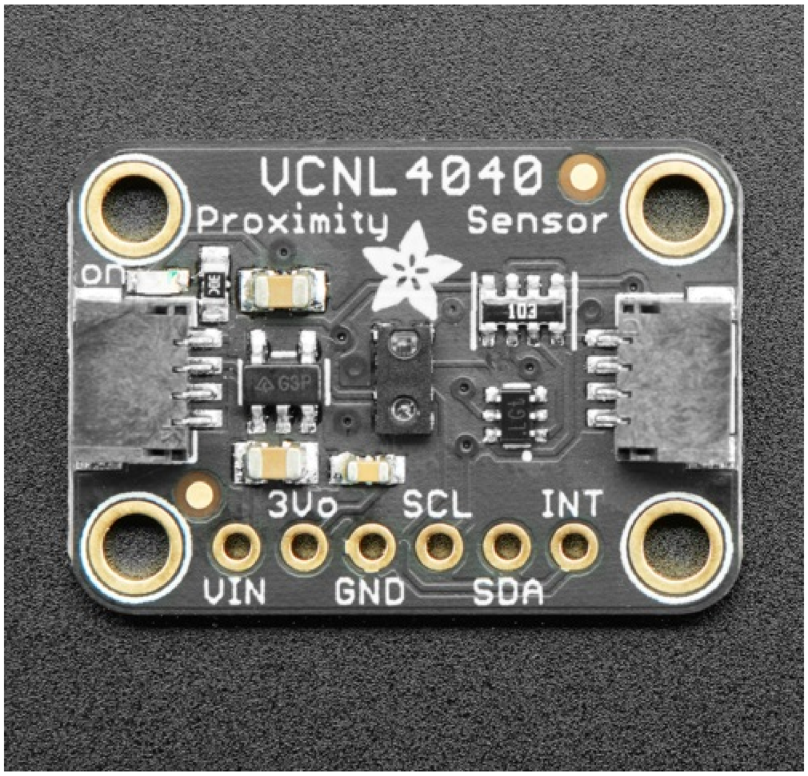
adafruit learning system

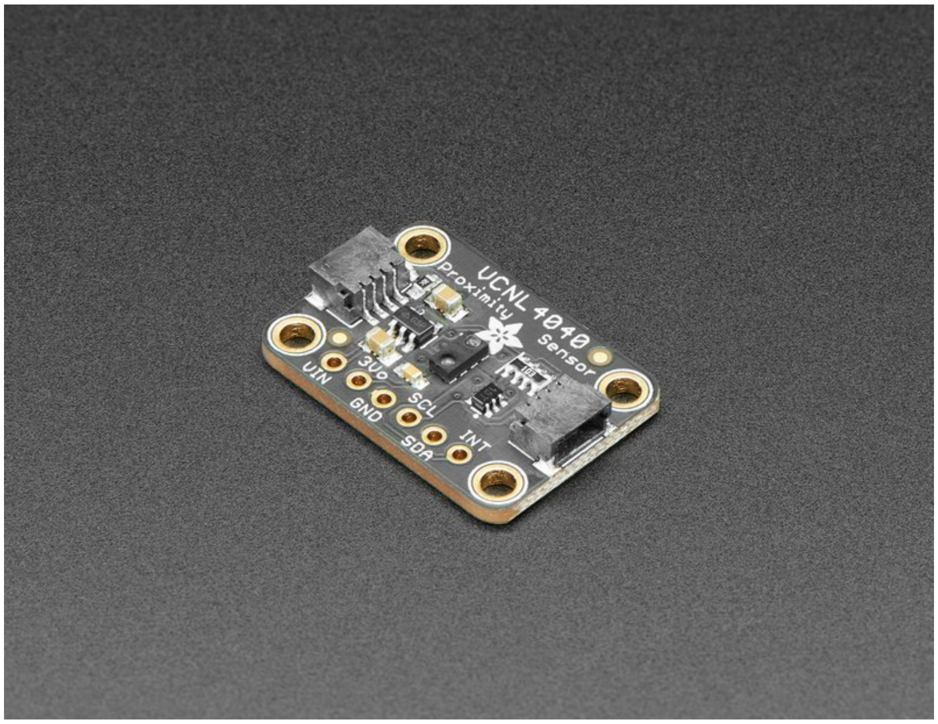
Adafruit VCNL4040 Proximity Sensor

Created by Bryan Siepert



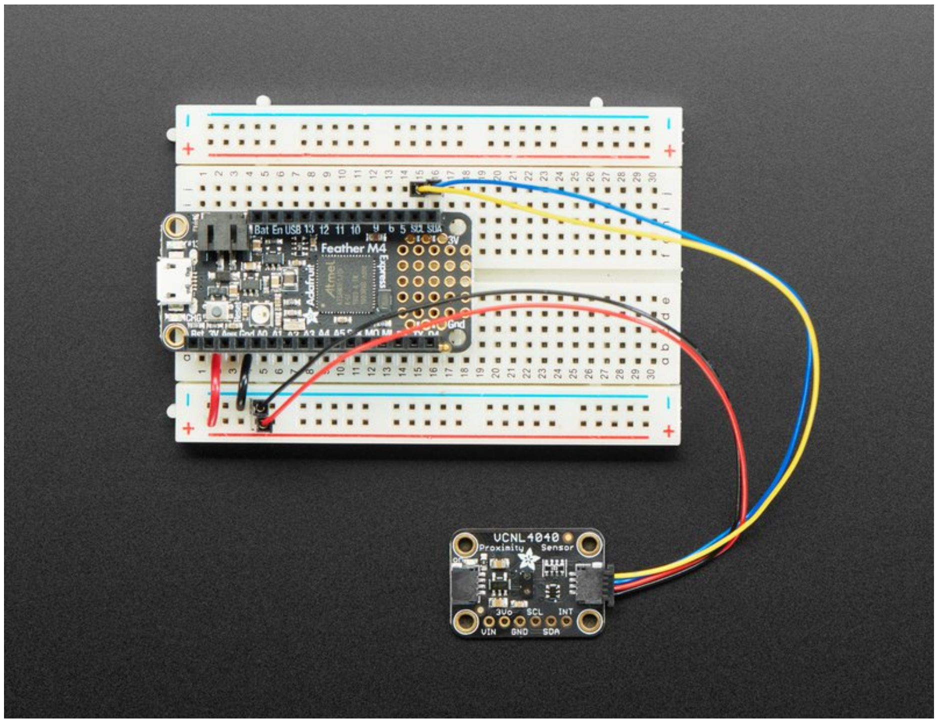
Last updated on 2019-09-04 03:53:14 AM UTC

