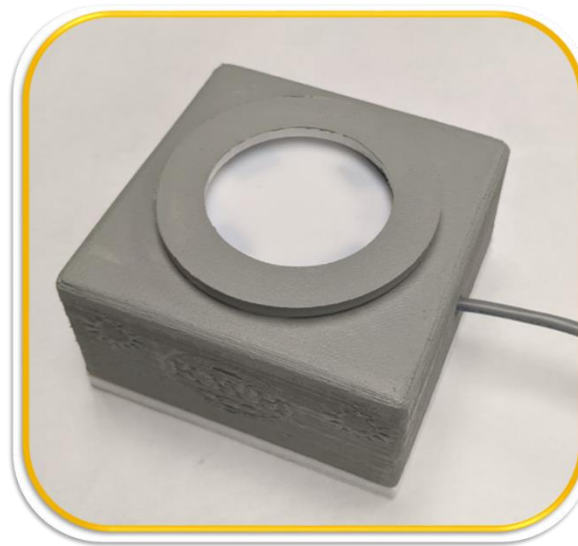




# **User Manual of a Low-Cost Pyranometer**

**The Solar Sailors**

## **The PV Matey**



**Adriana Oliveira, Brady Jerome, Lukas Vermeulen,  
Nickolas Borek**  
UNIVERSITY OF IDAHO, COLLEGE OF ENGINEERING

# Contents

<b>Background .....</b>	<b>1</b>
<b>Materials .....</b>	<b>1</b>
<b>Sensor.....</b>	<b>1</b>
<b>Circuit Assembly.....</b>	<b>3</b>
<b>Calibration Circuit.....</b>	<b>4</b>
<b>Calibration Circuit Assembly.....</b>	<b>5</b>
<b>Final Circuit Assembly.....</b>	<b>5</b>
<b>Case .....</b>	<b>8</b>
<b>Case Assembly .....</b>	<b>9</b>
<b>Using Arduino IDE.....</b>	<b>10</b>
<b>Connecting the Adafruit QT Py to Arduino IDE.....</b>	<b>10</b>
<b>Code.....</b>	<b>12</b>
<b>Code Setup.....</b>	<b>12</b>
<b>Calibration .....</b>	<b>12</b>
<b>Results and Tables.....</b>	<b>16</b>
<b>Photocurrent Testing .....</b>	<b>16</b>
<b>Filtering .....</b>	<b>17</b>
<b>Calibration Results .....</b>	<b>17</b>
<b>Appendix.....</b>	<b>19</b>
<b>Component Links.....</b>	<b>19</b>
<b>Connecting to Arduino IDE.....</b>	<b>20</b>
<b>Links to More Information.....</b>	<b>20</b>
<b>Soldering .....</b>	<b>20</b>
<b>Wiki Page Link .....</b>	<b>20</b>
<b>Code.....</b>	<b>20</b>

## Background

KiloWatts for Humanity, a non-profit organization that works to bring renewable energy to energy deficient countries, contacted the University of Idaho to sponsor a project. This project was to build a low-cost pyranometer that would be able to measure the solar irradiance of light hitting their solar panels. This pyranometer is used to determine if the solar panels are working properly, or if they need to be repaired or cleaned.

The team deployed by the University of Idaho, known as the Solar Sailors, developed the pyranometer as part of their Senior Design class. The team consists of Adriana Oliveira (EE), Brady Jerome (EE), Lukas Vermeulen (EE), and Nickolas Borek (ME).

## Materials

Figure 1 shows the bill of materials for this project. It details the full product and its four subassemblies: Sunlight Capture, Microcontroller, Output to Data Logger, and Case. The links for each component listed in the bill of materials are found in the Appendix.

PV Matey Bill of Materials			Manufacturing Plan						Cost		
Part No. or Assy No.	Description	Quantity	Make or Buy?	Source (or Fabrication Location)	Fabrication Technique (if appl.)	Is the Drawing Avail.?	Is the Material Avail.?	Who will lead (Name)?	Target Completion Date	Unit Cost (\$)	Extended Cost (\$)
<b>Full Assy 1</b>											
Subassembly 1	Sunlight Capture	1									
Subassembly 2	Microcontroller	1									
Subassembly 3	Output to Data Logger	1									
Subassembly 4	Case	1									
<b>Subassembly 1 Sunlight Capture</b>											
Part 1	SFH 3310 Phototransistor	1	Buy	Mouser	N/A	No	No	Lukas V.	1-Mar	\$0.86	\$0.86
Part 2	Teflon Sheet*	1	Buy	ePlastics	N/A	No	No	Lukas V.	1-Mar	\$3.08	\$3.08
Part 3	Acrylic Disk	1	Buy	US plastics	N/A	No	No	Lukas V.	1-Mar	\$1.72	\$1.72
Part 4	680 Ω Resistor	1	Buy	Mouser	N/A	No	No	Lukas V.	1-Mar	\$0.10	\$0.10
<b>Subassembly 2 Microcontroller</b>											
Part 5	Adafruit QT py	1	Buy	Adafruit	N/A	No	No	Adriana O. and Brady J.	1-Mar	\$6.00	\$6.00
Part 6	1N914 Diode	1	Buy	Mouser	N/A	No	No	Lukas V.	1-Mar	\$0.10	\$0.10
Part 7	5V, 1A Power Supply	1	Buy	Amazon	N/A	No	No	Lukas V.	1-Mar	\$7.89	\$7.89
<b>Subassembly 3 Output to Data Logger</b>											
Part 8	36 Ω Resistor	4	Buy	Mouser	N/A	No	No	Lukas V.	1-Mar	\$0.10	\$0.40
Part 9	Op Amp LM358P	1	Buy	Mouser	N/A	No	No	Lukas V.	1-Mar	\$0.41	\$0.41
Part 10	22 AWG 3-Conductor Copper Wire	100	Buy	Mouser	N/A	No	No	Lukas V.	1-Mar	\$0.29	\$29.00
<b>Subassembly 4 Case</b>											
Part 11	Circuitry Housing*	1	Make	Personal Equipment	3D Print from PETG	Yes	Yes	Nick B.	1-Mar	\$0.20	\$0.20
Part 12	Housing Lid*	1	Make	Personal Equipment	3D Print from PETG	Yes	Yes	Nick B.	1-Mar	\$0.20	\$0.20
Part 13	Acrylic Disk Cap*	1	Make	Personal Equipment	3D Print from PETG	Yes	Yes	Nick B.	1-Mar	\$0.20	\$0.20
Part 14	USB Cord Seal: Upper Piece*	1	Make	Personal Equipment	3D Print from PETG	Yes	Yes	Nick B.	1-Mar	\$0.10	\$0.10
Part 15	USB Cord Seal: Lower Piece*	1	Make	Personal Equipment	3D Print from PETG	Yes	Yes	Nick B.	1-Mar	\$0.10	\$0.10
Estimate *										Total BOM Cost	\$ 50.36

