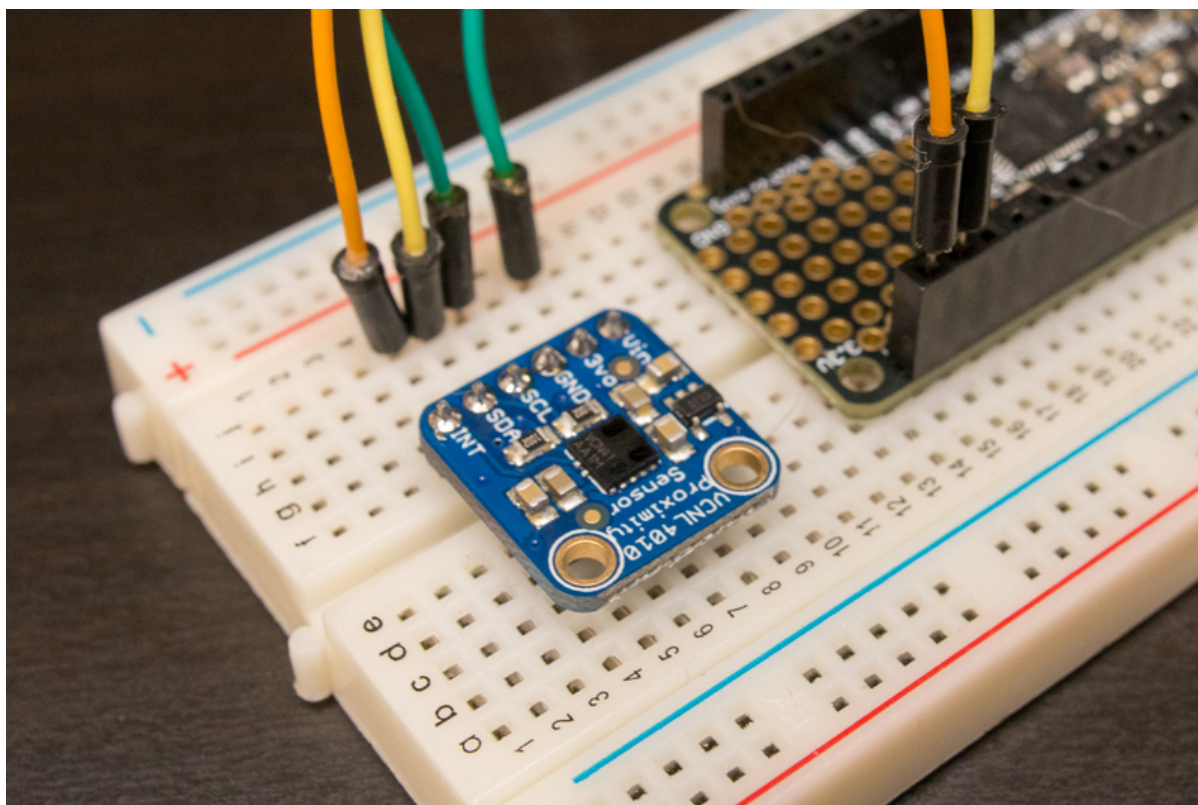




Using VCNL4010 Proximity Sensor

Created by Tony DiCola



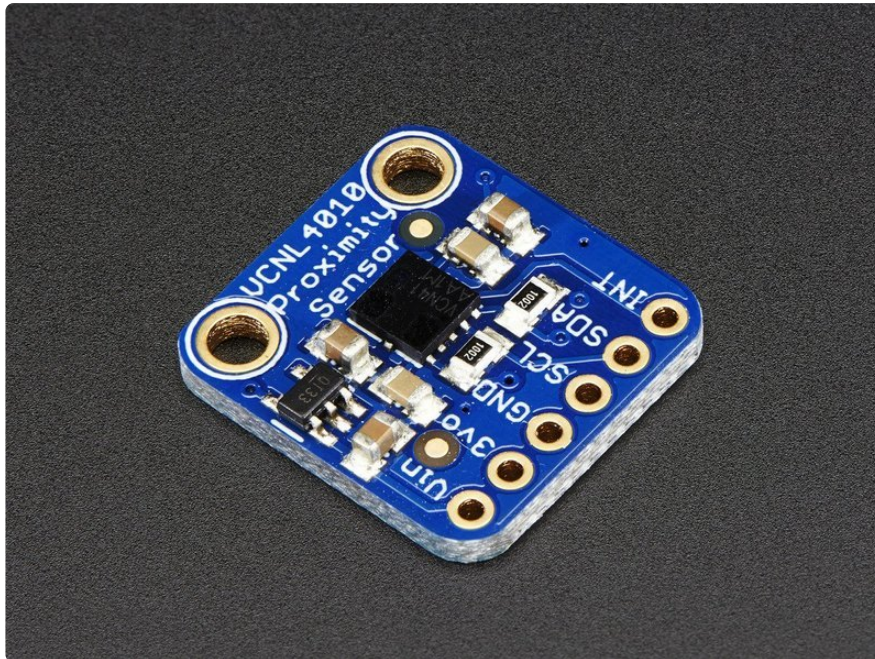
<https://learn.adafruit.com/using-vcnl4010-proximity-sensor>