Overview

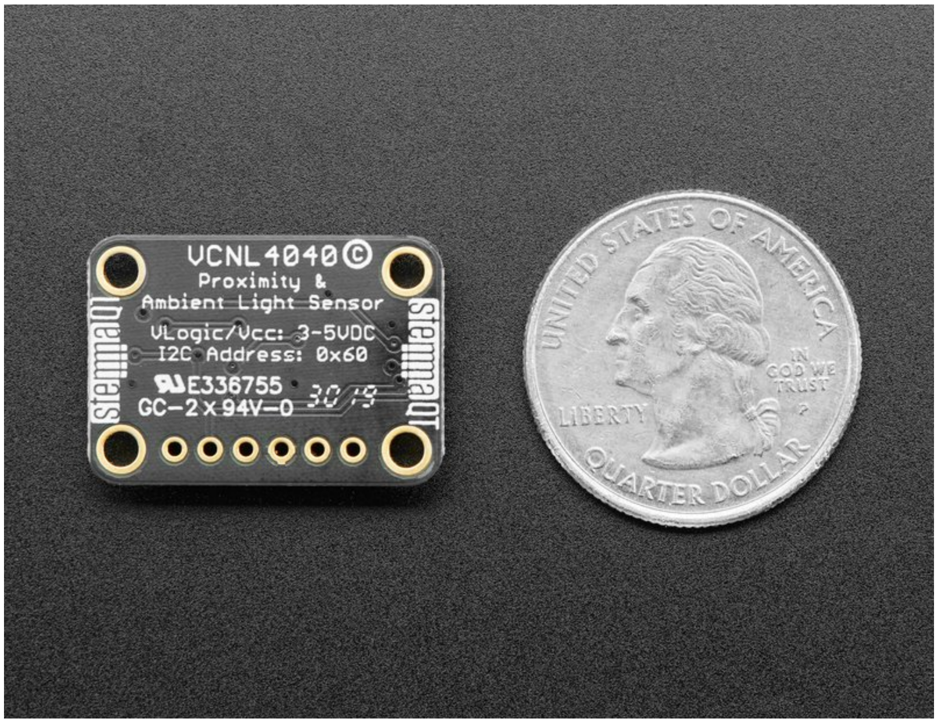


The VCNL4040 is a handy two-in-one sensor, with a proximity sensor that works from 0 to 200mm (about 7.5 inches) and light sensor with range of 0.0125 to 6553 lux.

