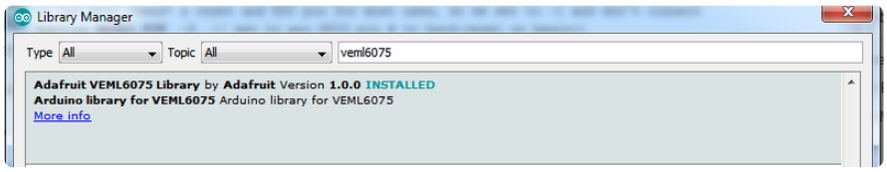
The VEML6075 has a default I2C address of 0x10 and cannot be changed!

Install Arduino Libraries

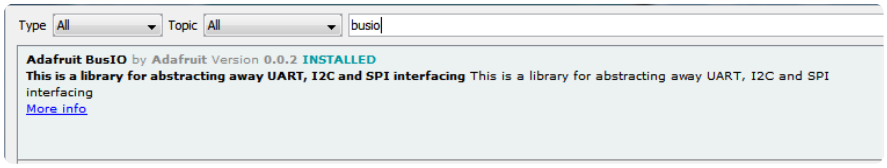
Lets begin by installing all the libraries we need. Open up the library manager in Arduino IDE



Search for and install the latest version of the Adafruit VEML6075 library

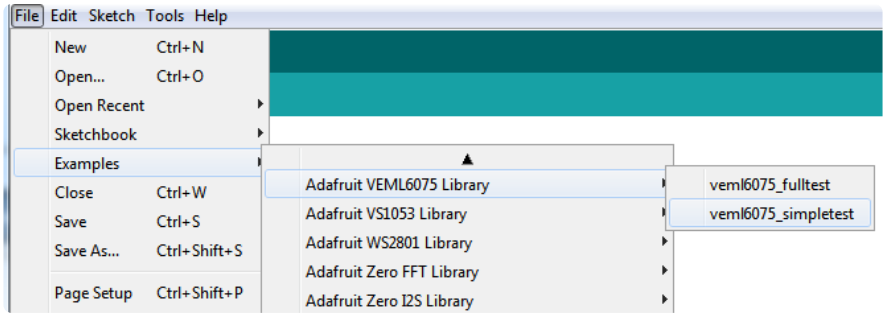


And the Adafruit BusIO library



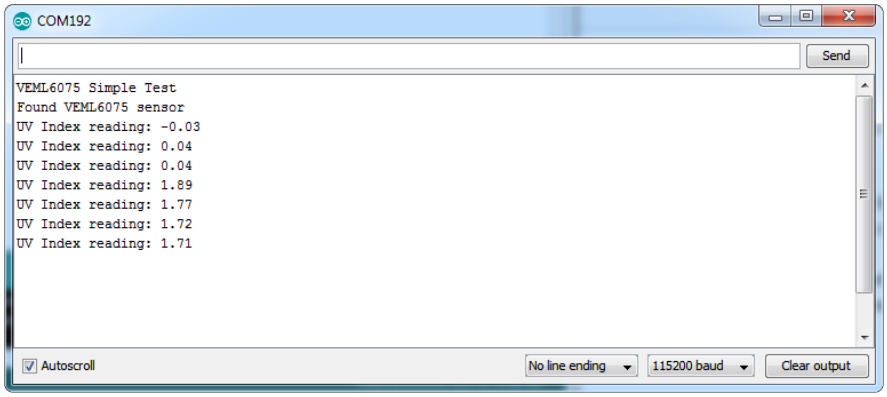
Basic Example

Start by opening up the Adafruit VEML6075 -> simpletest under the Adafruit VEML6075 library:



And upload it to your board!

Open up the serial console to see the readings. I used a UV lamp to shine some UV light on it. Note that indoors with office lighting you may get very low or even negative values.



Advanced settings

For 99% of users, the default configuration is recommended because we use the calibration values from the VEML6075 app note for a non-covered sensor (no glass or teflon filter). You'll get the most accurate results that way.

If you do want to change settings here's what you can adjust:

- Integration time - this will take readings over a shorter or longer period of time, default is 100ms If changed the UV Index calculated value will not be correct anymore
- Normal/High Dynamic mode - We're not sure what this is (it isn't described) but the default is Normal. If changed the UV Index calculated value will not be correct anymore
- Forced/Continuous mode - Whether to get continuous readings or require reading only on request. Default is reading only on request
- UV Coefficients - These are the calibration numbers that will convert the raw UVA/UVB readings into a UV Index. We use the defaults for "No teflon, open air" from the App note for the 6 floating point values. If you plan to cover the sensor with a thin teflon sheet, you can use those other values to configure the calculation