**Figure 1: PV Matey Bill of Materials**

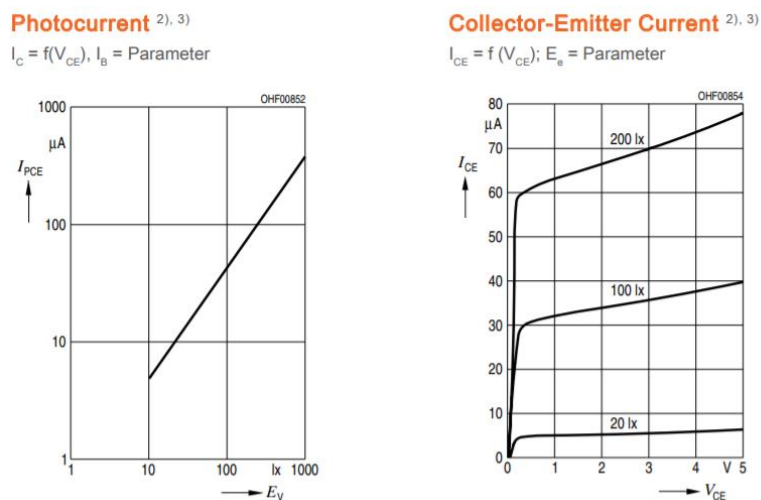
## Sensor

The PV Matey uses the SFH 3310 phototransistor. This sensor has a viewing angle of 150° and rated for 1000 lux. This sensor takes in light and outputs a current depending on the amount of light it receives. This is used to find the amount of irradiance in the area to check how much sunlight the solar panels are receiving. With this sensor, it is possible to capture a good range of visible light to get an accurate reading on the amount of solar irradiance in the area.

Figure 2 shows an image of the sensor. Figure 3 shows the Photocurrent and Collector-Emitter Current of the sensor. Figure 4 shows the Relative Spectral Sensitivity of the sensor. Figure 4 also shows that our sensor's spectral range is 350-970 nm wavelengths, which is roughly the spectrum of visible light. This sensor is used because the spectral range of a solar panel is roughly the same range as our phototransistor. This means they react similarly to visible light, while omitting the infrared and ultraviolet spectrum, giving the user more accurate data. See the datasheet for the SFH 3310 in the appendix for more information on the sensor.



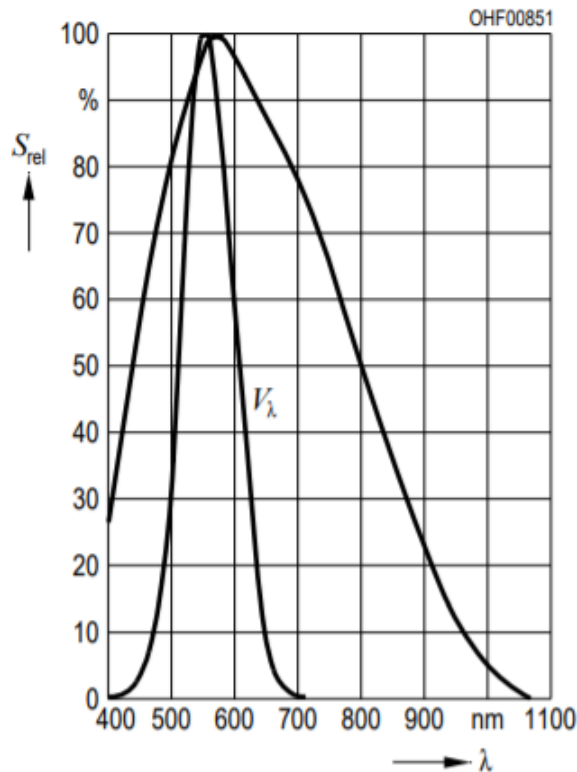
**Figure 2:** SFH 3310 Phototransistor Sensor



**Figure 3:** Photocurrent and Collector-Emitter Current for the SFH 3310 Phototransistor

### Relative Spectral Sensitivity <sup>1) page 8</sup>

$$S_{\text{rel}} = f(\lambda)$$



**Figure 4:** Relative Sensitivity of the SFH 3310 Phototransistor

## Circuit Assembly

The circuit consists of a diode, the sensor, resistors, an op amp, and the microcontroller. The diode is connected between the power source and the rest of the circuit, with the anode towards the power source. This is done to properly power the QT Py using the 5V pin. The phototransistor's collector (the long leg) and "V+" leg of the op amp are also connected to the cathode of the diode.

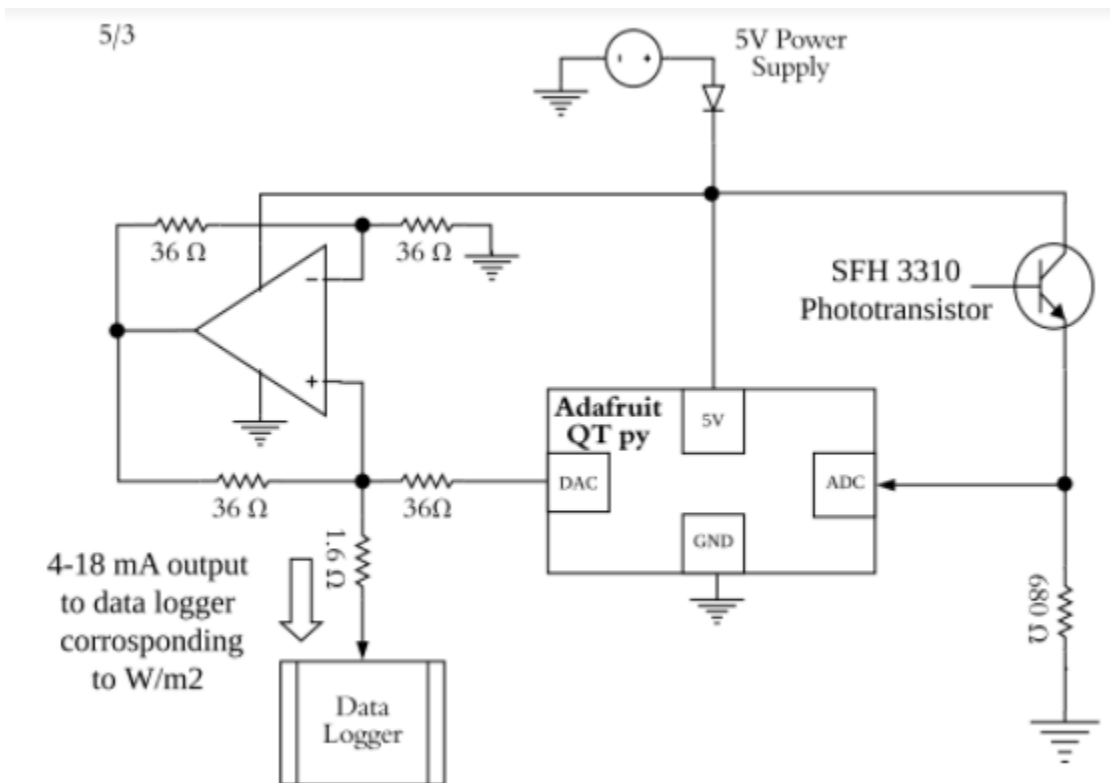
The phototransistor produces a current based on the light hitting the sensor. The QT Py cannot directly read current, so a resistor is placed in series with the phototransistor and the QT Py measures the voltage across the resistor. The resistor voltage is proportional to the current produced by the phototransistor through Ohm's law.

The QT Py uses first-order linear regression to convert the resistor voltage to the equivalent irradiance in  $W/m^2$ . This process is discussed in the Calibration section. The irradiance value lies on a range of 0-2,000  $W/m^2$ . After the QT Py calculates the irradiance value from the resistor voltage, the irradiance value from 0-2,000  $W/m^2$  is multiplied by 1023/2000. This converts the irradiance to a

value that the digital-to-analog converter (DAC) can output since the range of the QT Py's DAC, found on pin A0, is 0-1023. The new value,  $V_{out}$ , from 0-1023, that corresponding to of  $0-2,000 W/m^2$  is sent to the DAC with the command,

`"analogWrite(A0, V_out);"`.