Last updated on 2024-06-03 02:16:40 PM EDT

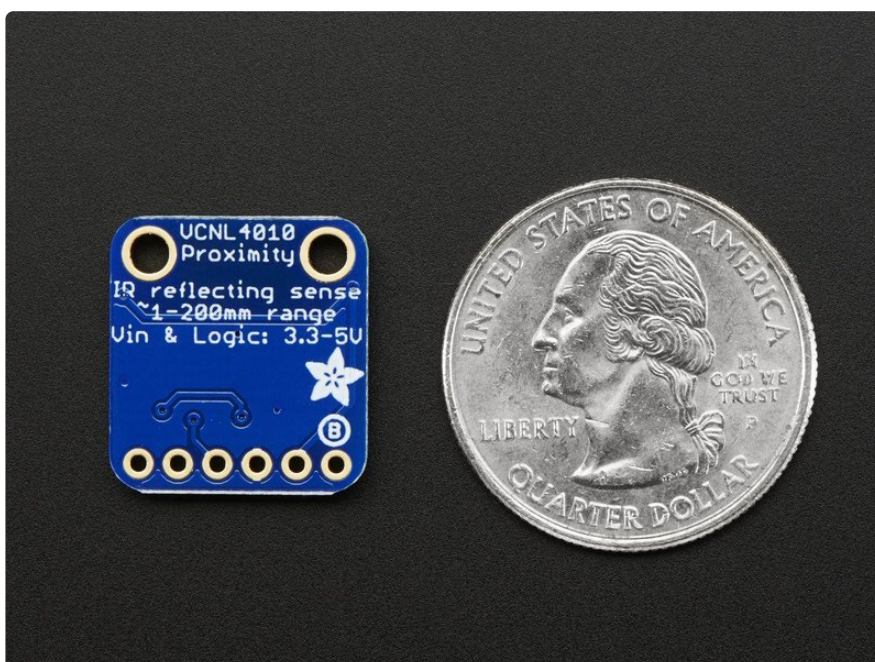
Table of Contents

Overview	3
Pinouts	5
<ul style="list-style-type: none">• Power Pins:• Data Pins	
Arduino	6
<ul style="list-style-type: none">• Install Adafruit_VCNL4010 library• Load Demo	
Arduino Library Docs	8
Python & CircuitPython	8
<ul style="list-style-type: none">• CircuitPython Microcontroller Wiring• Python Computer Wiring• CircuitPython Installation of VCNL4010 Library• Python Installation of VCNL4010 Library• CircuitPython & Python Usage• Full Example Code	
Python Docs	13
Downloads	13
<ul style="list-style-type: none">• Files:• Schematic and Fab Print	