UV COEFFICIENTS AND RESPONSIVITY						
	a	b	c	d	UVA_{resp}	UVB_{resp}
No teflon (open air)	2.22	1.33	2.95	1.74	0.001461	0.002591
0.1 mm teflon 4.5 mm window	2.22	1.33	2.95	1.74	0.002303	0.004686
0.1 mm teflon 5.5 mm window	2.22	1.33	2.95	1.74	0.002216	0.005188
0.1 mm teflon 10 mm window	2.22	1.33	2.95	1.74	0.002681	0.004875
0.25 mm teflon 10 mm window	2.22	1.33	2.95	1.74	0.002919	0.009389
0.4 mm teflon 10 mm window	2.22	1.17	2.95	1.58	0.004770	0.006135
0.7 mm teflon 10 mm window	2.22	1.17	2.95	1.58	0.007923	0.008334
1.0 mm teflon 5.5 mm window	2.55	1.00	3.80	1.10	0.006000	0.003100

Arduino Library Docs

Arduino Library Docs ()

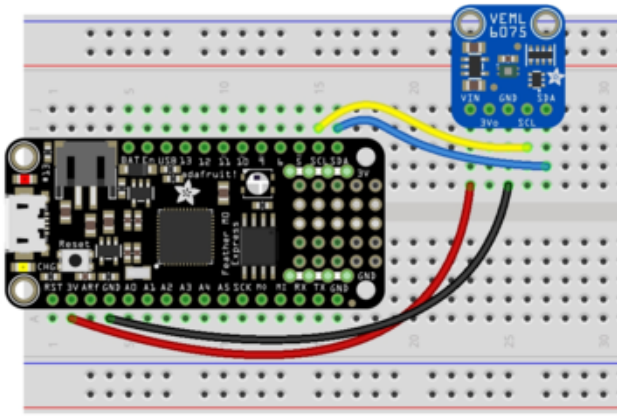
Python & CircuitPython

It's easy to use the VEML6075 sensor with Python or CircuitPython and the Adafruit CircuitPython VEML6075 () module. This module allows you to easily write Python code that reads the UV index 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 VEML6075 to your board exactly as shown on the previous pages for Arduino. You can use I2C. Here's an example of wiring a Feather M0 to the sensor with I2C:

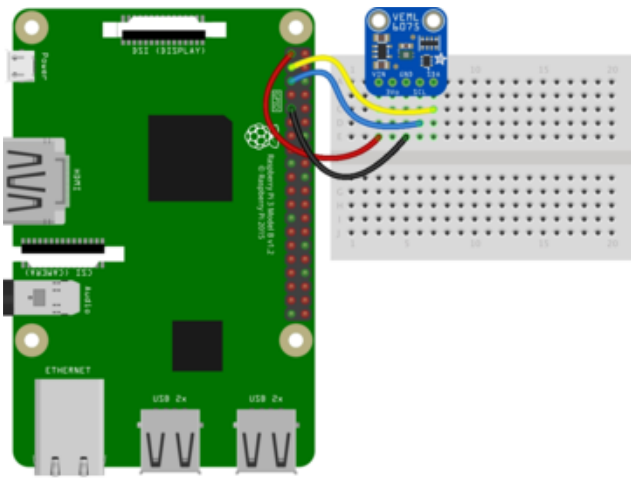


Board 3V to sensor VIN
Board GND to sensor GND
Board SCL to sensor SCL
Board SDA to sensor SDA

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 SCL
Pi SDA to sensor SDA

CircuitPython Installation of VEML6075 Library

You'll need to install the Adafruit CircuitPython VEML6075 () 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 (). Our CircuitPython starter guide has a great page on how to install the library bundle ().

For non-express boards like the Trinket M0 or Gemma M0, you'll need to manually install the necessary libraries from the bundle:

- adafruit_veml6075.mpy
- adafruit_bus_device

Before continuing make sure your board's lib folder or root filesystem has the adafruit_veml6075.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 VEML6075 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-veml6075`

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 & Python Usage

To demonstrate the usage of the sensor we'll initialize it and read the UV index from the board's Python REPL.

Since you're using an I2C connection run the following code to import the necessary modules and initialize the I2C connection with the sensor:

```
import time
import board
import busio
import adafruit_veml6075
i2c = busio.I2C(board.SCL, board.SDA)
veml = adafruit_veml6075.VEML6075(i2c, integration_time=100)
```

The integration_time is the amount of time the VEML6075 is sampling data for, in milliseconds. Valid times are 50, 100, 200, 400 or 800ms. We've chosen to set it to 100 milliseconds.

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

- uv_index - The calculated UV Index.
- uva - The calibrated UVA reading, in 'counts' over the sample period
- uvb - The calibrated UVB reading, in 'counts' over the sample period

For example to print the UV index:

```
print("UV index:", veml.uv_index)
```

```
>>> print("UV index:", veml.uv_index)
UV index: 0.0238851
```

Indoors you will get very low or even negative UV index values! Take your sensor outside into the sun, or use a solar lamp

That's all there is to using the VEML6075 sensor with CircuitPython!