Then the voltage-to-current converter converts the output voltage from the DAC to the equivalent current. The  $1.6\Omega$  resistor represents the equivalent resistance of 100 of wire to the data logger. The output current ranges from 4-18mA. For more information on the voltage-to-current converter and its operation, see the Appendix section. Figure 5 shows the circuit schematic for this product.



**Figure 5:** PV MATEY Circuit Schematic

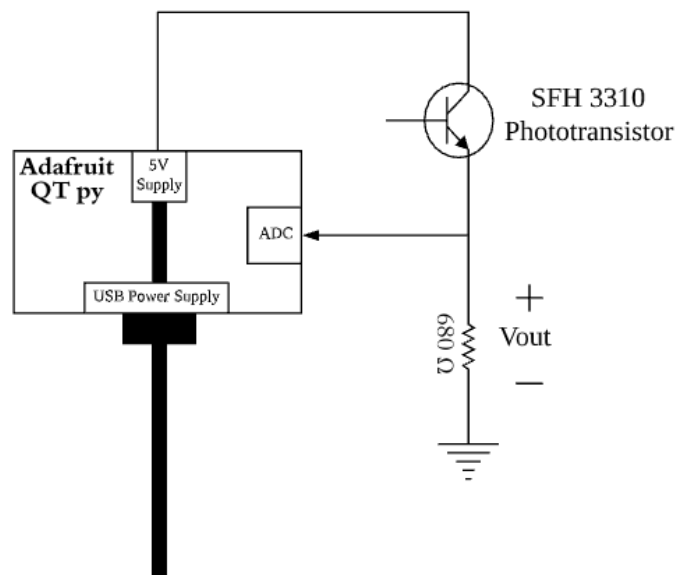
### Calibration Circuit

The calibration circuit differs from the full circuit because it does not utilize the voltage-to-current converter. This circuit can be implemented on a bread board using case\_rev1.2 instead of a printed circuit board (PCB) so the user can easily change out components if needed. The PCB from the final circuit assembly can be used if the 5V pin is not connected to a power source or if the power source is disconnected.

### Calibration Circuit Assembly

1. Connect the collector (long leg) of the phototransistor to the 5V supply pin of the QT Py.
2. Connect the emitter is connected in series with a  $680\Omega$  resistor and to pin A1 of the QT Py.
3. Plug the QT Py to the computer using a USB C cable.

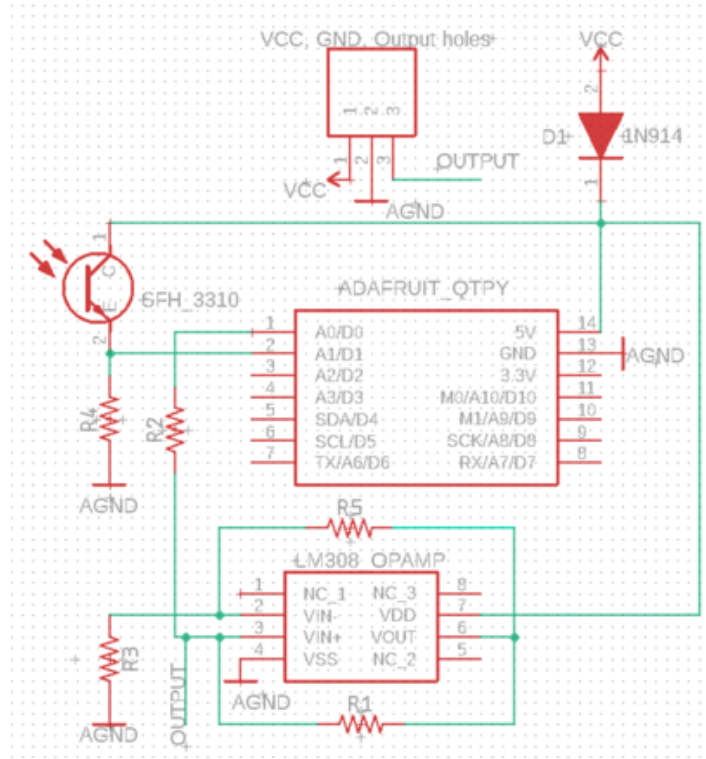
The QT Py reads the voltage across the resistor as the current produced by the phototransistor changes. See the Calibration section for further details on how to properly calibrate the sensor. Figure 6 shows the circuit schematic for the calibration circuit.



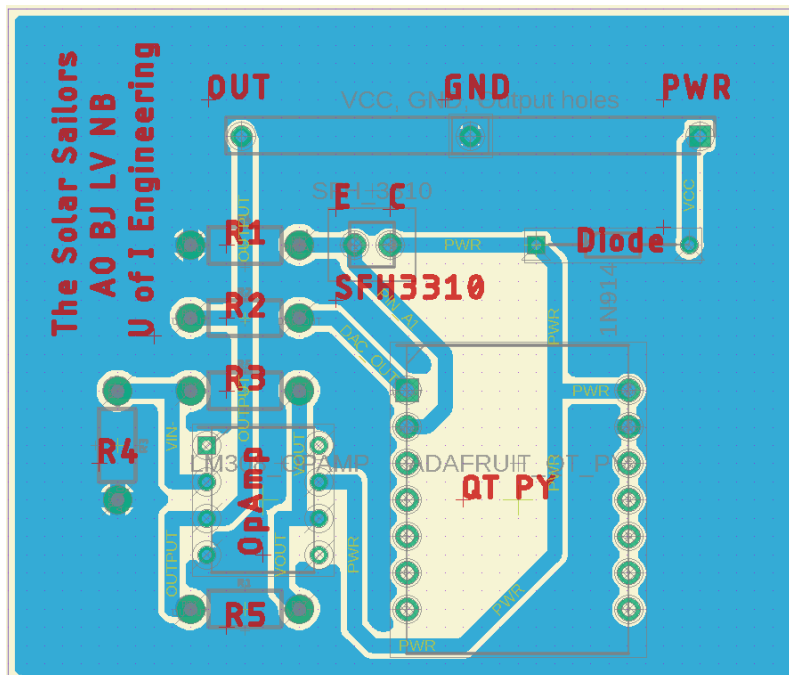
**Figure 6:** PV Matey Calibration Circuit

### Final Circuit Assembly

The final circuit schematic is shown in Figure 5 and implemented on a PCB. The files for the PCB can be downloaded from the Wiki page for this product, which is found in the Appendix. They can be printed through any company that prints PCBs. Figures 7 and 8 show the circuit schematic and PCB layout, which are designed using Eagle software.



**Figure 7:** Circuit Schematic



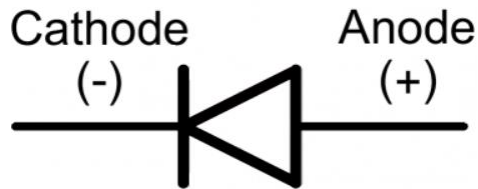
**Figure 8:** PCB Layout

Figure 8 shows the location for each component on the PCB. The locations of each component are label in red lettering on the top of the PCB while the



connections between components are on the bottom of the PCB. To assemble the circuit on the PCB, follow the instructions listed below.