Overview

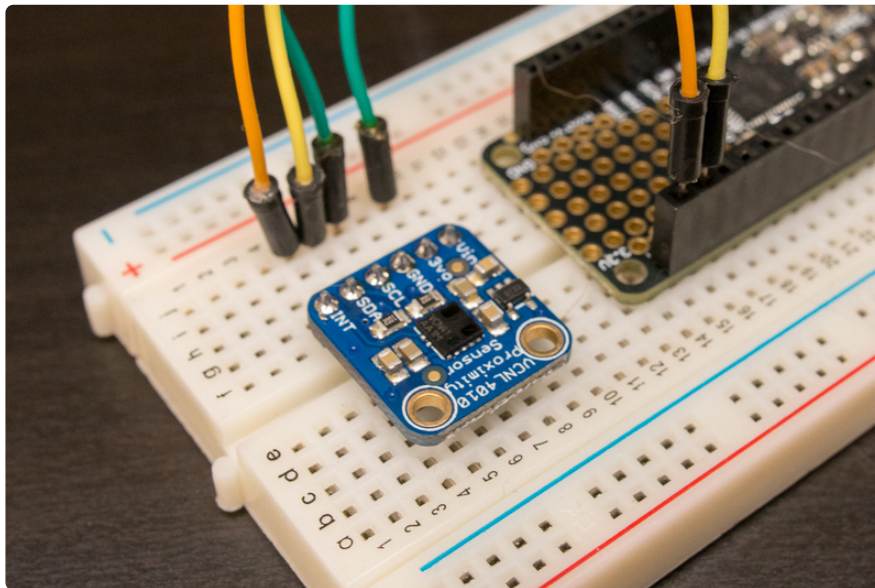


The VCNL4010 sensor is a nice way to add a small-distance proximity sensor to your microcontroller project. For longer distances (in the range of cm, you can use a [SHARP IR distance sensor \(http://adafru.it/164\)](http://adafru.it/164), but those are only good if the object is over 10 cm away. The VCNL4010 is designed for much shorter distances, no more than 200mm (about 7.5") and under our experimentation we found it worked best at distances of about 10-150mm. It would be good for say detecting when a hand moved nearby, or before a robot smacks into a wall. The sensor also has an ambient light sensor built in.

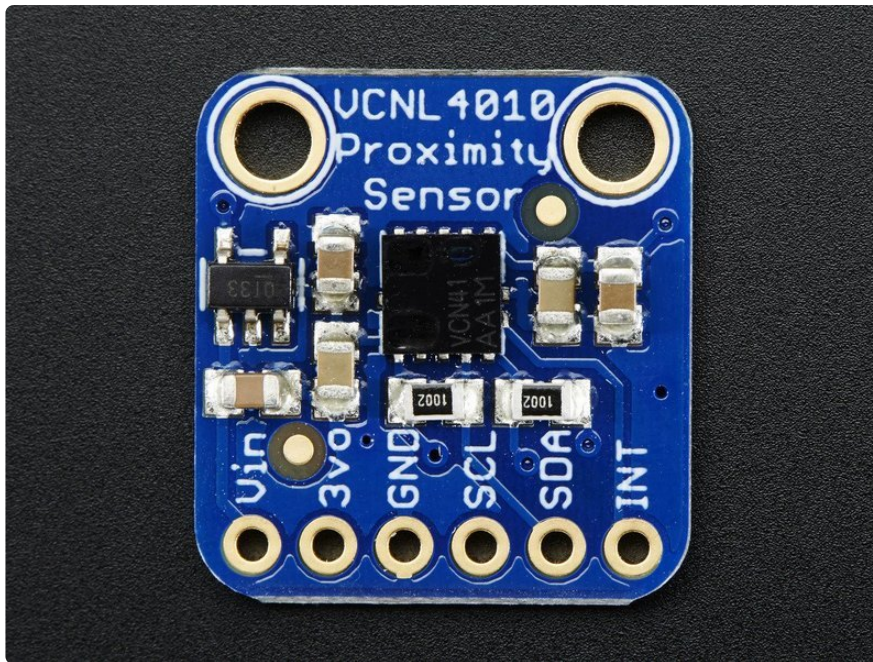


The sensor uses an infrared LED to bounce light off objects in front of it and gauge the returned intensity. You could do all this yourself with LEDs and light sensors, but the VCNL4010 wraps all that logic up into a stand-alone chip for you! With the VCNL4010 you can easily read the proximity (i.e. if an object is near or far) and even ambient light level over a simple I2C connection.