Full Example Code

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

```
import time
import board
import busio
import adafruit_veml6075

i2c = busio.I2C(board.SCL, board.SDA)

veml = adafruit_veml6075.VEML6075(i2c, integration_time=100)

print("Integration time: %d ms" % veml.integration_time)

while True:
    print(veml.uv_index)
    time.sleep(1)
```

Python Docs

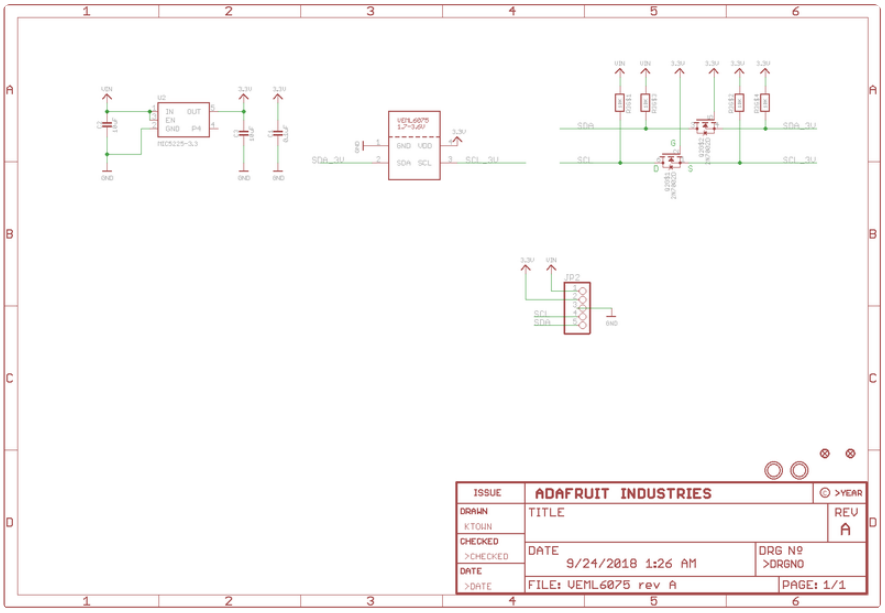
Python Docs ()

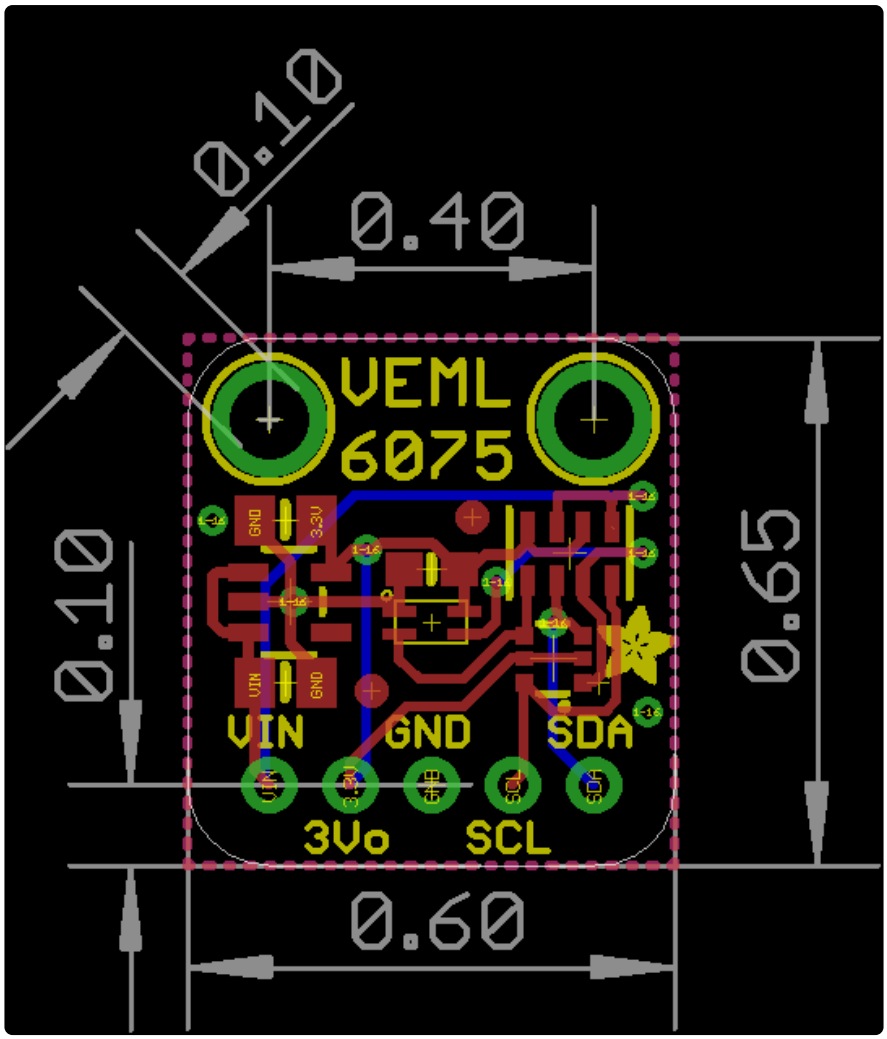
Downloads

Files & Datasheets:

- Latest datasheets and App Notes from Vishay ()
- Fritzing object in Adafruit Fritzing library ()
- PCB CAD files in GitHub ()

Schematic & Fabrication Print




0.10
0.40
VEML
6075
0.10
0.65
VIN
GND
SDA
3Vo
SCL
0.60