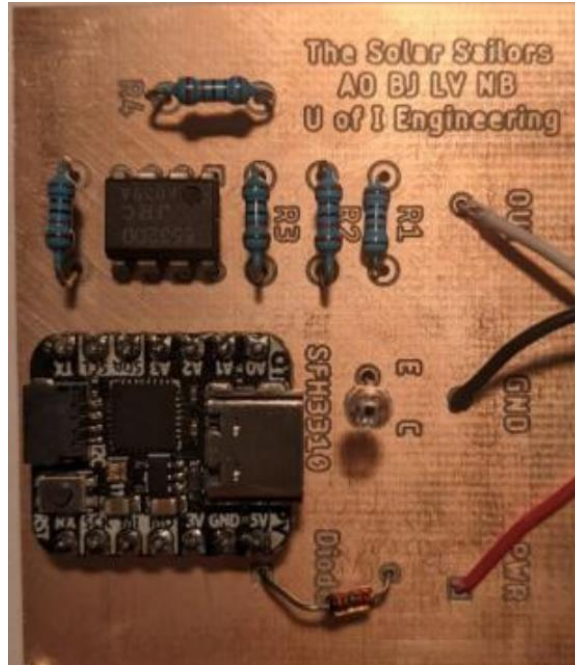
1. Place the legs of the diode in the holes labeled "Diode". Place the cathode in the left hole and the anode in the right hole. Figure 9 shows the diagram of a diode.



**Figure 9:** Diagram of a Diode

2. Place the SFH 3310 in the holes labeled "SFH3310". Place the collector (long leg) in the hole labeled "C" and the emitter in the hole labeled "E".
3. Place all resistors in their specific holes.
  - a. Place the 680 $\Omega$  resistor in the holes labeled "R1".
  - b. Place four, 36 $\Omega$  resistors in the holes labeled "R2", "R3", "R4", and "R5".
4. Solder the headers that came with the QT Py to the microcontroller. Then place the microcontroller with the soldered headers in the holes labeled "QT Py". Orient the QT Py with the USB C port facing the phototransistor.
5. Place the Op Amp in the holes labeled "OpAmp". Orient the Op Amp with the circle in the upper corner towards the label "R4".
6. Use a wire-strippers to strip the outer insulation from the three-conductor cable. Then strip the insulation from the red, black, and white wires.
  - a. Place the red wire in the hole labeled "PWR".
  - b. Place the black wire in the hole labeled "GND".
  - c. Place the white wire in the hole labeled "OUT".
7. Solder all components to the bottom of the PCB.
  - a. See the Appendix for instructions for best soldering techniques. More instructions can be found on the internet, if needed.
8. Repeat the process in step 6 at the other end of the three-conductor wire.
  - a. Place the red wire in the positive hole of the screw terminal power source.
  - b. Place the black wire in the negative hole of the screw terminal power source.
  - c. Place the white wire in the data logger or connect it to a multimeter to measure the current output from the PV Matey.

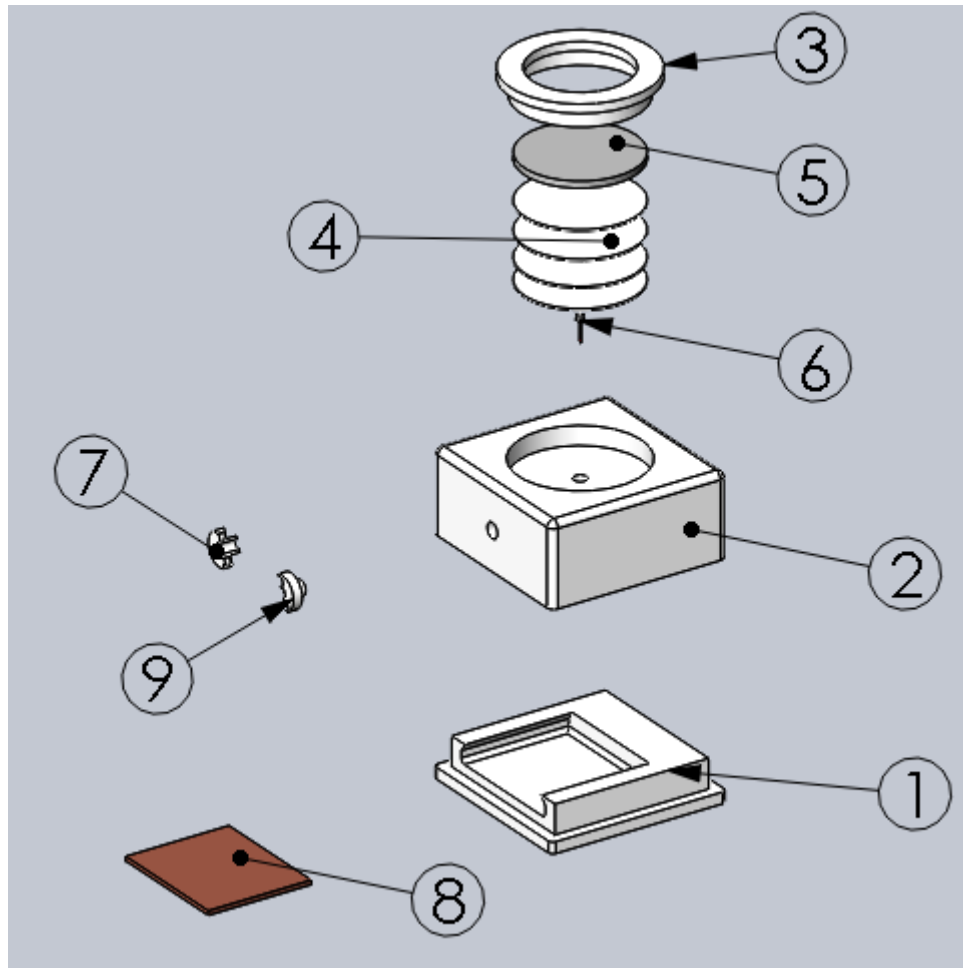
Figure 10 shows the completed circuit soldered to the PCB.



**Figure 10:** Components attached to the PCB

## Case

This case design is made using PETG filament and is impact and water resistant. It holds all the components of the prototype in a small box that can then be tethered to the side of a solar panel. The sensor goes into the center of the lid, with enough room to fully capture the range of the sensor, which is then covered by four layers of Teflon and an acrylic disk. A sensor cap ring seals the disk in place and prevent water from entering. The circuit and microcontroller are held in place in a breadboard on the bottom of the case. Figure 11 shows a complete view of the case assembly.



**Figure 11:** Complete View of Case Assembly

### Case Assembly

The files can be found by following the Wiki link found in the Appendix.

1. Print the provided files for case top (2) and case bottom (1) at 0.3mm layer height at a speed of 45mm/s and 3 wall shells. For optimal print conditions for the case bottom (1) print with the bottom face on the build plate. For optimal print setting for the case top (2) print the face with the sensor hole connected to the build plate and with conical supports activated.
2. Print the provided files for the sensor cap (3), female wire collar (7), and male wire collar (9) with 0.2 mm layer height at a speed of 45mm/s. For optimal print setting ornate each part with the large face orientated on the build plate using this orientation supports are not necessary.
3. Take the 2" acrylic disk and wrap one layer of electrical tape around the edge of the disk then trim it so that only ~1mm of the tape overlaps the top and bottom of the disk.
4. Slot the acrylic disk into the sensor cap (3) this will be a tight fit and will require some force to get it to fit. To make this process easier place the acrylic disk into a freezer for an hour to temporarily shrink the disk.