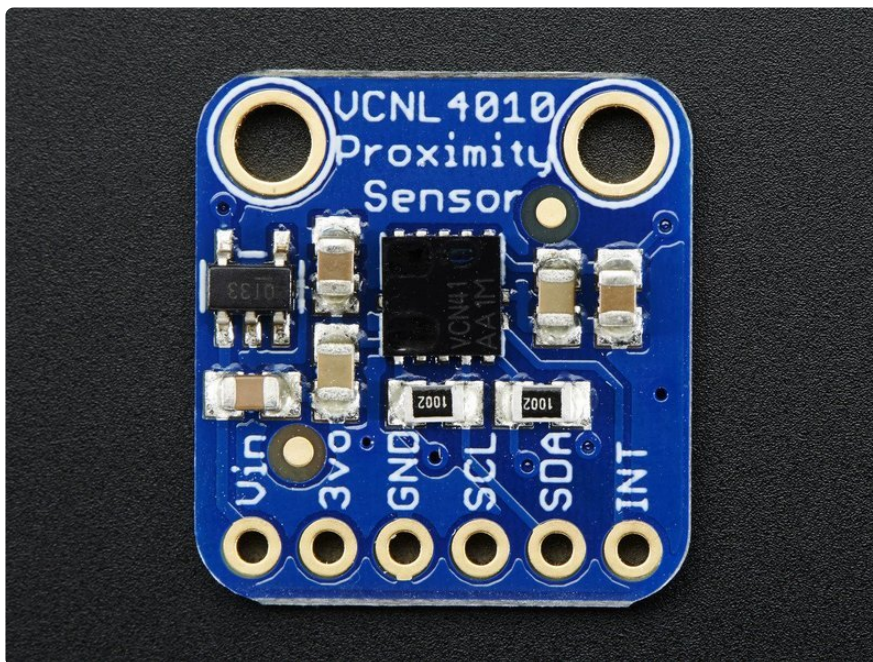
Best of all the with a CircuitPython module you can use the VCNL4010 from Python code! This guide shows how to connect and use the VCNL4010 with a CircuitPython or Arduino board.



This sensor is easy to use with any microcontroller that has i2c capability. It is 5 volt compliant so you can use it with 3.3V or 5V logic with no risk of damage. There is an onboard 3.3V ultra low dropout regulator so you can power it with 3.3 to 5.0V. However, if you can give it 5.0V that is ideal since the VIN voltage powers the IR LED and the higher the voltage you can give it, the more powerful it is.



Pinouts



Power Pins:

- **Vin** - this is the power pin. Since the sensor chip uses 3 VDC for logic, 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 50mA from this if you like

- **GND** - common ground for power and logic

Data Pins

- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line. Can use 3V or 5V logic, and has a weak pullup to **Vin**
 - **SDA** - I2C data pin, connect to your microcontrollers I2C data line. Can use 3V or 5V logic, and has a weak pullup to **Vin**
 - **INT** - Interrupt output, you can use this to give you an alert when the sensor detects something - but it isn't included in our library code
-

Arduino

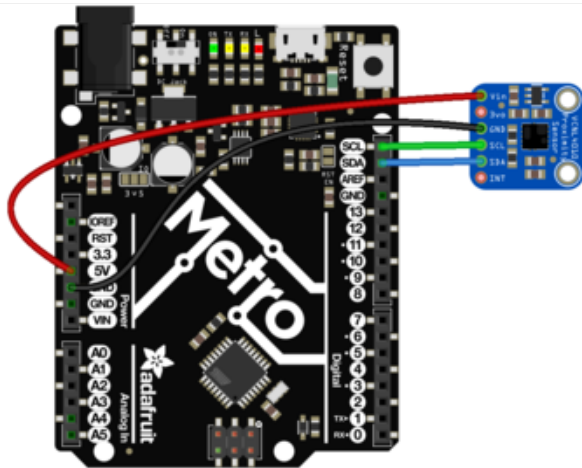
You'll need the following hardware to follow this guide:

- **VCNL4010 sensor** (<http://adafru.it/466>).
- **A board that is Arduino Compatible.** Must have an I2C interface port (often labeled SDA & SCL)
- **Breadboard** (<http://adafru.it/65>) and **jumper wires** (<http://adafru.it/153>). You'll need these parts to connect components to your development board.

Start by soldering the headers to the VCNL4010 breakout board. Check out the [Adafruit guide to excellent soldering \(https://adafru.it/dxy\)](https://adafru.it/dxy) if you're new to soldering. Then continue on below to learn how to wire it to a Metro or Arduino compatible.

