TrophIO

Climate-smart Controlled Environment Agriculture

1.Introduction

In the face of climate change, traditional farming faces increased challenges, especially in disadvantaged communities. Controlled Environment Agriculture offers a solution by using technology to optimize crop yield in hydroponic systems. This project utilizes the Internet of Things (IoT) to monitor key environmental factors in a solar-powered, wireless system. It supports training and research, building climate resilience in New Mexico and surrounding tribal communities.

2. Hardware Overview

2.1 PARTICLE DEVICE

What is a microcontroller?

A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system.

Photon 2

The Photon 2 is a development module with a microcontroller with Wi-Fi, BLE, and significant memory for large applications.

Particle. Photon 2 Datasheet & Documentation. Available at: https://docs.particle.io

2.2 SENSORS

Adafruit AS7341 10-Channel Light / Color Sensor

multi-channel spectrometer

- Detects the amounts of light within different wavelengths, visible and near infrared.
- 16 different sensors detect 8 separate, overlapping bands of colored light.
- 16-bit 6-channel ADC (analog-to-digital converter) takes raw measurements and converts them to digital values that can be read over I2C.

BME 280 Temperature/Humidity/ Barometric Pressure

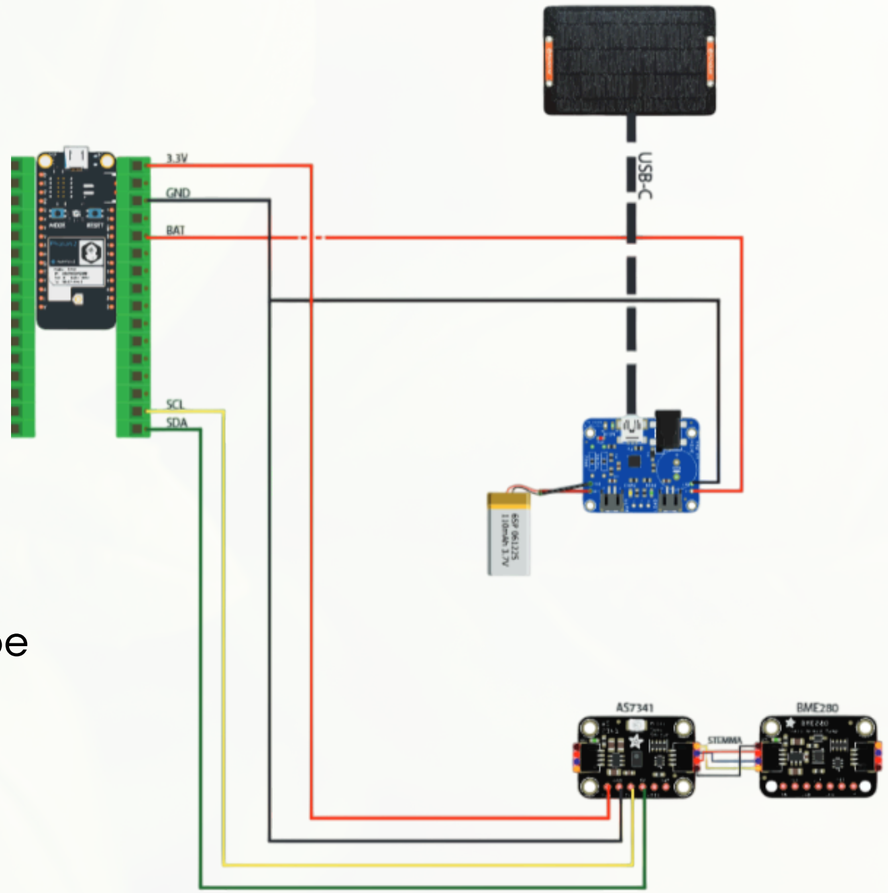
Environmental Sensor

- The BME280 utilizes Micro-Electro-Mechanical Systems (MEMS) technology to measure pressure, humidity, and temperature.
- Especially developed for applications where size and low power consumption are key design parameters.
- Measures humidity with ±3% accuracy, barometric pressure with ±1 hPa absolute accuraccy, and temperature with ±1.0°C accuracy.

Adafruit. BME280 Temperature, Humidity and Pressure Sensor – I2C/SPI Interface. Available at: https://www.adafruit.com/product/2652
Adafruit. AS7341 10-Channel Light / Color Sensor Breakout – STEMMA QT / Qwiic. Available at: https://www.adafruit.com/product/4698

2.3 WIRING PERIPHERAL DEVICES

- Components
- Junction Box
- Microcontroller
- 3.7V Lithium-ion Battery
- 5V 6W Solar Panel
- Battery Charge Controller
- Light Color Sensor (AS7341)
- Temp/Pres/Hum Sensor (BME280)



There are 4 wires soldered to the light color sensor (AS7341) that need to be wired to the microcontroller as follows:

VIN – Connect this pin to 3.3V on the microcontroller

GND – Connect this pin to ground on the microcontroller

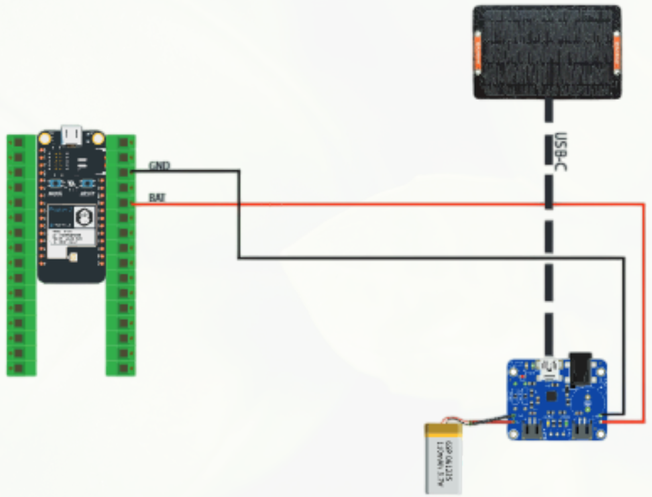
SCL – Connect this pin to SCL on the microcontroller

SDA – Connect this pin to SDA on the

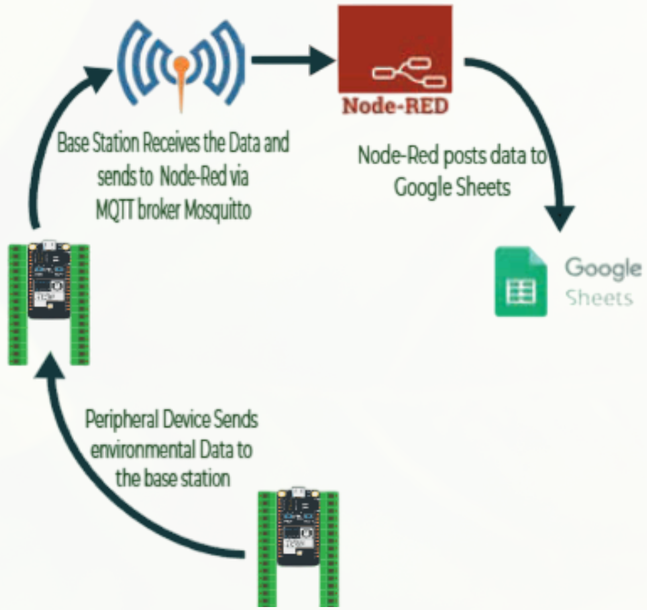
The BME280 connects to the AS7341 using a STEMMA QT connector.

2.4 WIRING BASE STATION

Wiring For Base Station



Data Flow



The base station is responsible for aggregating data from each peripheral device using BLE communication, then sending that data to an online dashboard (Node-Red) using MQTT, and ultimately storing those data points in the cloud (ex. Google Sheets).

3. Software Overview

3.1 Visual Studio Code (VSCode)

VSCode is a powerful, free, and open-source code editor developed by Microsoft. It supports a wide range of programming languages and extensions, making it ideal for embedded systems development.