5. Wrap two layers of electrical tape around the outside bottom surface of the sensor cap (3).
6. Place 4 Teflon sheets (4) into the sensor cap (3) under the acrylic disk (5) then slot the entire assembly into the case top (2) in the sensor hole. This will require some force it will be a very snug fit.
7. Wrap the small wall of the case bottom (1) with two layers of electrical tape to make the application easier cut the tape at each of the four corners and make a cut at each end of the slot.
8. Place the completed PCB (8) into the case bottom slot and thread the wire through the cable hole.
9. Wrap the data cable with one layer of electrical tape on the section where the cable leaves the case. Make sure there is enough slack wire in the case to safely open the case without damaging any components.
10. Aline the sensor (6) with the sensor hole.
11. Press fit the case top (2) and the case bottom (1) together. Try to slot these together as close to level as possible to mitigate the possibility of bunching/tearing the electrical tape.
12. Clip together the female wire collar (7) and male wire collar (9) around wire segment with the electrical tape around it.
13. Using adhesive fuse part (7) and (9) together.
14. Using the same adhesive as before fuse the wire collar to the cable hole.
  - a. Due to the fixed nature of the case when any operation to open it is preformed make sure that the electrical tape in the impacted regions is replaced to ensure the case continues to be water resistant.

## Using Arduino IDE

### Connecting the Adafruit QT Py to Arduino IDE

Follow Link 1 found in the Appendix for details the steps to connect you Adafruit QT Py to Arduino IDE. The steps are also abbreviated below.

\*Before Connecting the QT Py to your computer\*

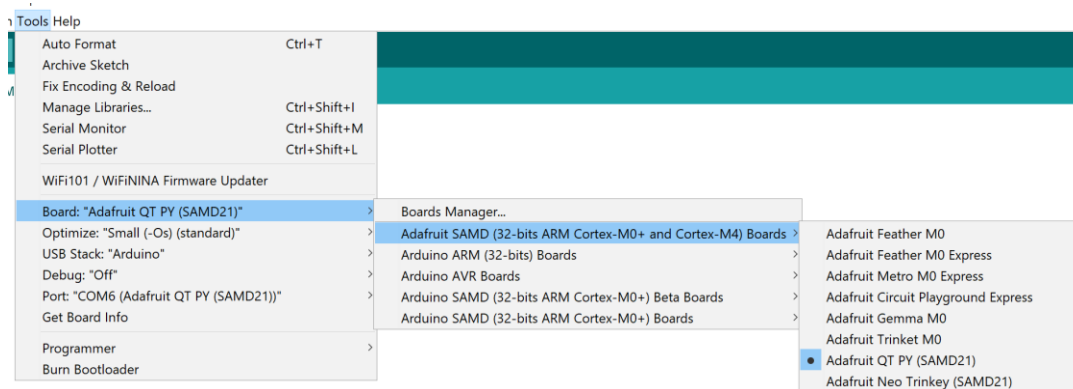
1. Download the latest version of Arduino IDE.
2. Go to **File** menu and click **Preferences**.
  - a. A dialog box will pop up.

The QT Py URL must be added to the **Additional Boards Manager URLs** box. The URLs for additional boards can be found at Link 2 in the Appendix. The QT Py is found using Link 3 in the Appendix.

- b. Paste the QT Py link into the **Additional Boards Manager URLs** box. Click **OK**.

Next, the board will be installed with the Board Manager. Follow Link 1 and scroll to the bottom of the page and click the **Using with Arduino IDE** box.

3. Connect the QT Py to your computer using a USB C cable. The RGB light on the QT Py will start changing colors.
4. Go to the **Tools** menu and click **Board**.
  - a. There may be a board listed after **Board:**, but it can be ignored.
  - b. Click **Board Manager**. A popup window will open with a list of boards. Next *Arduino* support for the QT Py will be installed.
    - i. Type "Arduino SAMD" into the **Filter your search** box at the top of the window.
    - ii. Click the **Arduino SAMD Boards (32-bits ARM Cortex M0+) by Arduino** box and install.
    - iii. Wait until this finishes installing before moving on.
    - iv. Another pop-up window may appear asking you to allow the app to make changes to your computer. Click **Yes**.
  - c. Next *Adafruit* Support for the QT Py will be installed.
    - i. Type "Adafruit SAMD" into the **Filter your search** box at the top of the window.
    - ii. Click the **Adafruit SAMD Boards by Adafruit** box and install.
    - iii. Wait until this finishes installing before moving on.
    - iv. Another pop up window may appear asking you to allow the app to make changes to your computer. Click **Yes**.
    - v. Once this finishes installing, close the popup window.
5. Quit and restart Arduino IDE. This ensures that the support for the QT Py is properly installed.
  - a. Go to the **Tools** menu and click **Board**. Go to **Adafruit SAMD (32-bit ARM Cortex-M0+ and Cortex-M4) Boards**. Scroll down to **Adafruit QT PY (SAMD21)** and make sure it is selected. Figure 12 shows a screen clipping of step 6.a.



**Figure 12:** Step 6.a

\*Note: If you are running Windows 7 or Windows 8, you must install the latest Adafruit Drivers. If you are running anything newer than Windows 8, your QT Py is connected! Follow Link 4 in the Appendix and scroll down to **Install Drivers (Windows 7 and Windows 8 Only)** and follow the steps to install the latest Adafruit Drivers.

## Code

The code is used to scale the input from the ADC and output a voltage to through the DAC, using Arduino IDE as a calibration test for the code. Figure 13 shows the microcontroller code.

```
float V=0;           //Variable for the recorded voltage
float V_out=0;
float B=0;          //Variable for the Bias can be wither positive or negative
float M=0;          //Variable for the slope of voltage vs irradiance.
float Factor=1;     //Flat multiplication factor used to voltage to irradiance conversion
float Irr=0;        //Variable for the irradiance
float Conversion=4.8876; //Variable for the conversion of a 10 bit value to a 5000mv signal (5000/(2^Bit-1))

void setup() {
  Serial.begin(9600); //Sets the serial monitor
}
void loop() {
  delay(500);        //pauses the code for 0.5sec

  V = Conversion*analogRead(A1);
  Irr=(V*M+B) *Factor;
  V_out=Irr*1023/2000;

  //Outputs the voltage read from the phototransistor to the V to C circuit
  analogWrite(A0,V_out);;

  //Debugging Code comment out if not being used
  //Serial.print(Irr); //W/m^2

  // calibration code
  //Serial.print(V); //mv
}
```

**Figure 13:** Microcontroller Code

## Code Setup

1. Input the values derived from calibration for B (Bias), M (Slope) and Factor.
2. Set the delay function to the desired data collection rate.
3. Upload the code to the QT-PY.
4. Check the serial monitor to verify that the code is functioning properly.
5. Take a multimeter to Pin A0 to verify that it is outputting the expected voltage values.
  - a. **Note:** While being powered by the 5V pin the QT-Py will deactivate the USB making it impossible to access the serial monitor or upload code so ensure the QT-Py is only receiving power from the USB during the code setup (See "Calibration Circuit Assembly").
  - b. **Note:** Be sure to calculate calibration values before powering the QT-Py via the 5V pin.

## Calibration

Calibration can **only** occur on a calibration circuit.

(see "**Calibration Circuit Assembly**").

The following equation is a linear calibration curve that can be found in **line X** of the code.

$$\begin{aligned} \text{Calibrated output}(W/m^2) &= (M * PV \text{ Matey}(mv) - B) * F \\ M &= \text{Slope} \\ B &= Y \text{ intercept} \\ F &= \text{Variance Factor} \end{aligned}$$

The user can be improved the PV Matey's accuracy by adjusting the slope, y intercept, and variance factor (see "**Code Setup**").