The sensor uses **I2C address 0x13** and cannot be changed.

Connect the sensor to your board. The following example shows how to connect it to a Metro:

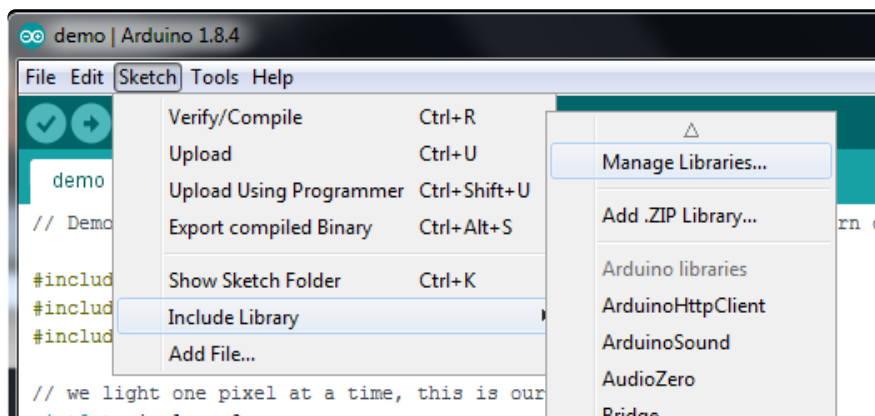


Board 5V to Sensor Vin. (Metro is a 5V logic chip)
 Board ground / GND to sensor ground / GND.
 Board SCL to sensor SCL.
 Board SDA to sensor SDA.

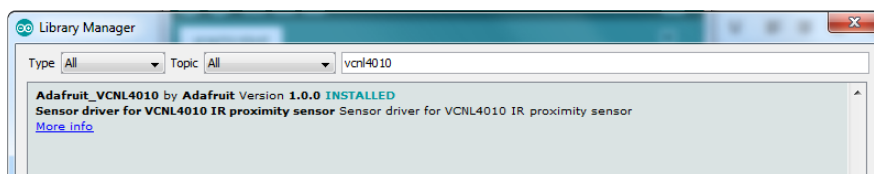
Install Adafruit_VCNL4010 library

To begin reading sensor data, you will need to [install the Adafruit_VCNL4010 library \(code on our github repository\) \(https://adafru.it/C4W\)](https://adafru.it/C4W). It is available from the Arduino library manager so we recommend using that.

From the IDE open up the library manager...

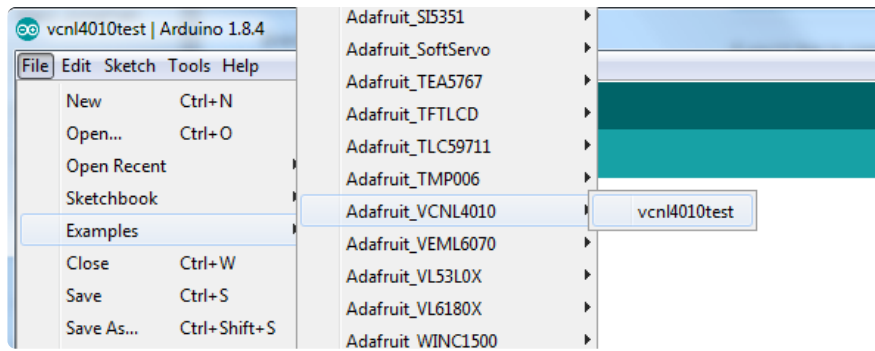


And type in **adafruit vcnl4010** to locate the library. Click **Install**



Load Demo

Open up **File->Examples->Adafruit_VCNL4010->vcnl4010test** and upload to your microcontroller wired up to the sensor



Then open up the serial console at 9600 baud, you'll see ambient light and proximity data printed out! Note that both are unitless, it isn't like the proximity is in millimeters or inches, its just higher the closer you are.