3.1.1 Flashing Code to the Photon 2

1. Connect your Photon 2 via USB-C.
2. Open your .ino or .cpp file.
3. Compile the code locally using the command Particle: Compile application (local).
4. Flash the compiled code by selecting Particle: Flash application for connected device.

Visual Studio Code. Code Editor for Development. Available at: https://code.visualstudio.com

If you have a Mac M1/M2 (Apple Silicon)

Terminal needs to be “Open using Rosetta” option:

https://www.courier.com/blog/ tips-and-tricks-to-setup-your-apple-m1-for-development/

Getting Git and for Windows Users

Windows:

https://git-scm.com/download/win

Mac (from Terminal)

```
# install homebrew
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/
      install.sh)"

# might be necessary on M1/M2 Macs
eval $(/opt/homebrew/bin/brew shellenv)

# install git
brew install git
```

Linux (from Terminal)

```
sudo apt-get install git-all
```

Command Line Interface - Basic Navigation

The Command Line Interface (CLI) will allow us to directly navigate the computers operating system. We will use:

- macOS or Linux: Terminal
- Windows: PowerShell or Git Bash (we will use Git Bash)
-

The following commands will work on all three systems, except where noted below. macOS and Linux are case-sensitive, Windows is not.

- pwd: Show the present working directory.
- ls: To get the list of all the files or folders.
- cd: Used to change the directory.
- du: Show disk usage. (not available in PowerShell).
- man: Used to show the manual of any command.

Installing Particle CLI

Windows (After download, right click and Run as Administrator):
https://docs.particle.io/tutorials/developer-tools/cli/

Mac

```
# install the CLI
bash <( curl -sL https://particle.io/install-cli )

# install the DFU-util, a utility program for programming devices over USB
brew install dfu-util

# either or both of the following might be necessary on M1/M2 Macs
eval $(/opt/homebrew/bin/brew shellenv)   # if brew doesn't work
arch -arm64 brew install dfu-util         # if architecture error
```

Linux

```
# install the CLI
bash <( curl -sL https://particle.io/install-cli )

# install the DFU-util, a utility program for programming devices over USB
sudo apt-get install dfu-util
```

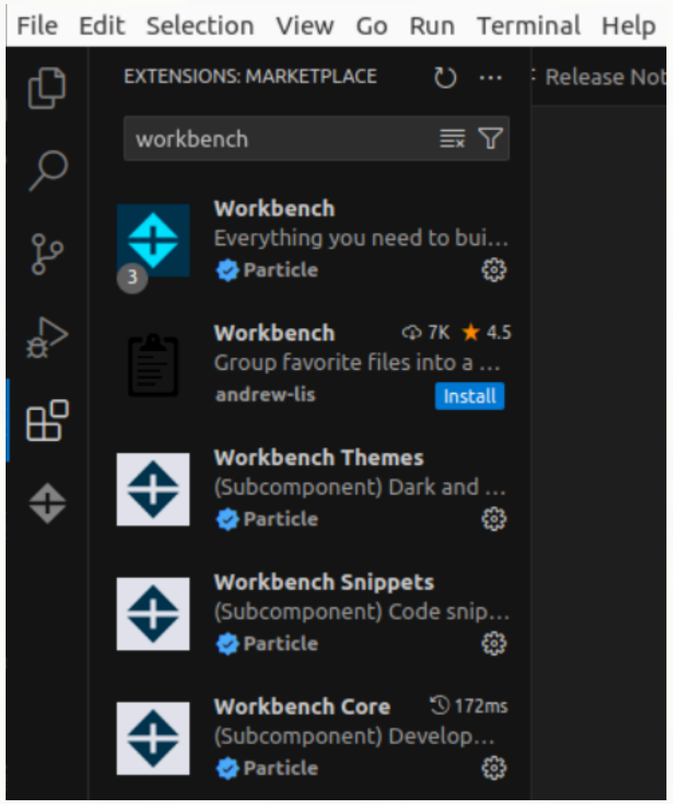
Test that the Particle CLI installed correctly by going to GitBash or Terminal and type particle.

Particle Software - Visual Studio Code

1. Create Particle login:
   https://login.particle.io/signup

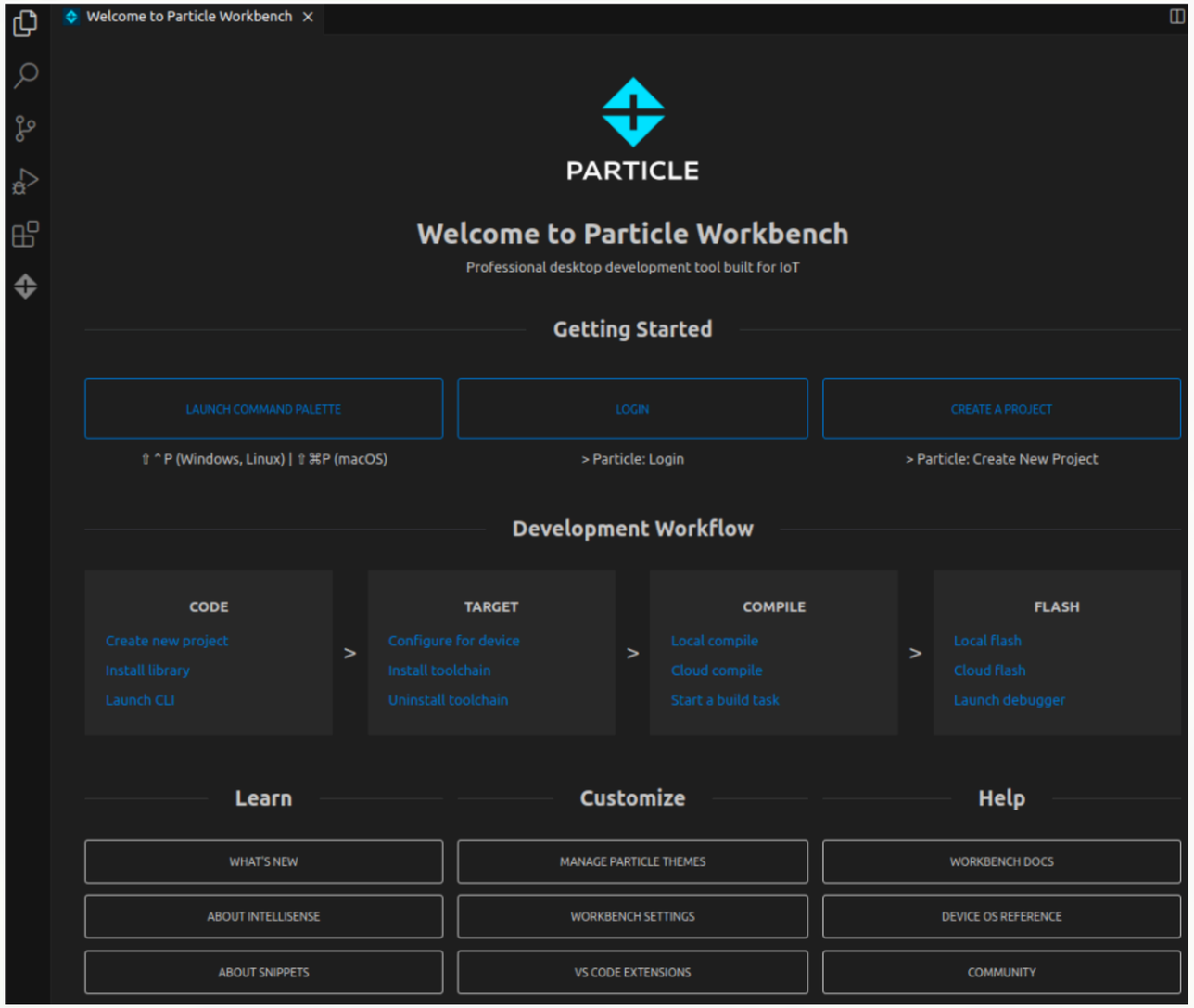
2. Install VSCode:
   https://code.visualstudio.com/download