We've all been there. That thing is *close* but *how close?* When you need to measure a small distance with reasonable accuracy, such as the rough height of particularly calm bumble bee, the VCNL4040 Proximity Sensor from Vishay can do that for you. If perchance you also needed to measure the amount of light at the same time, perhaps to let the bee to know if it's time for bed, you're in luck! The VCNL4040 can do that too (bumble bee not included, we tried putting it in the anti-static bag but it started buzzing in a threatening manner).



"OK, *finally* I can get started on my bee measuring and light sensing project, but *how do I use it?"* you say. To make life easier so you can focus on your important work, we've taken the VCNL4040 and put it onto a breakout PCB along with support circuitry to let you use this little wonder with 3.3V (Feather) or 5V (Arduino/ Metro328) logic levels. Additionally, since it speaks I2C, you can easily connect it up with two wires (plus power and ground!). We've even included SparkFun qwiic (https://adafru.it/Ftc) compatible **STEMMA QT (https://adafru.it/Ft4)** connectors for the I2C bus so **you don't even need to solder!** Just wire up to your favorite micro and you can use our CircuitPython/Python or Arduino drivers to easily interface with the VCNL4040 and make approximate approximations of proximity in no time!



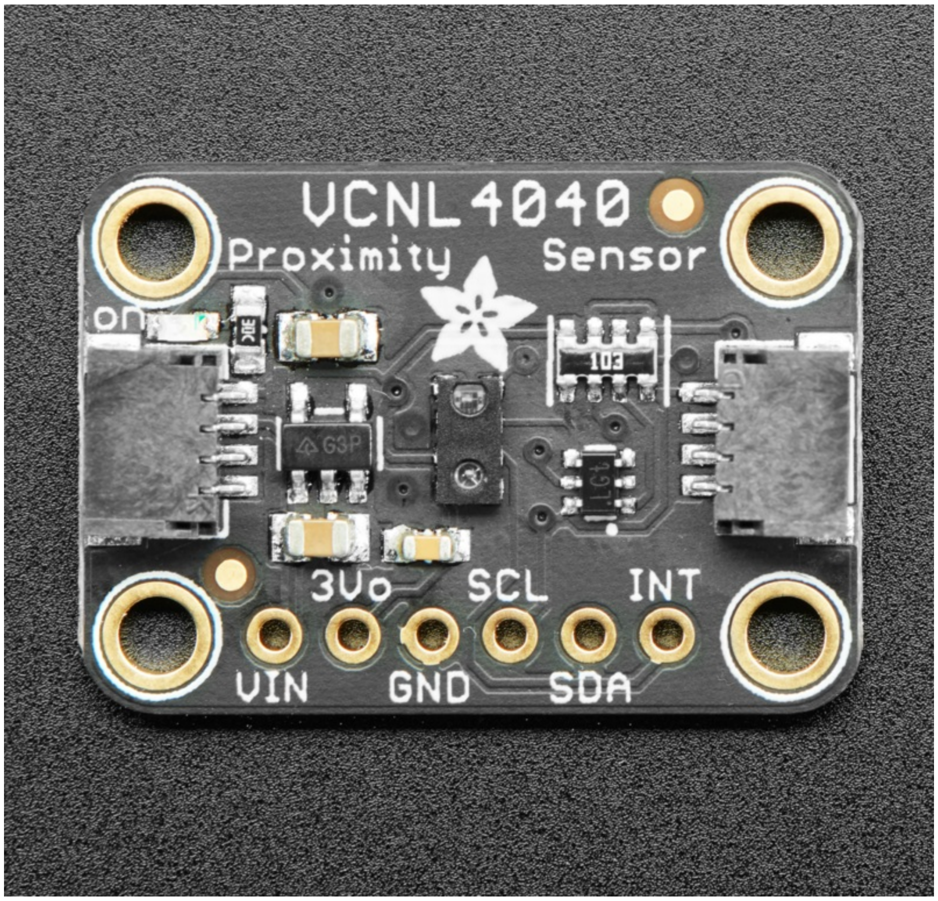
To give you some ability to tune your measurements for your situation, the VCNL4040 lets you adjust the integration time (how long light is being measured) to set the sensitivity of your measurements depending on the lighting conditions. With an integration time of 80ms, since the sensor is only collecting measurements for a short period of time, each measured bit of light represents a larger amount of light. For this reason the maximum range in lux for the 80ms integration time is 6553.5 lux.