Arduino Library Docs

[Arduino Library Docs \(https://adafru.it/Aod\)](https://adafru.it/Aod)

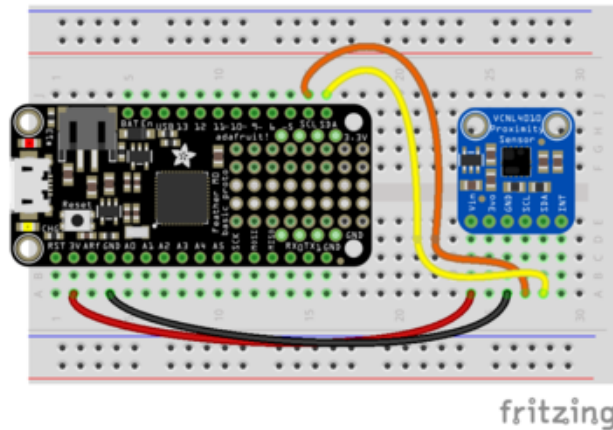
Python & CircuitPython

It's easy to use the VCNL4010 sensor with Python and CircuitPython, and the [Adafruit CircuitPython VCNL4010 \(https://adafru.it/C4X\)](https://adafru.it/C4X) module. This module allows you to easily write Python code that reads small-distance proximity 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 \(https://adafru.it/BSN\)](https://adafru.it/BSN).

CircuitPython Microcontroller Wiring

First wire up a VCNL4010 to your board exactly as shown in the previous pages for Arduino. Here's an example of wiring a Feather M0 to the sensor with I2C:



Board 3.3V to sensor Vin (Feather is 3.3V logic)

Board ground / GND to sensor ground / GND.

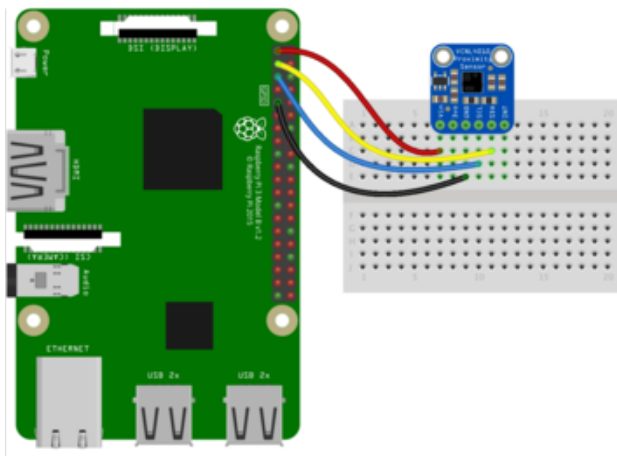
Board SCL to sensor SCL.

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) (<https://adafru.it/BSN>).

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 VCNL4010 Library

To use the VCNL4010 you'll need to install the [Adafruit CircuitPython VCNL4010](https://adafru.it/C4X) (<https://adafru.it/C4X>) library on your CircuitPython board.

First make sure you are running the [latest version of Adafruit CircuitPython](https://adafru.it/Amd) (<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 \(https://adafru.it/zdx\)](https://adafru.it/zdx). Our introduction guide has [a great page on how to install the library bundle \(https://adafru.it/ABU\)](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_vcnl4010.mpy`
- `adafruit_bus_device`

Before continuing make sure your board's lib folder or root filesystem has the `adafruit_vcnl4010.mpy`, and `adafruit_bus_device` files and folders copied over.

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

Python Installation of VCNL4010 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 \(https://adafru.it/BSN\)](https://adafru.it/BSN)!

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

- `sudo pip3 install adafruit-circuitpython-vcnl4010`

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 proximity and ambient light from the 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_vcnl4010
i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_vcnl4010.VCNL4010(i2c)
```

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

- **proximity** - The value of the proximity sensor. This has no units and is **inversely** proportional to the distance of an object from the sensor. I.e. the **farther** an object from the sensor the **lower** this value. The range of values are 0 (very far away, up to ~200mm) to 65535 (right in front of the sensor).
- **ambient_lux** - The detected ambient light as a value in lux.
- **ambient** - The raw ambient light sensor value--this has no units but is proportional to the brightness. You probably want to use the lux property above instead of this raw value!