3. Install the Particle Workbench extension

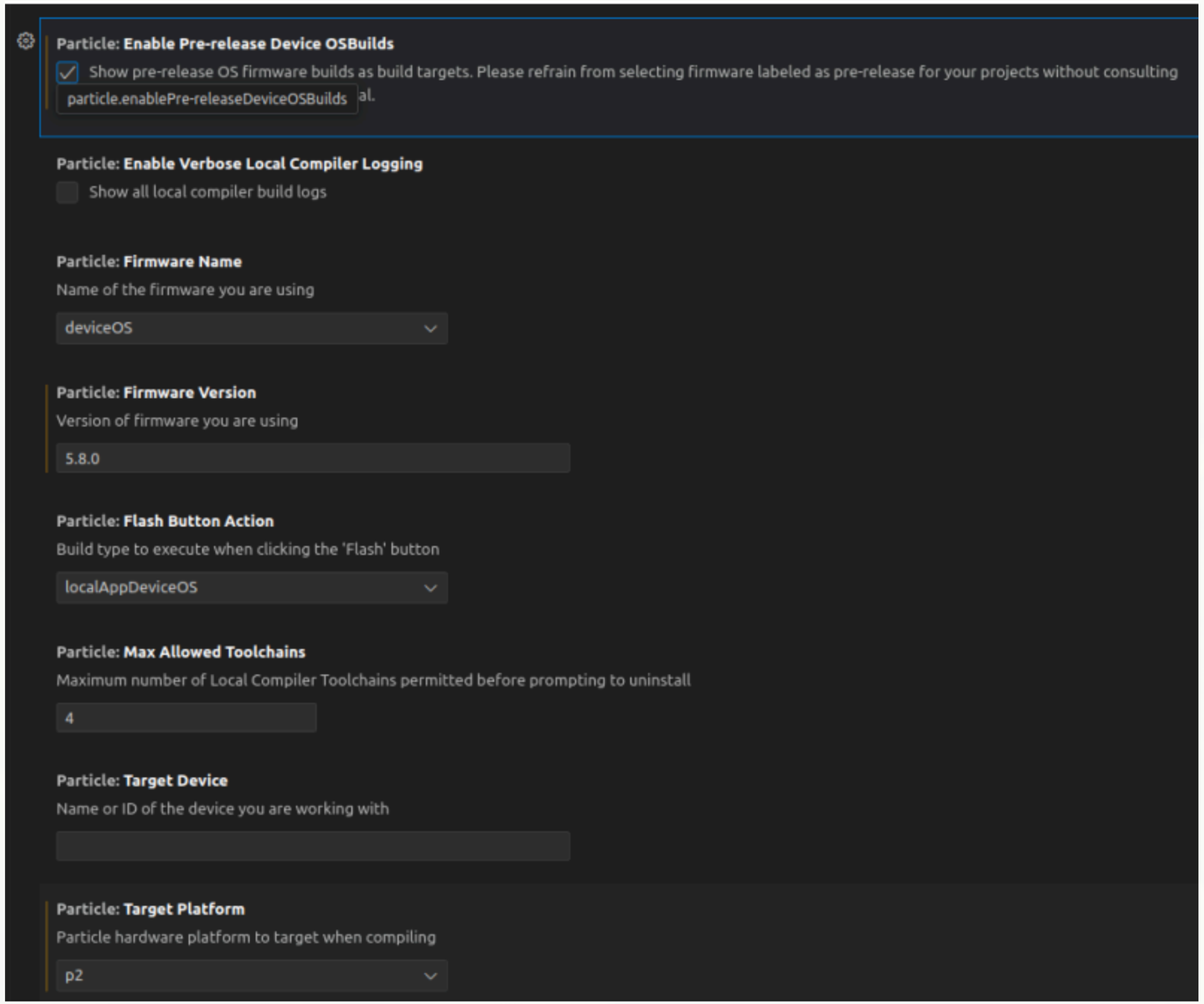


Particle Workbench - VSCode Configuration

Select customize→ Workbench Settings:



Particle Workbench - VSCode Configuration



- Enable Pre-Release
- Firmware Version to 5.9.0
- Target Platform to P2

Particle Setup

1. Plug the Photon2 into a USB port. It should begin blinking blue.
2. Open GitBash or Terminal.
3. Login into your Particle Account.

```
particle login
```

4. Ensure you have the latest Particle CLI.

```
particle update-cli
```

5. Setup WiFi using https://docs.particle.io/tools/developer-tools/configure-wi-fi/
6. Put the Photon2 in DFU mode (blinking yellow) by holding down MODE. Tap RESET and continue to hold down MODE. The status LED will blink magenta (red and blue at the same time), then yellow. Release when it is blinking yellow.

Claim Your Device

1. Claim the device to your account. This can only be done if it's breathing cyan. Replace e00fce681ffffffffc08949b with the device ID you got earlier from particle serial identify. Then, rename it to the name of your choice.blue.

```
particle device add e00fce681ffffffffc08949b
particle device rename e00fce681ffffffffc08949b myPhoton2
```

2. Ensure that your setup flag is marked as done.

```
particle usb setup-done
```

3. You have successfully set up your Photon2!

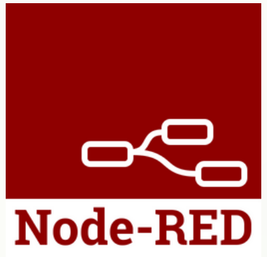
3.2 Node - RED

Node-RED is an open-source, browser-based flow editor that allows you to wire together data sources, services, and APIs. It uses a visual interface to simplify the logic and flow of IoT applications.

3.2.1 Why Node-RED

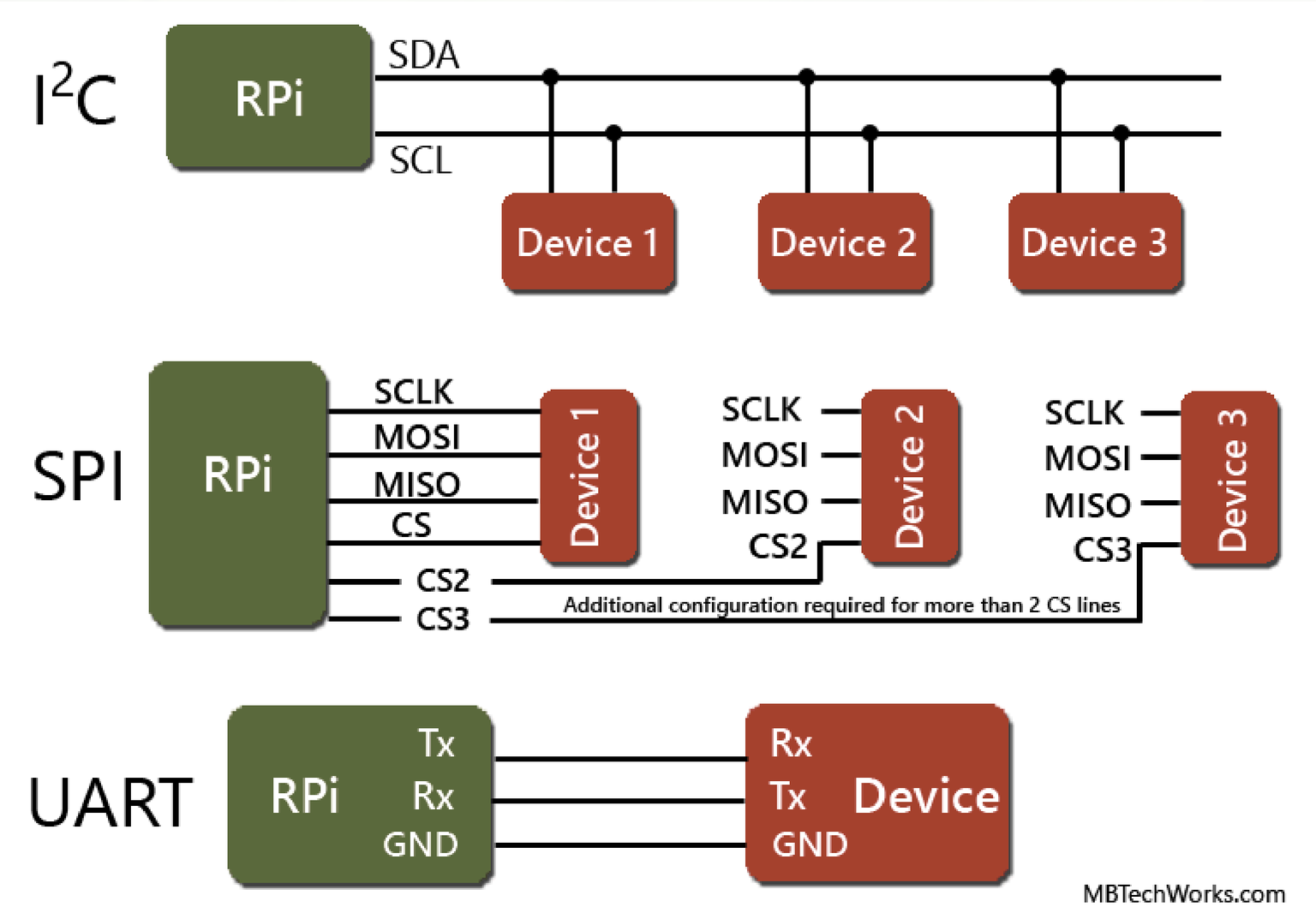
In our CEA monitoring system, Node-RED acts as the main data processing and visualization tool:

- It receives environmental data from the Photon 2 microcontrollers via MQTT.
- It allows us to transform, route, and display this data on a customizable dashboard.



Node-RED. Official Documentation. Available at: https://nodered.org/docs/

4. Communication Protocols



BLE

Bluetooth Low Energy

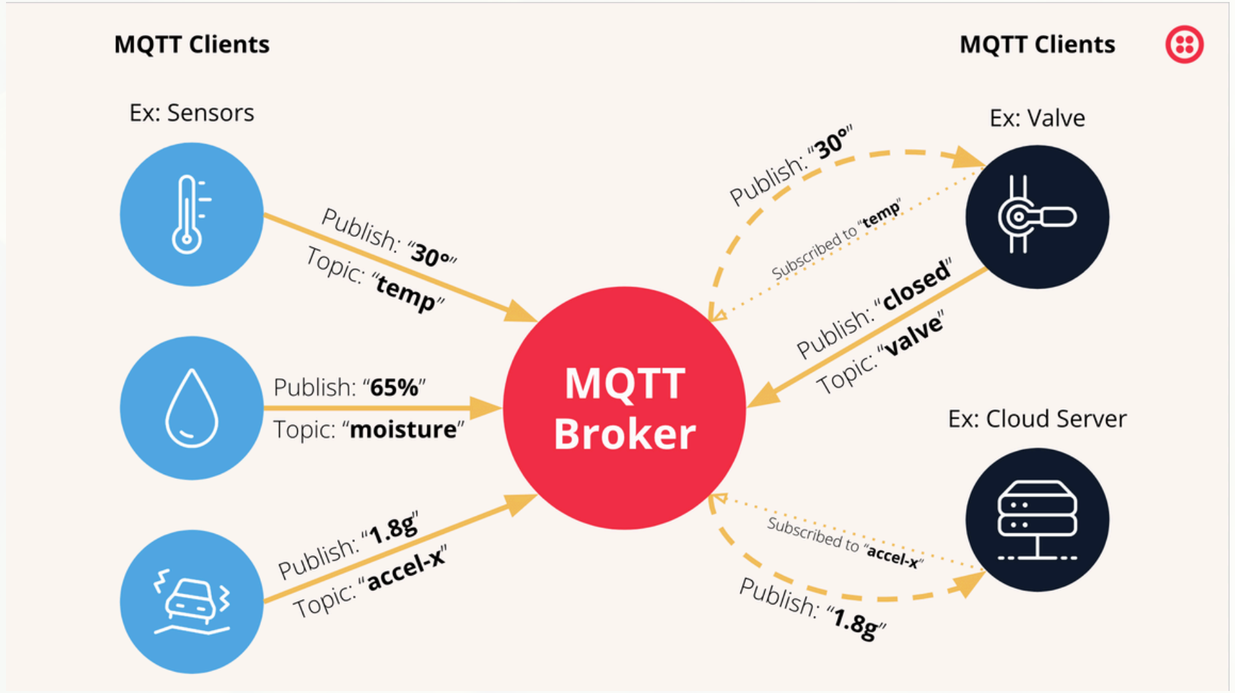
- Short-range wireless communication.
- Used for low-power sensor communication or device pairing.

Versions

- BLE 4.0: Low power, basic data rates
- BLE 5.0+: Longer range, higher speeds, better mesh support
- Great for debugging, mobile interfaces, or local data sync.

MQTT

Message Queuing Telemetry Transport



5. Solar Power Considerations

Using solar panels to power IoT devices offers numerous benefits, but careful consideration of several factors is crucial for successful implementation.

5.1 Solar Panel Power Ratings

We are using the FlexSolar 5V 6W Mini USB Solar Panel, a compact and portable panel designed for small electronics like IoT devices.

Key features:

- Output: 5V, up to 6W (1.2A max current)
- Type: High-efficiency monocrystalline cells
- Design: Durable, waterproof surface; compact for easy mounting.
- Ideal For: Low-power IoT applications, portable use

5.2 Understanding Battery Capacity

We are using a **3.7V Lithium-ion** battery, commonly found in compact electronics.

Why Li-ion?

- High energy density.
- Rechargable.
- Compact size and lightweight.
- Compatible with 5V solar input via proper charge controller.

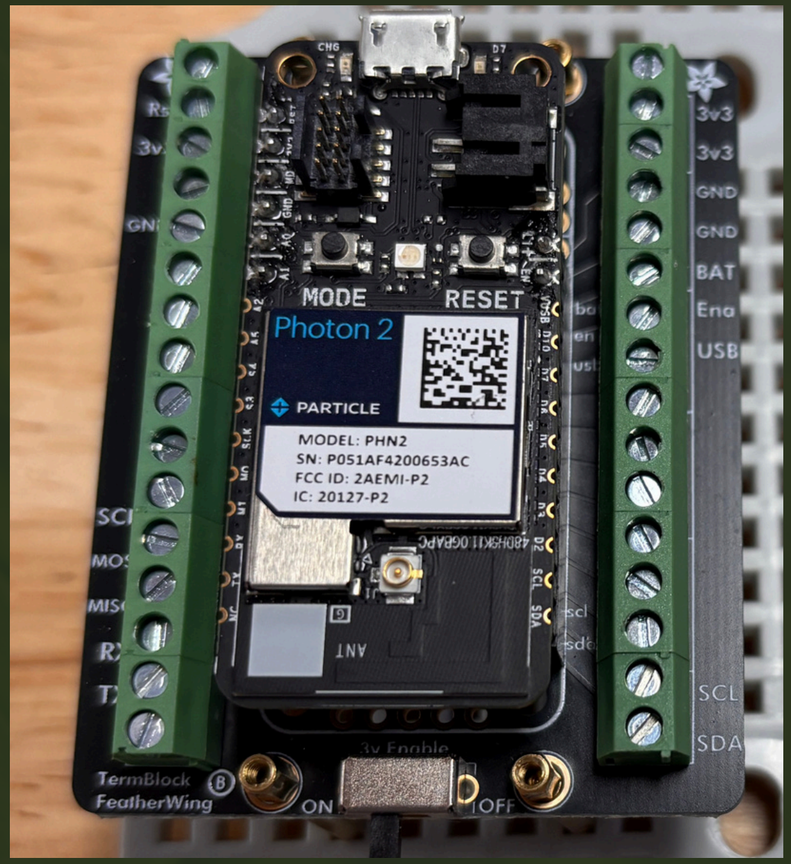
5.3 Power Consumption & Conservation

- Use sleep modes on the Photon 2 between sensor readings.
- Reduce sensor reading frequency as needed.