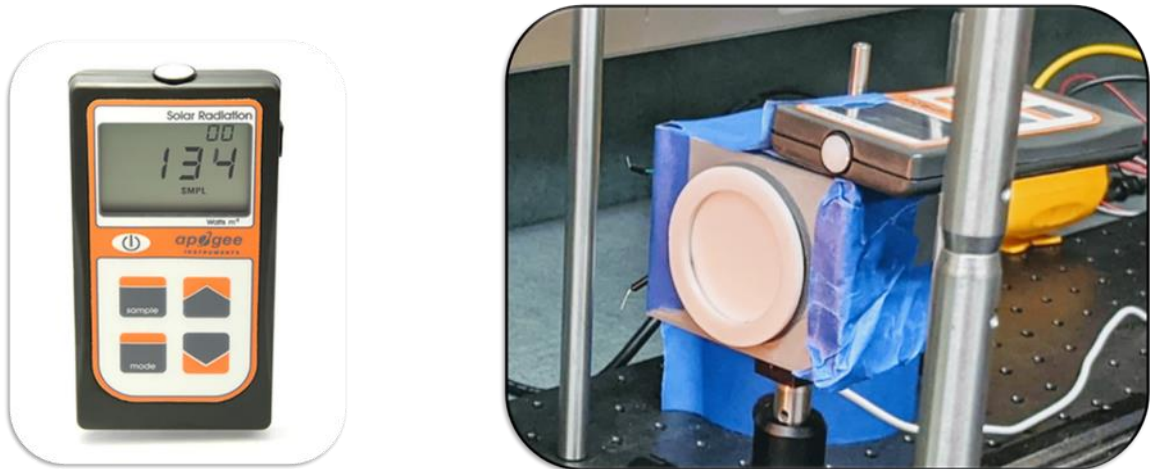
Below are instructions on how to create a calibration curve from scratch.

1. Set up an adjustable light source. We used the Fiber Lite DC-950



**Figure 14:** Fiber Lite DC-950

2. Closely mount the PV Matey and a calibrated pyranometer. Our original calibration involved using the MP-100: Pyranometer Integral Sensor. Both sensors should be equidistant from the light source. Figures 15(a) and 15(b) illustrate this step.

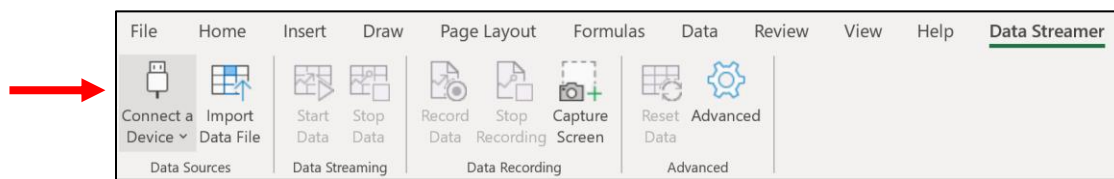


**Figure 15:** (A) The MP-100:Pyranometer Integral Sensor (B) Mounted sensors

3. Connect the QT-Py to a computer using a USBC to USB cable
4. Incrementally increase the intensity of the light source and record the analog output of the PV Matey (mV) and the calibrated pyranometer ( $W/m^2$ ).

Data from the MP-100 was recorded by hand into Excel. Depending on the model of pyranometer available at time of calibration, a data logger may be used.

- a. Record PV Matey data with Data Streamer:  
Data Streamer in Excel is a simple way to record data from a microcontroller. If data streamer can be successfully connected to the QT-Py, you can skip to step (5).



**Figure 16:** Connect a Device Option in the Data Streamer Tab of Excel

- b. Record PV Matey data from Arduino IDE:  
In Arduino IDE make sure that the QT-Py board is selected (See **"Using Arduino IDE"** section).

Open the serial monitor on the top right of the application. Copy and paste data from the COM window into Excel.





Figure X: Serial monitor option in Arduino IDE

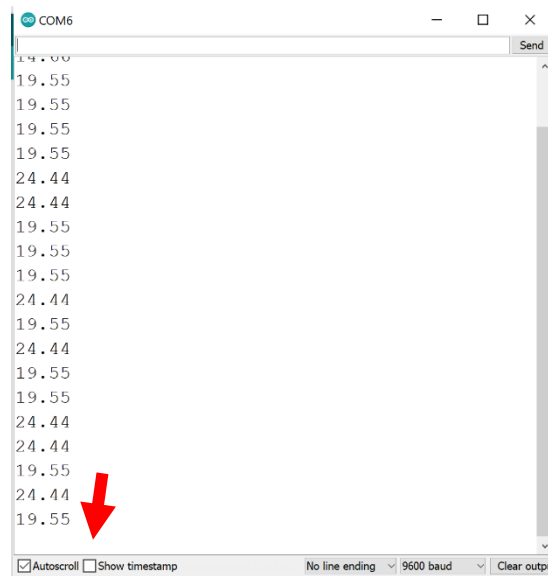
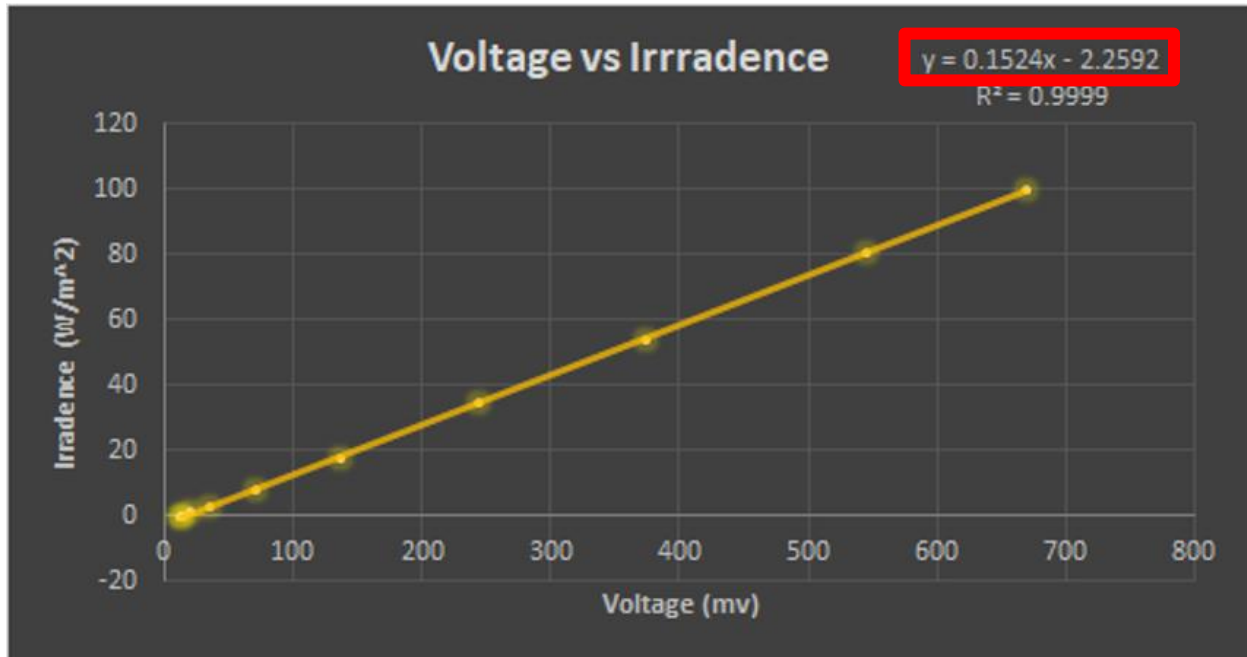


Figure 17: COM Window

- i. To pause readings, disconnect the USB from your computer.
- ii. Turn on time stamps temporarily at the beginning and end of light adjustments if needed. This can be used to accurately compare time stamped data from a commercial pyranometer.