If you use a longer integration time and get the same amount of light measurements "events", you know that there is comparatively less light because we're waiting around longer for light to show up and getting the same number, meaning the rate of light events is lower. For this reason at the maximum integration time of 640ms the VCNL4040 will only measure up to 819.2 lux.

It is also worth keeping in mind that as you increase the integration time you are also increasing the sensitivity and resolution within the measurement range

In addition to adjusting the sensitivity of light measurements, you can change the current and duty cycle of the IR LED that powers the proximity detection to adjust how sensitive proximity measurements are

Pinouts



Power Pins

- **Vin** - this is the power pin. Since the sensor 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

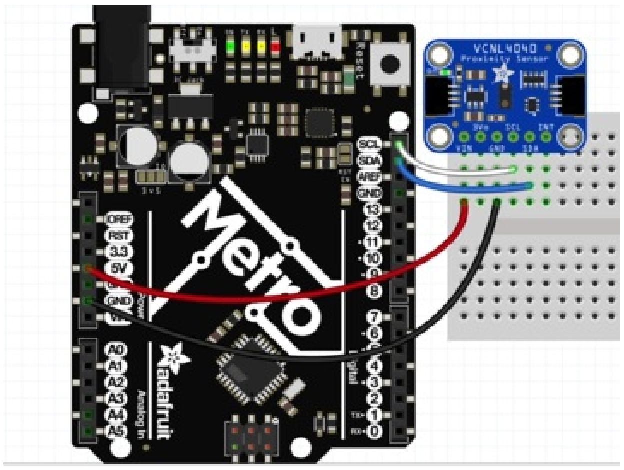
I2C Logic Pins

- **SCL** - this is the I2C clock pin, connect to your microcontroller's I2C clock line.
- **SDA** - this is the I2C data pin, connect to your microcontroller's I2C data line
- **STEMMA QT (https://adafru.it/Ft4) -** These connectors allow you to connectors to dev boards with **STEMMA QT** connectors or to other things with various associated accessories (https://adafru.it/Ft6)

Other Pins

- **INT** -This is the interrupt pin. You can setup the VCNL4040 to pull this low when certain conditions are met such as proximity or lux level thresholds being crossed.

Arduino

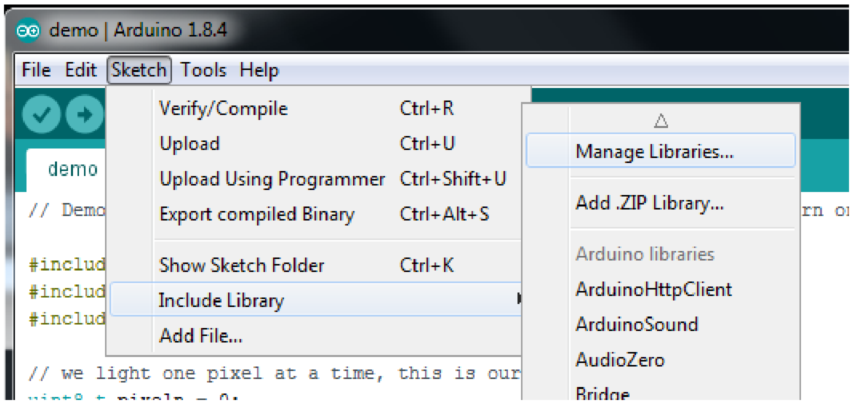


- Connect **Metro or Arduino 5V** to **board VIN**
- Connect **Metro or Arduino GND** to **board GND**
- Connect **Metro or Arduino SCL** to **board SCL**
- Connect **Metro or Arduino SDA** to **board SDA**