6. Assembly / Setup Instructions

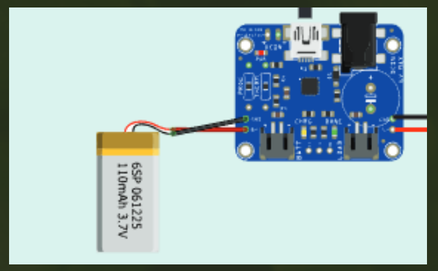
1. Prepare the Base

- Place the Photon 2 securely onto the terminal board.
- Align the pins and gently press down until seated properly.

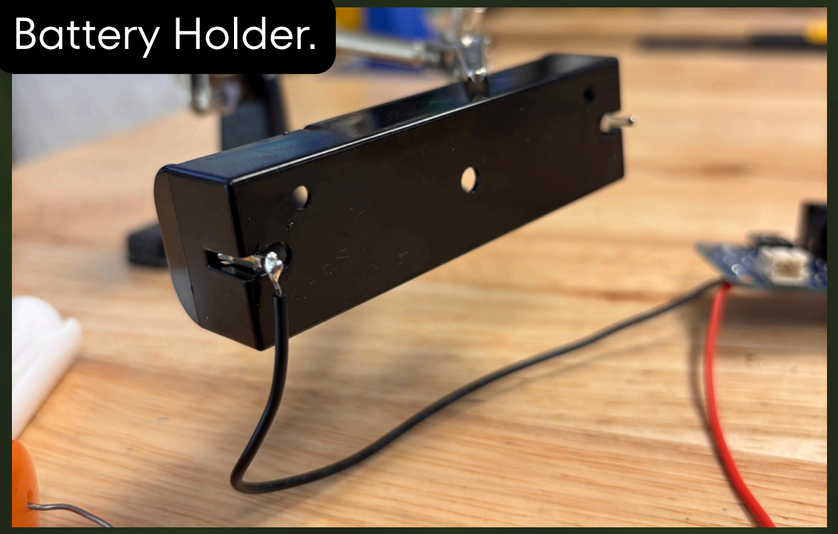


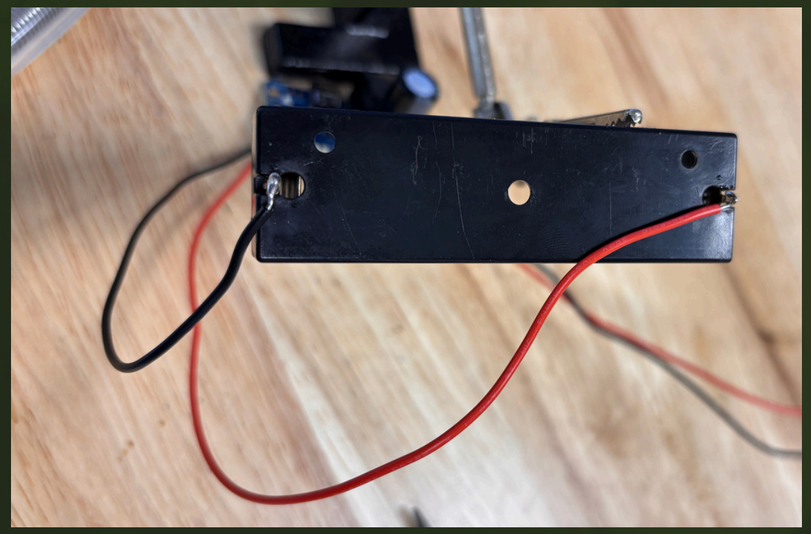
2. Wire the Power System

- Connect the battery holder to the charge controller.
- Solder or connect the battery wires (+/-) to the appropriate terminals on the charge controller.

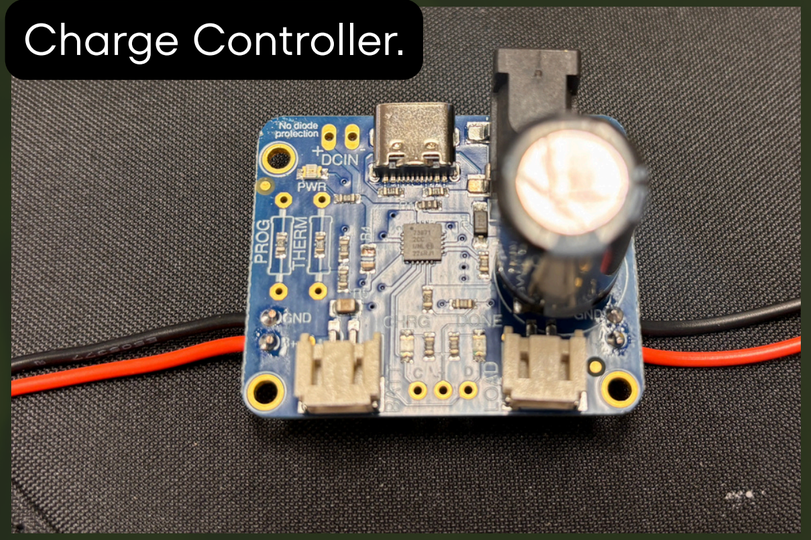


Battery Holder.



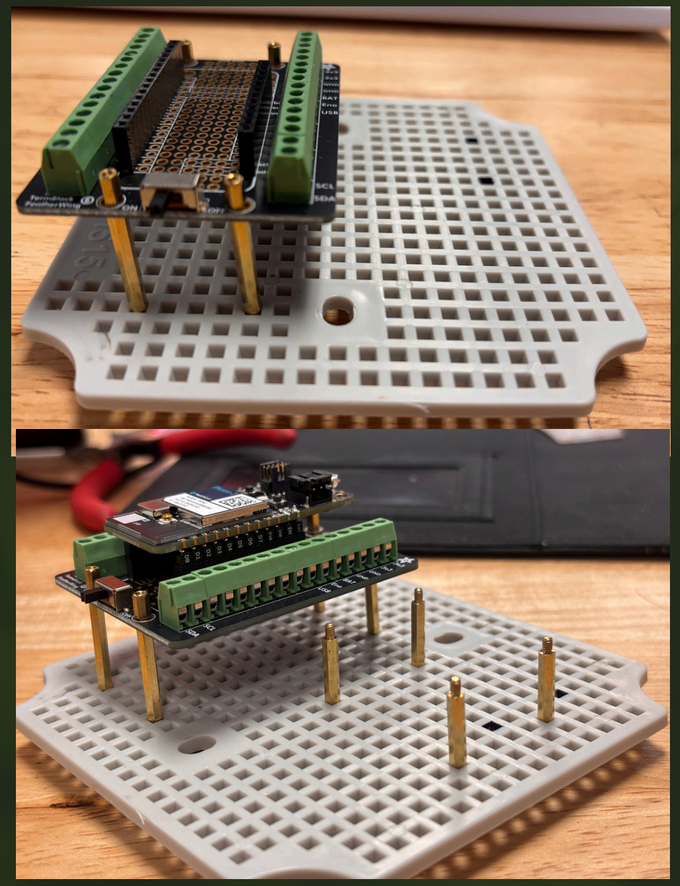


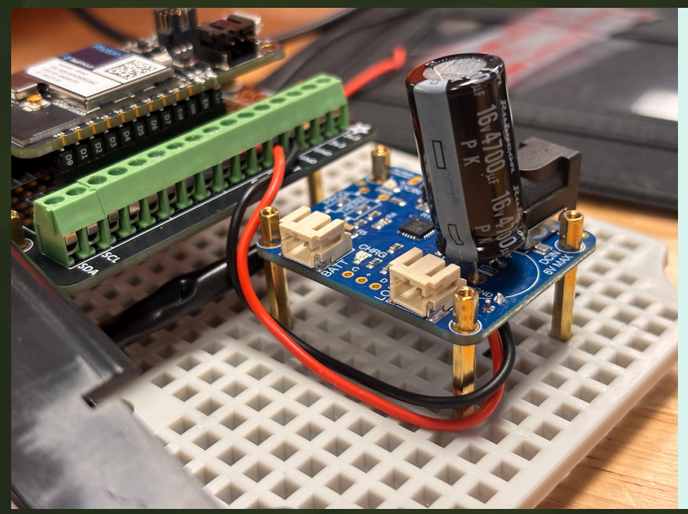
Charge Controller.