Timestamps need to be manually removed in Excel, so it is not recommended to leave time stamps on for long periods of time.

5. Create a graph using collected data with the PV Matey (mV) on the x-axis and the calibrated pyranometer data ( $W/m^2$ ) on the y-axis. Add a linear trendline to create the calibration curve equation.



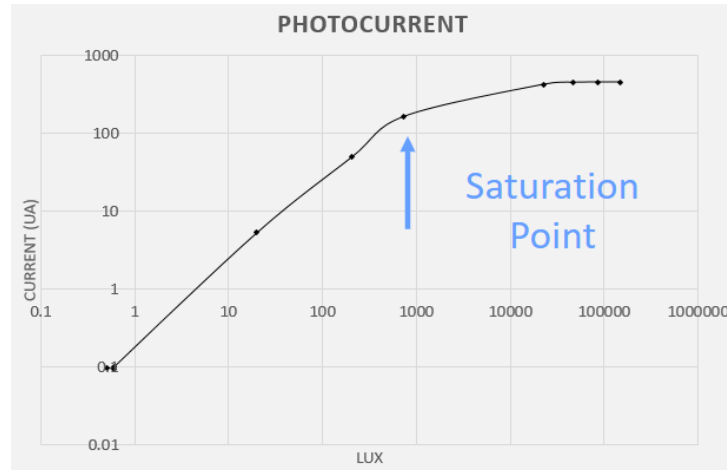
**Figure 18:** Calibration curve

6. Upload new code with values for (M) and (B) from step (5). The variance factor (F) should equal one.
7. If there is a wide variation between the PV Matey and the commercial pyranometer's data, the user can tune the PV Matey's accuracy by adjusting the variance factor (F) in the code.
  - a. Compare the average output of the PV Matey for a given time with the average output of a calibrated pyranometer in the desired mounting location.
  - b. Divide these averages and adjust the variance factor (F) by this ratio.

## Results and Tables

### Photocurrent Testing

To validate the SFH 3310 sensor, an adjustable light source and lux meter were used to create a photocurrent to lux graph. This test expanded past the data available in the datasheet. The graph below indicates that at 1000 lux the sensor hits a "saturated" region. Data past this point is no longer linear and losses accuracy. To avoid saturation past this point, an attenuator is needed.



**Figure 19:** Photocurrent from the SFH-3310

### Filtering

To evenly attenuate the light down below 1000 lux, layers of Teflon were used. Using a light source and lux meter in the test below, we found that each sheet of Teflon attenuates 73.8% of light and that four sheets are required to prevent saturation under the brightest natural circumstances.

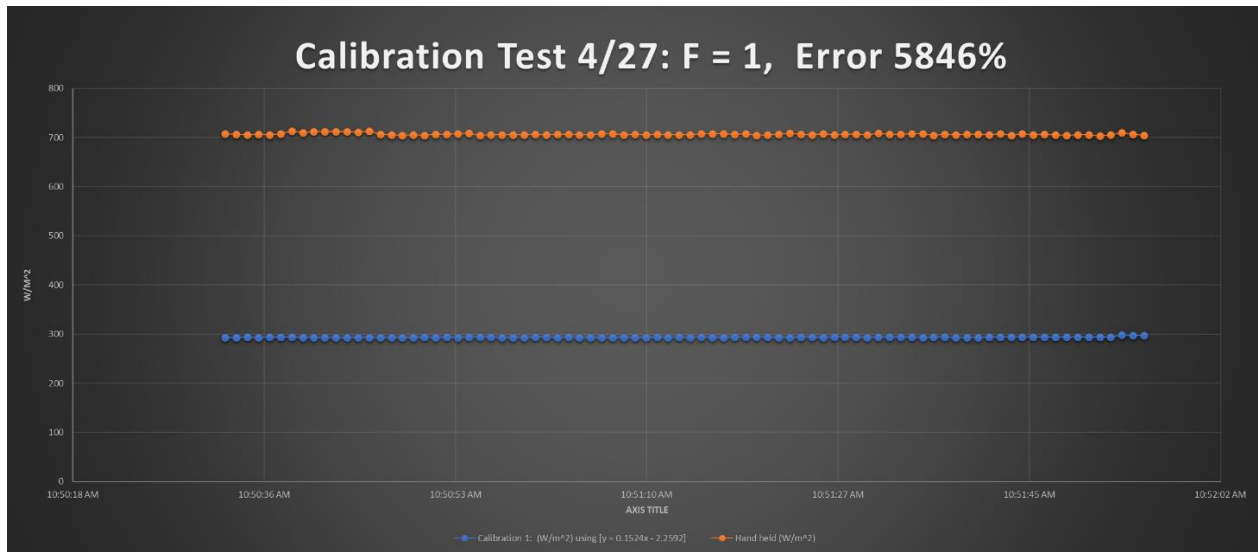


**Figure 20:** Teflon Filter Testing

### Calibration Results

The graph below shows a disparity between the data from the calibration curve (W/m<sup>2</sup>) and the output from a commercial pyranometer (W/m<sup>2</sup>)

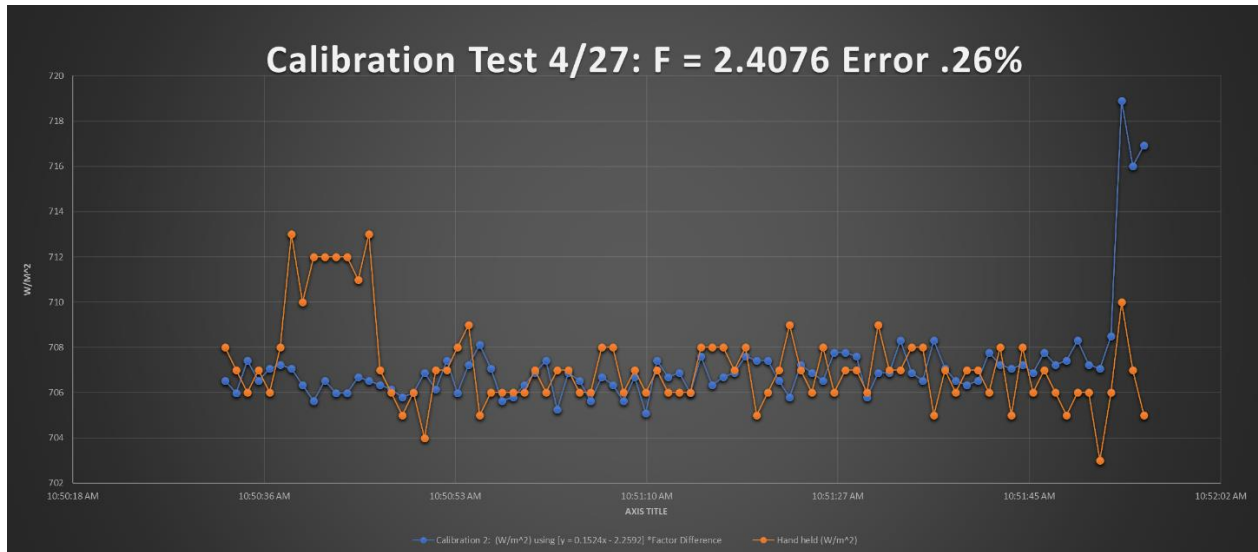
$$y = 0.1524x - 2.2592$$



**Figure 21:** Calibration testing without a Variance Factor

To improve accuracy, a variance factor was added.

$$F = \frac{\text{Commercial Pyranometer} \left( \frac{W}{m^2} \right)}{\text{PV Matey (mV)}} = 2.4076$$



**Figure 22:** Calibration Testing with a Variance Factor

The graph above shows calibration with a variance factor. It shows that the error is 0.26% between values with a factor of 2.4076 between the PV Matey and the pyranometer that we had access to.