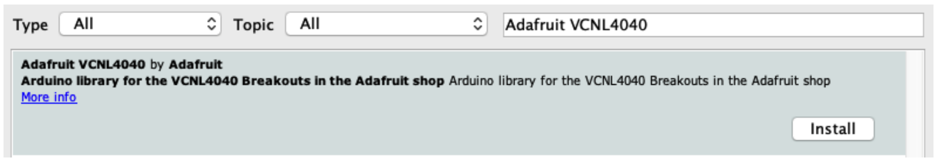
The final results should resemble the illustration above, showing an Adafruit Metro (https://adafru.it/METROXMETR) development board.

Installation

You can install the **Adafruit VCNL4040 Library** for Arduino using the Library Manager in the Arduino IDE:

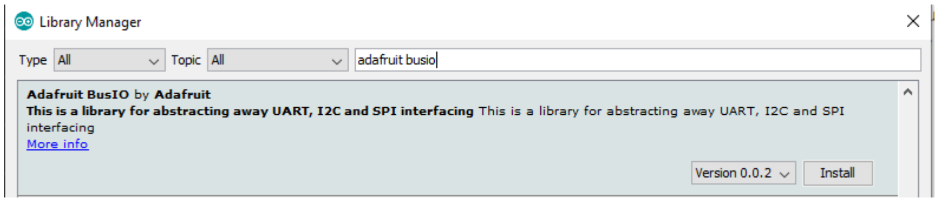


Click the **Manage Libraries ...** menu item, search for **Adafruit VCNL4040**, and select the **Adafruit VCNL4040** library:



Adafruit VCNL4040 by **Adafruit**
Arduino library for the VCNL4040 Breakouts in the Adafruit shop Arduino library for the VCNL4040 Breakouts in the Adafruit shop

Then follow the same process for the **Adafruit BusIO** library.

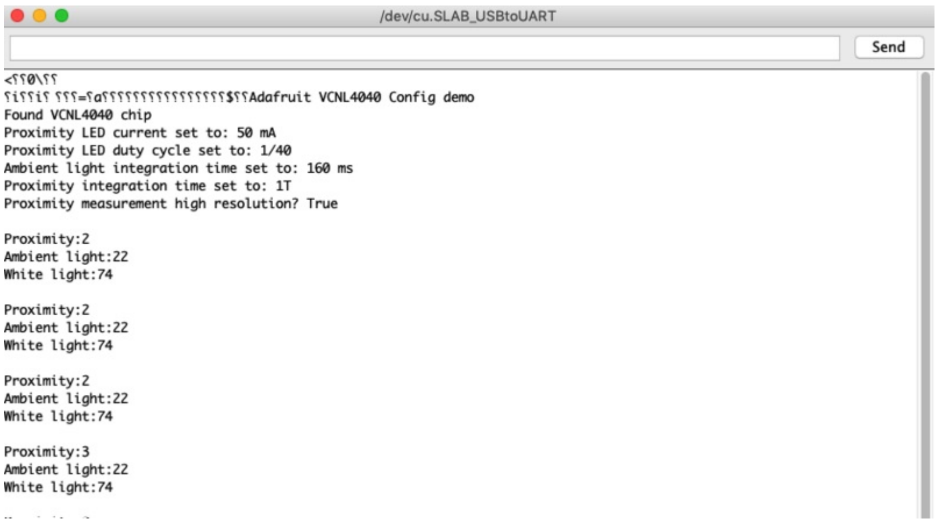


Adafruit BusIO by Adafruit
This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing
More info

Load Example

Open up **File -> Examples -> Adafruit VCNL4040 -> vcnl4040_test** and upload to your Arduino wired up to the sensor.