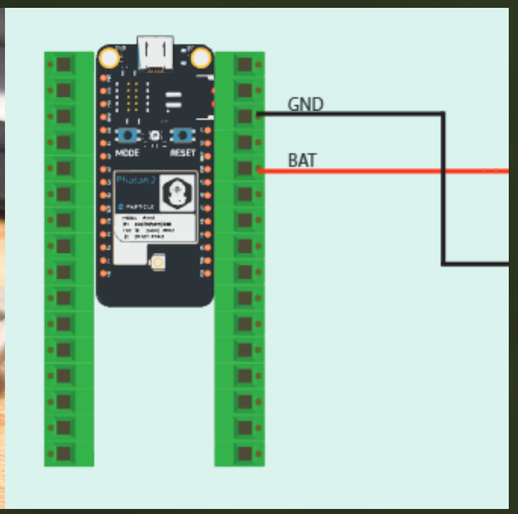


3. Secure the PCB Assembly

- Screw down the terminal board, charge controller, and Photon assembly onto the plate.
- Connect the charge controller's output to the corresponding power input terminals on the terminal board (ensure polarity is correct to avoid damaging the Photon) .







4. Prepare the Enclosure

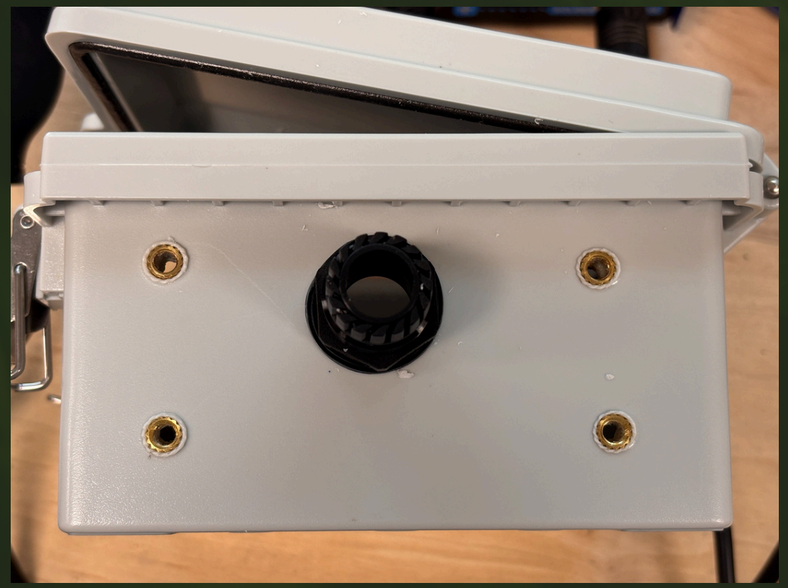
- Use the mounting jig to mark and drill screw holes on top of the plastic enclosure.

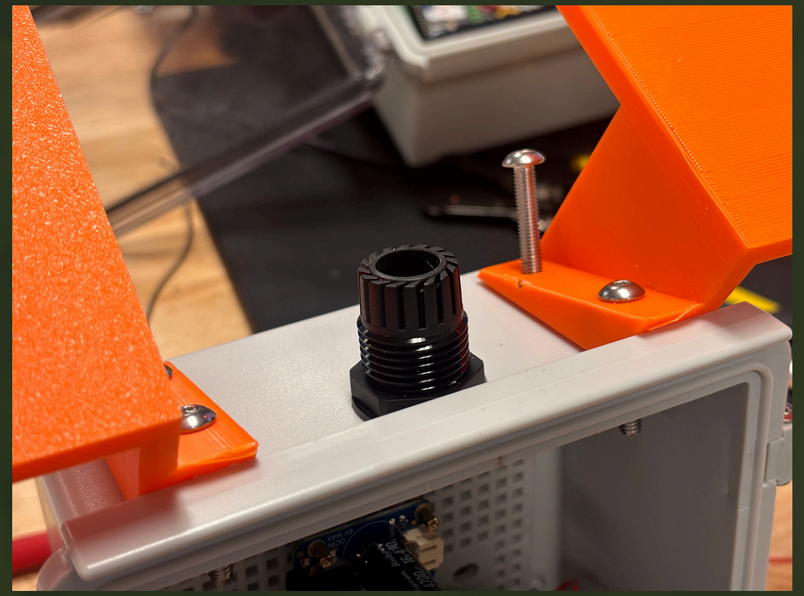
- Drill an additional hole for cable routing (for sensor wires or power cables)

- Insert threaded brass inserts in the drilled holes to allow secure screwing.

5. Install Cable Gland and Support (orange)

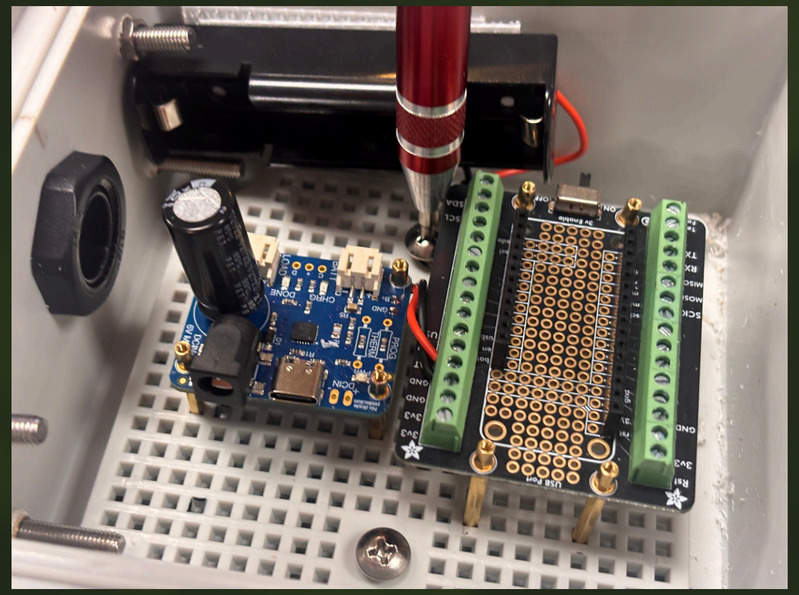
- Install the cable gland into the drilled wire hole; tighten it securely.
- Install the support using screws and the threaded inserts.





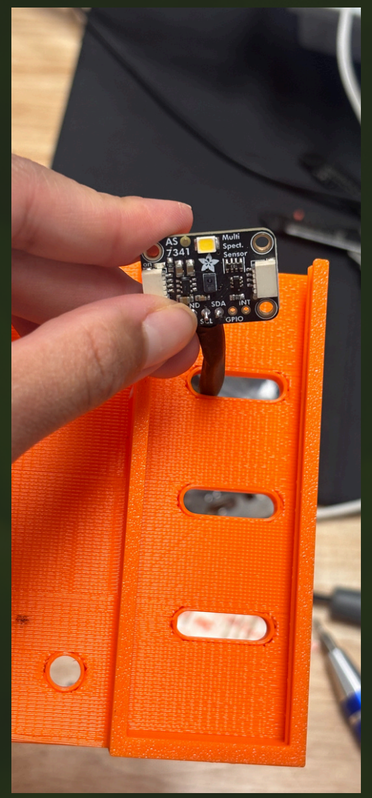
6. Place Internal Components

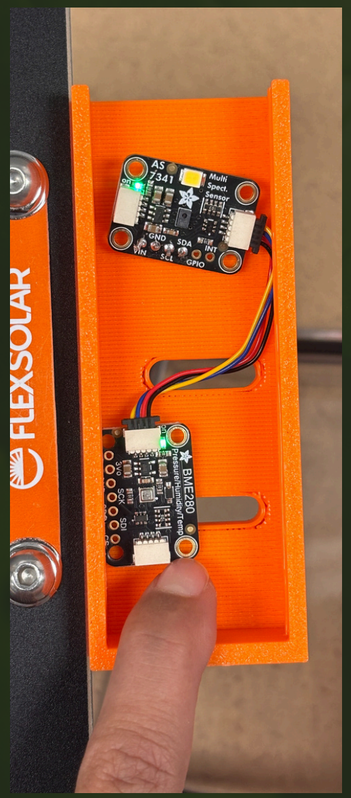
- Insert the assembled PCB (Photon 2, terminal board, charge controller) into the enclosure.
- Place the battery holder on the wall of the box using velcro.