## Appendix

### Component Links

680Ω Resistor (1): <https://www.mouser.com/ProductDetail/Vishay-Dale/CCF07680RGKE36?qs=sGAEpiMZZMtG0KNrPCHnjXeT93wkI59768WqROCKsvM%3D>

Adafruit QT Py (1): <https://www.adafruit.com/product/4600>

3D printing PETG filament (1): [https://smile.amazon.com/DURAMIC-3D-Printing-FilamentDimensional/dp/B07TRPQ4MN/ref=sr\\_1\\_2\\_sspa?crid=1FH4OZWBN70YJ&dc\\_hild=1&keywords=petg+3d+printer+filament&qid=1613577099&srefix=PETG+3d%2Caps%2C226&sr=8-2-spons&pvc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzME9aOFJGVjZLUesmZW5jcnlwdGVkSWQ9QTA0NTUyODQ3NkrRDMzhUT1VDRyZlbnNyeXB0ZWRBZEIkPUeWnjkWMDkzMTdYQkZGQjgzMlVYTzZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=](https://smile.amazon.com/DURAMIC-3D-Printing-FilamentDimensional/dp/B07TRPQ4MN/ref=sr_1_2_sspa?crid=1FH4OZWBN70YJ&dc_hild=1&keywords=petg+3d+printer+filament&qid=1613577099&srefix=PETG+3d%2Caps%2C226&sr=8-2-spons&pvc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzME9aOFJGVjZLUesmZW5jcnlwdGVkSWQ9QTA0NTUyODQ3NkrRDMzhUT1VDRyZlbnNyeXB0ZWRBZEIkPUeWnjkWMDkzMTdYQkZGQjgzMlVYTzZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=)

Phototransistor SFH 3310 (1): <https://www.mouser.com/ProductDetail/720SFH3310>

\*Note: This phototransistor can be changed out for another phototransistor that has the same pinout and footprint. Be sure to check the datasheet for the SFH 3310 phototransistor when looking for a replacement phototransistor.

2" Acrylic Disks (1): <https://www.usplastic.com/catalog/item.aspx?itemid=40805&catid=441>

12"x12" Teflon Sheet (1): <https://www.eplastics.com/PTFENAT0-040X12X12>

36Ω Resistor (4): <https://www.mouser.com/ProductDetail/Vishay-Dale/CCF0736R0GKE36?qs=sGAEpiMZZMsPqMdJzcrNwnPpI93fAKplt%2Fje72JuHDK%3D>

Op Amp LM358P (1): <https://www.mouser.com/ProductDetail/Texas-Instruments/LM358P?qs=X1HXWtTiZ0QtOTT8%252BVnsyw%3D%3D>

\*Note: This Op Amp can be changed out for another Op Amp that has the same pinout and footprint. Be sure to check the datasheet for the Op Amp LM358P when looking for a replacement Op Amp.

22 AWG Insulated Copper Wire (29'): <https://www.superbrightleds.com/moreinfo/power-wires-cables/pvc-jacketed-3-conductor-22-awg-power-wire-pp-frpvc-gray/1850/4475/>

5V, 1A Power Source (1): [https://www.amazon.com/inShareplus-Voltage-Transformer-100-240V-Connector/dp/B07GL9YB3Z/ref=sr\\_1\\_1\\_sspa?dchild=1&keywords=5v%2Bpower%2Bsupply&qid=1618527091&sr=8-1-spons&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyNE9ONjFDN004SDlYjMvUyY3J5cHR](https://www.amazon.com/inShareplus-Voltage-Transformer-100-240V-Connector/dp/B07GL9YB3Z/ref=sr_1_1_sspa?dchild=1&keywords=5v%2Bpower%2Bsupply&qid=1618527091&sr=8-1-spons&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyNE9ONjFDN004SDlYjMvUyY3J5cHR)

[IZEIkPUEwODEwOTAzVUVGTIRHOEVWWFFHJmVuY3J5cHRIZEFkSWQ9QTA2OTAzMzk  
yRk81S084QIU0MkxaJndpZGdlE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JIZGlyZWN0J  
mRvTm90TG9nQ2xpY2s9dHJ1ZQ&th=1](https://www.mouser.com/ProductDetail/512-1N914T50A)

1N914 Diode (1): <https://www.mouser.com/ProductDetail/512-1N914T50A>

### Connecting to Arduino IDE

Link 1: <https://learn.adafruit.com/adafruit-qt-py/arduino-ide-setup>

Link 2: <https://github.com/arduino/Arduino/wiki/Unofficial-list-of-3rd-party-boards-support-urls#list-of-3rd-party-boards-support-urls>

Link 3: [https://adafruit.github.io/arduino-board-index/package\\_adafruit\\_index.json](https://adafruit.github.io/arduino-board-index/package_adafruit_index.json)

Link 4: <https://learn.adafruit.com/adafruit-qt-py/using-with-arduino-ide>

### Links to More Information

Official Adafruit QT Py Website: <https://learn.adafruit.com/adafruit-qt-py>

Voltage-To-Current Converter (Howland Current Pump):

<https://www.allaboutcircuits.com/technical-articles/the-howland-current-pump/>

### Soldering

Best soldering technique: <https://www.youtube.com/watch?v=AqvHogekDI4>

### Wiki Page Link

[http://mindworks.shoutwiki.com/wiki/Development\\_of\\_a\\_Low\\_Cost\\_Pyranometer](http://mindworks.shoutwiki.com/wiki/Development_of_a_Low_Cost_Pyranometer)

### Code

```
float V=0;           //Variable for the recorded voltage
float V_out=0;
float B=0;           //Variable for the Bias can be wither positive or negative
float M=0;           //Variable for the slope of voltage vs irradiance.
float Factor=1;      //Flat multiplication factor used to voltage to irradiance conversion
float Irr=0;         //Variable for the irradiance
float Conversion=4.8876; //Variable for the conversion of a 10 bit value to a 5000mv signal (5000/(2^Bit-1))

void setup() {
  Serial.begin(9600); //Sets the serial monitor
}

void loop() {
  delay(500);        //pauses the code for 0.5sec

  V = Conversion*analogRead(A1);
```

## 21

```
Irr=(V*M+B)*Factor;
```

```
V_out=Irr*1023/2000;
```

```
//Outputs the voltage read from the phototransistor to the V to C circuit
```

```
analogWrite(A0,V_out);;
```

```
//Debugging Code comment out if not being used
```

```
//Serial.print(Irr); //W/m^2
```

```
// calibration code
```

```
//Serial.print(V); //mv
```

```
}
```