```
print('Proximity: {}'.format(sensor.proximity))
print('Ambient light: {} lux'.format(sensor.ambient_lux))
```

```
>>> print('Proximity: {}'.format(sensor.proximity))
Proximity: 2320
>>> print('Ambient light: {} lux'.format(sensor.ambient_lux))
Ambient light: 294.75 lux
>>>
```

In addition there are a couple properties you can both read and write to change the behavior of the sensor:

- **led_current** and **led_current_mA** - These properties control the amount of current given to the sensor's LED for proximity detection. The sensor supports 0 - 200mA of current and the `led_current` property is set with units of 10mA (i.e. a value of 20 is 200mA), while the `led_current_mA` property takes the value in milliamperes directly (like 200mA). The default is 200mA of current for the brightest LED.
- **frequency** - This property controls the frequency of measurement timings. You can set it to a value of:
 - `adafruit_vcnl4010.FREQUENCY_3M125` - 3.125 mhz
 - `adafruit_vcnl4010.FREQUENCY_1M5625` - 1.5625 mhz
 - `adafruit_vcnl4010.FREQUENCY_781K25` - 781.25 khz
 - `adafruit_vcnl4010.FREQUENCY_390K625` - 390.625 khz (the default)

Read the [VCNL4010 datasheet \(http://adafru.it/4664010\)](http://adafru.it/4664010) to understand how LED current and measurement frequency impact power consumption and accuracy!

```
sensor.led_current_mA = 120
sensor.frequency = adafruit_vcnl4010.FREQUENCY_3M125
```

```
>>> sensor.led_current_mA = 120
>>> sensor.frequency = adafruit_vcnl4010.FREQUENCY_3M125
>>> print('Proximity: {}'.format(sensor.proximity))
Proximity: 4732
>>> print('Ambient light: {} lux'.format(sensor.ambient_lux))
Ambient light: 294.5 lux
>>>
```

That's all there is to using the VCNL4010 sensor with CircuitPython!

Here's a complete example of reading the proximity and ambient light every second and printing it at the serial terminal. Save this as **code.py** on your board and open the serial terminal to see the output.

Full Example Code

```
# SPDX-FileCopyrightText: 2017 Tony DiCola for Adafruit Industries
# SPDX-License-Identifier: MIT

# Simple demo of the VCNL4010 proximity and light sensor.
# Will print the proximity and ambient light every second.
import time
import board
import adafruit_vcnl4010

i2c = board.I2C() # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C() # For using the built-in STEMMA QT connector on a
# microcontroller
sensor = adafruit_vcnl4010.VCNL4010(i2c)

# You can optionally adjust the sensor LED current. The default is 200mA
# which is the maximum value. Note this is only set in 10mA increments.
# sensor.led_current_mA = 120 # Set 120 mA LED current

# You can also adjust the measurement frequency for the sensor. The default
# is 390.625 khz, but these values are possible to set too:
# - FREQUENCY_3M125: 3.125 Mhz
# - FREQUENCY_1M5625: 1.5625 Mhz
# - FREQUENCY_781K25: 781.25 Khz
# - FREQUENCY_390K625: 390.625 Khz (default)
# sensor.frequency = adafruit_vcnl4010.FREQUENCY_3M125 # 3.125 Mhz

# Main loop runs forever printing the proximity and light level.
while True:
    proximity = sensor.proximity # Proximity has no units and is a 16-bit
    # value. The LOWER the value the further
    # an object from the sensor (up to ~200mm).
    print("Proximity: {}".format(proximity))
    ambient_lux = sensor.ambient_lux
    print("Ambient light: {} lux".format(ambient_lux))
    time.sleep(1.0)
```

Python Docs

[Python Docs \(https://adafru.it/C4-\)](https://adafru.it/C4-)

Downloads

Files:

- [VCNL4010 Datasheet / App Note / Product Sheet \(http://adafru.it/4664010\)](http://adafru.it/4664010)
- [EagleCAD PCB files on GitHub \(https://adafru.it/C50\)](https://adafru.it/C50)
- [Fritzing library \(https://adafru.it/aP3\)](https://adafru.it/aP3)

Datasheets for the older VCNL4000 version of this board:

- [VCNL4000 datasheet \(https://adafru.it/C51\)](https://adafru.it/C51)
- [VCNL4000 App Note \(https://adafru.it/C52\)](https://adafru.it/C52)
- [VCNL4000 Product sheet \(https://adafru.it/C53\)](https://adafru.it/C53)

Schematic and Fab Print