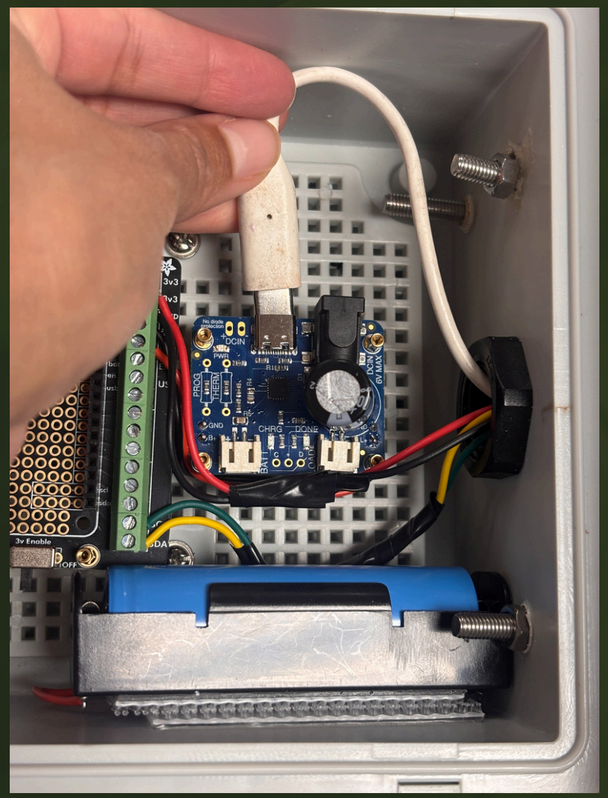


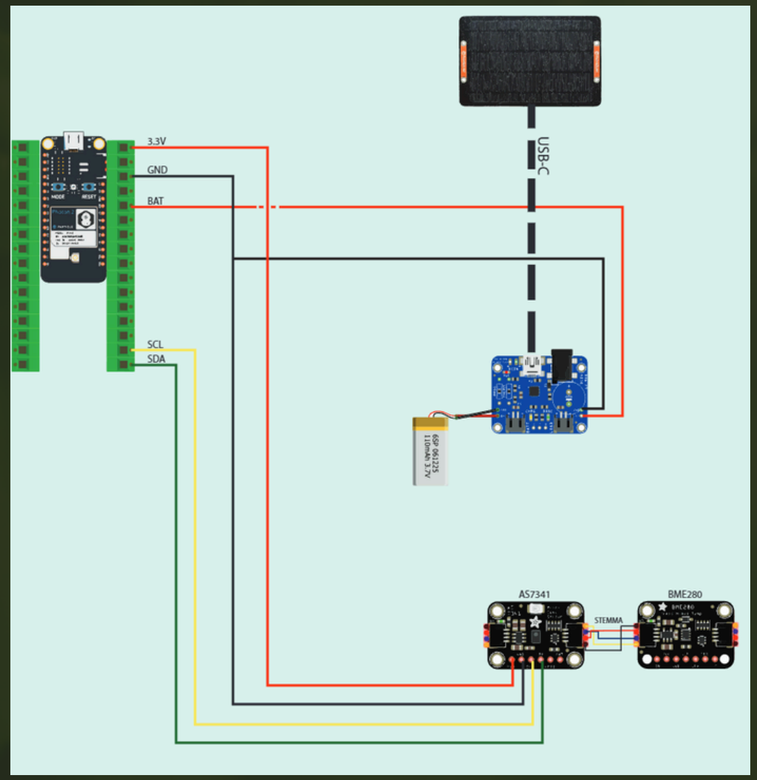
7. Mount External Parts

- Mount any external sensors and solar panels onto the brackets.
- Feed sensor or panel wires through the cable gland.
- Plug the solar panel’s USB-C connector into the charge controller’s USB-C input port to allow solar charging.
- Connect any sensor wires to the appropriate input terminals on the terminal board.