Upload the sketch to your board and open up the Serial Monitor (**Tools->Serial Monitor**) at 115200 baud. You should see the the values for the current configuration settings printed on startup, followed by readings for ambient light, proximity levels, and raw white light readings similar to this:



```
<ʕʕ0\ʕʕ
ʕiʕʕiʕ ʕʕʕ=ʕaʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕ$ʕʕAdafruit VCNL4040 Config demo
Found VCNL4040 chip
Proximity LED current set to: 50 mA
Proximity LED duty cycle set to: 1/40
Ambient light integration time set to: 160 ms
Proximity integration time set to: 1T
Proximity measurement high resolution? True

Proximity:2
Ambient light:22
White light:74

Proximity:2
Ambient light:22
White light:74

Proximity:2
Ambient light:22
White light:74

Proximity:3
Ambient light:22
White light:74
```

You can place your hand close to the sensor to see the proximity value change, and to see the ambient light levels change you can either cover the sensor or shine a light at it.

Example Code

The following code is part of the standard library and illustrates the basic function of sensing light levels and measuring proximity:

```
#include <Adafruit_VCNL4040.h>

Adafruit_VCNL4040 vcnl4040 = Adafruit_VCNL4040();

void setup() {
  Serial.begin(115200);
  // Wait until serial port is opened
  while (!Serial) { delay(1); }

  Serial.println("Adafruit VCNL4040 Config demo");

  if (!vcnl4040.begin()) {
    Serial.println("Couldn't find VCNL4040 chip");
    while (1);
  }
  Serial.println("Found VCNL4040 chip");

  //vcnl4040.setProximityLEDCurrent(VCNL4040_LED_CURRENT_200MA);
  Serial.print("Proximity LED current set to: ");
  switch(vcnl4040.getProximityLEDCurrent()) {
    case VCNL4040_LED_CURRENT_50MA: Serial.println("50 mA"); break;
    case VCNL4040_LED_CURRENT_75MA: Serial.println("75 mA"); break;
    case VCNL4040_LED_CURRENT_100MA: Serial.println("100 mA"); break;
    case VCNL4040_LED_CURRENT_120MA: Serial.println("120 mA"); break;
    case VCNL4040_LED_CURRENT_140MA: Serial.println("140 mA"); break;
    case VCNL4040_LED_CURRENT_160MA: Serial.println("160 mA"); break;
    case VCNL4040_LED_CURRENT_180MA: Serial.println("180 mA"); break;
    case VCNL4040_LED_CURRENT_200MA: Serial.println("200 mA"); break;
  }

  //vcnl4040.setProximityLEDDutyCycle(VCNL4040_LED_DUTY_1_40);
  Serial.print("Proximity LED duty cycle set to: ");
  switch(vcnl4040.getProximityLEDDutyCycle()) {
    case VCNL4040_LED_DUTY_1_40: Serial.println("1/40"); break;
    case VCNL4040_LED_DUTY_1_80: Serial.println("1/80"); break;
    case VCNL4040_LED_DUTY_1_160: Serial.println("1/160"); break;
    case VCNL4040_LED_DUTY_1_320: Serial.println("1/320"); break;
  }

  //vcnl4040.setAmbientIntegrationTime(VCNL4040_AMBIENT_INTEGRATION_TIME_80MS);
  Serial.print("Ambient light integration time set to: ");
  switch(vcnl4040.getAmbientIntegrationTime()) {
    case VCNL4040_AMBIENT_INTEGRATION_TIME_80MS: Serial.println("80 ms"); break;
    case VCNL4040_AMBIENT_INTEGRATION_TIME_160MS: Serial.println("160 ms"); break;
    case VCNL4040_AMBIENT_INTEGRATION_TIME_320MS: Serial.println("320 ms"); break;
    case VCNL4040_AMBIENT_INTEGRATION_TIME_640MS: Serial.println("640 ms"); break;
  }


  //vcnl4040.setProximityIntegrationTime(VCNL4040_PROXIMITY_INTEGRATION_TIME_8T);
  Serial.print("Proximity integration time set to: ");
  switch(vcnl4040.getProximityIntegrationTime()) {
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_1T: Serial.println("1T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_1_5T: Serial.println("1.5T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_2T: Serial.println("2T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_2_5T: Serial.println("2.5T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_3T: Serial.println("3T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_3_5T: Serial.println("3.5T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_4T: Serial.println("4T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_8T: Serial.println("8T"); break;
  }

  //vcnl4040.setProximityHighResolution(false);
  Serial.print("Proximity measurement high resolution? ");
  Serial.println(vcnl4040.getProximityHighResolution() ? "True" : "False");

  Serial.println("");

}

void loop() {

  Serial.print("Proximity:"); Serial.println(vcnl4040.getProximity());
  Serial.print("Ambient light:"); Serial.println(vcnl4040.getLux());
  Serial.print("Raw white light:"); Serial.println(vcnl4040.getWhiteLight());
  Serial.println("");

  delay(500);
}
```