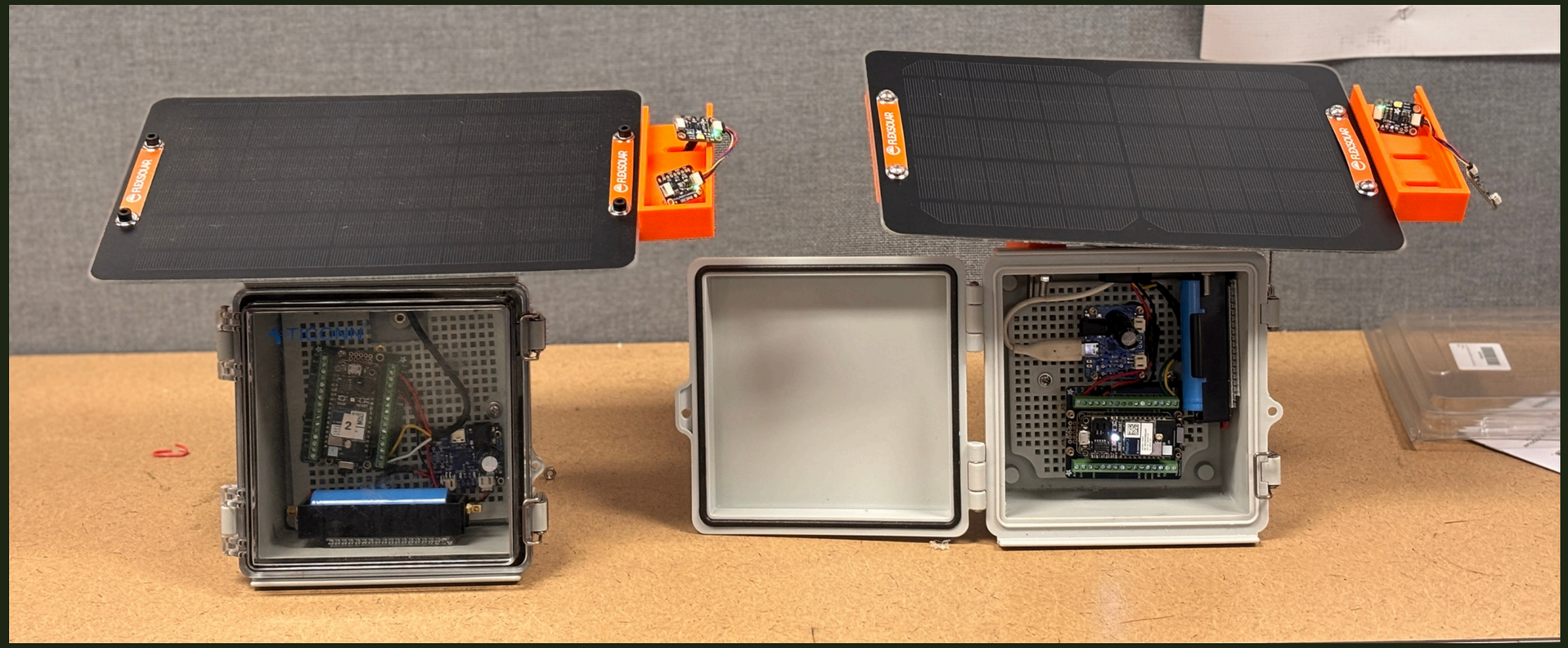




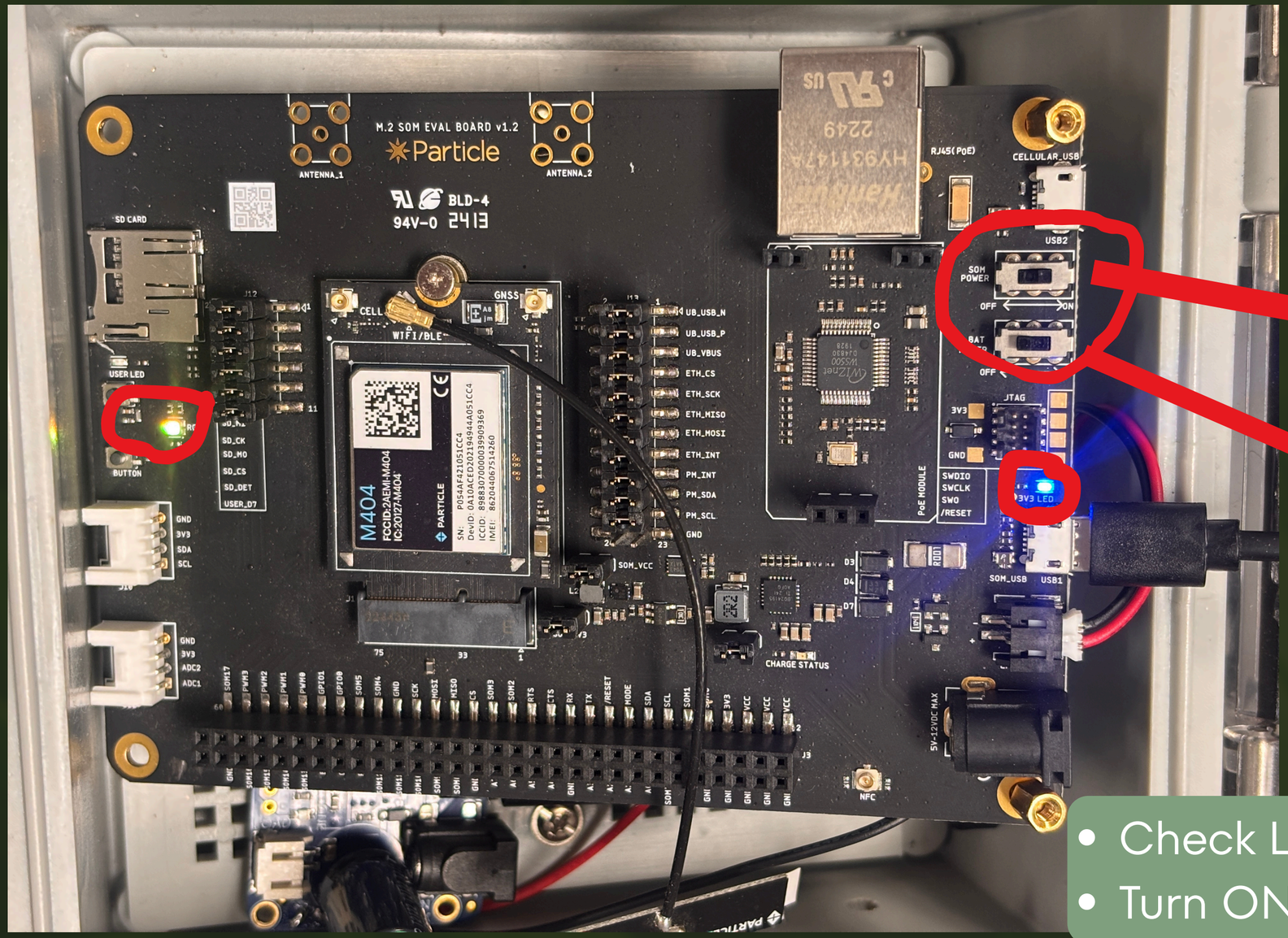


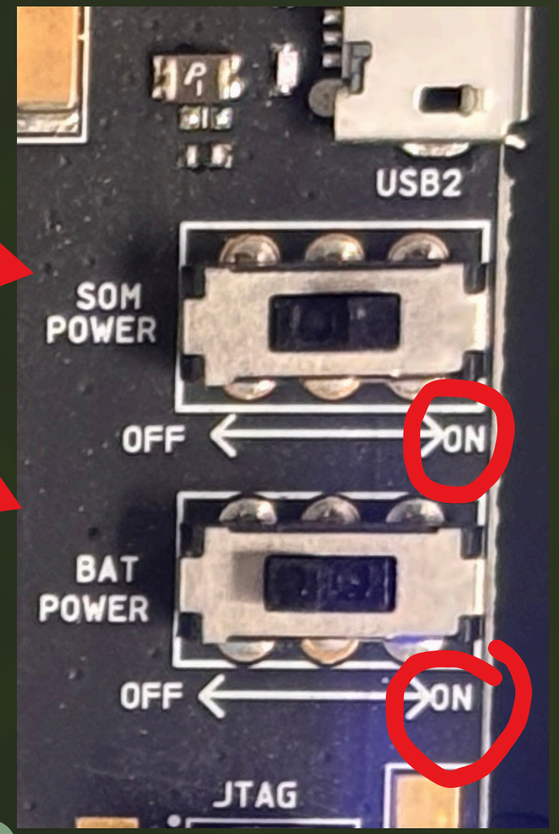
8. Final Checks

- Double-check all connections.
- Make sure screws are tight and no wires are pinched.
- Close and latch the enclosure securely



7. Turning ON Base Station





- Check LED status
- Turn ON BAT POWER and SOM POWER

8. Viewing Data on Node-RED

Logging into Node-RED

- Access via browser with IP address.

Configuring Nodes

- Add MQTT-in nodes for each sensor
- Use function nodes to parse data

Viewing Debug Window

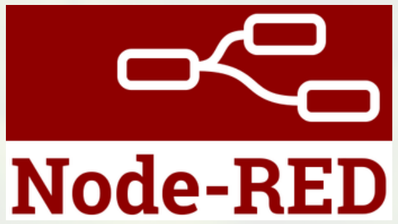
- Use debug node for each data stream
- Check timestamps and values

Google Sheets Integration

- Install the node-red-contrib-google-sheets node

Creating a Node-RED Dashboard

- Use UI nodes (charts, gauges)
- Organize data by time and variable



iNode-RED Google Sheets Integration Node-RED. node-red-contrib-google-sheets-advance. Available at: https://flows.nodered.org/node/node-red-contrib-google-sheets-advance

9. Troubleshooting, Documentation, & Resources

Particle troubleshouting

ISSUE	SOLUTION
RGB LED is off	Photon2 is not plugged. • Disconnect GND from board • Remove Photon2 brom board and plug back in
Photon 2 not connecting to Wi-Fi	RGB is fast flashing green WIFI crdentials are not correct. To set WIFI credentials use: particle serial wifi
Not automatically going into DFU-Mode	If the Photon2 fails to automatically enter DFU-mode during Flash • Manually place the Photon2 in DFU-mode. And then from Terminal/Git Bash: particle flash -- usb tinker • If this is a recurring issue, turn on multi-treading. In the code's header, place SYSTEM_THREAD(ENABLED)

Device troubleshouting

ISSUE	SOLUTION
Sensors not reading	• Check SDA/SCL connections. • Reseat connectors or solder joints.
Battery not charging	Test solar panel with multimeter (should output ~5V)
Photon 2 LED doesn't blink or stays off	Press RESET to wake up device, Check all wiring.
Base station does not power ON	Check that BAT POWER and SOM POWER are ON.

DATASHEETS

BME280 – Adafruit Product Page https://www.adafruit.com/product/2652

AS7341 – Adafruit Product Page https://www.adafruit.com/product/4698

Particle Photon 2 – Datasheet https://docs.particle.io/photon-2/datasheet/

TP4056 Li-Ion USB Charger with Protection. Adafruit. Available at: https://www.adafruit.com/product/390

RESOURCES

Visual Studio Code (VSCode) – Editor Download https://code.visualstudio.com

Google Sheets Integration for Node-RED https://flows.nodered.org/node/node-red-contrib-google-sheets-advance

Node-RED Docs https://nodered.org/docs/