Arduino Docs

Arduino Docs (https://adafru.it/FJE)

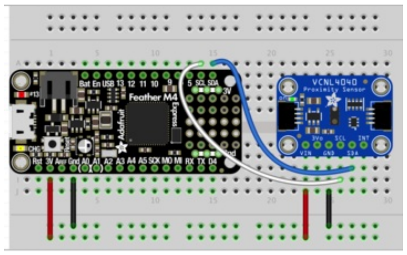
Python & CircuitPython

It's easy to use the VCNL4040 sensor with CircuitPython and the Adafruit CircuitPython VCNL4040 (https://adafru.it/Ftd) module. This module allows you to easily write Python code that reads the ambient light levels, white light levels, and proximity data from the breakout.

You can use this sensor with any CircuitPython microcontroller board or with a Linux single board 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 VCNL4040 to your board exactly as follows. Here is an example of the VCNL4040 wired to a Feather (https://adafru.it/Cmy) using I2C:



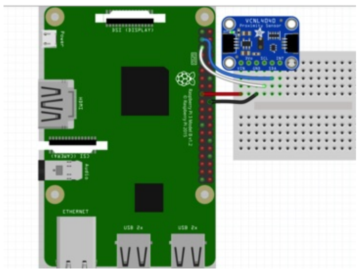
- **Board 3V** to **sensor VIN**
- **Board GND** to **sensor GND**
- **Board SCL** to **sensor SCL**
- **Board SDA** to **sensor SDA**

Note: This breakout includes pullup resistors on the I2C lines, no external pullups are required.

Python Computer Wiring

Since there's *dozens* of Linux computers/boards you can use we will show wiring for Raspberry Pi (https://adafru.it/scY). 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:



- **Pi 3V3** to **sensor VIN**
- **Pi GND** to **sensor GND**
- **Pi SCL** to **sensor SCL**
- **Pi SDA** to **sensor SDA**

CircuitPython Installation of VCNL4040 Library

You'll need to install the Adafruit CircuitPython VCNL4040 (https://adafru.it/Ftd) 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/uap). Our CircuitPython starter guide has a great page on how to install the library bundle (https://adafru.it/ABU).

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

- **adafruit_vcnl4040.mpy**
- **adafruit_bus_device**
- **adafruit_register**

Before continuing make sure your board's lib folder or root filesystem has the **adafruit_vcnl4040.mpy, adafruit_bus_device**, and **adafruit_register** files and folders copied over.

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

Python Installation of VCNL4040 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)!

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

- `sudo pip3 install adafruit-circuitpython-vcnl4040`

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!

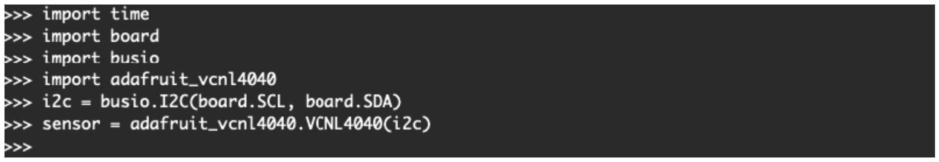
CircuitPython & Python Usage

To demonstrate the usage of the sensor we'll initialize it and read the light levels and proximity measurements from the board's Python REPL.

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_vcnl4040

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_vcnl4040.VCNL4040(i2c)
```

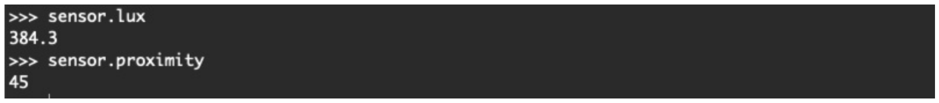


```
>>> import time
>>> import board
>>> import busio
>>> import adafruit_vcnl4040
>>> i2c = busio.I2C(board.SCL, board.SDA)
>>> sensor = adafruit_vcnl4040.VCNL4040(i2c)
>>>
```

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

- **lux -** The ambient light data in lux
- **proximity** - The proximity data measurement

For example to print ambient light level and proximity info:



```
>>> sensor.lux
384.3
>>> sensor.proximity
45
```

For more details, check out the library documentation (https://adafru.it/Fte).

That's all there is to using the VCNL4040 sensor with CircuitPython!

Full Example Code

```
import time
import board
import busio
import adafruit_vcnl4040

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_vcnl4040.VCNL4040(i2c)

while True:
    print("Proximity:", sensor.proximity)
    print("Light: %d lux"% sensor.lux)
    time.sleep(1.0)
```

Python Docs

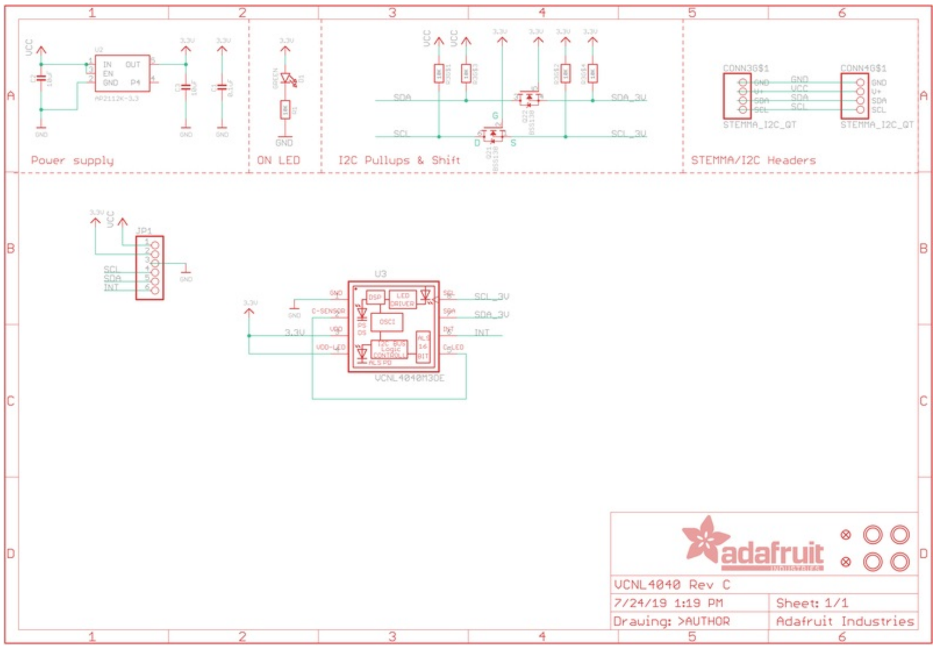
Python Docs (https://adafru.it/FiU)

Downloads

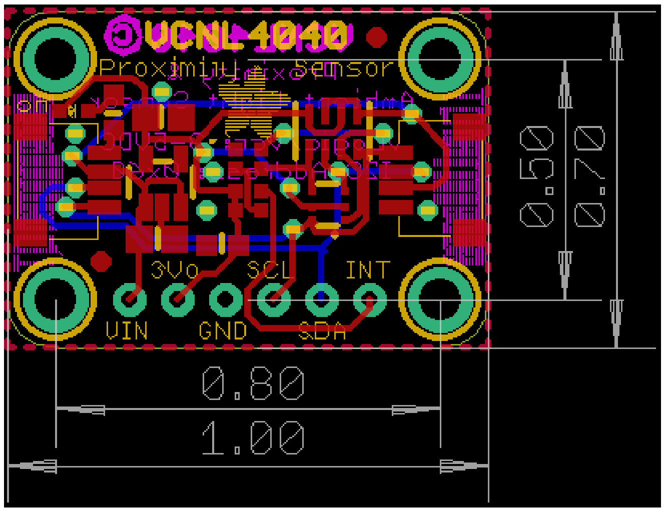
Files

- VCNL4040 Datasheet (https://adafru.it/Ftf)
- EagleCAD files on GitHub (https://adafru.it/Ftg)
- Fritzing object from Adafruit Fritzing Library (https://adafru.it/Fth)

Schematic



Fab Print


VCNL4040
Proximity Sensor
3Vo
SCL
INT
VIN
GND
SDA
0.50
0.70
0.80
1.00