TinyCamML build guide

Last edited: November 27, 2024

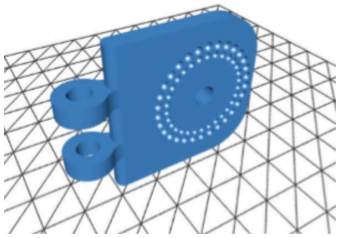
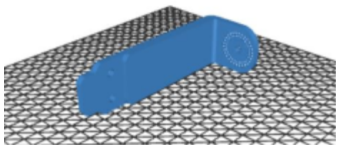
https://github.com/TinyCamML/Boron-and-OpenMV

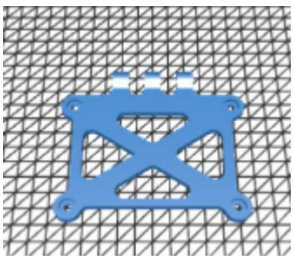
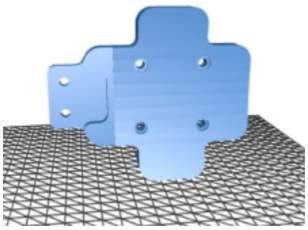
Table of Contents

Building the housing

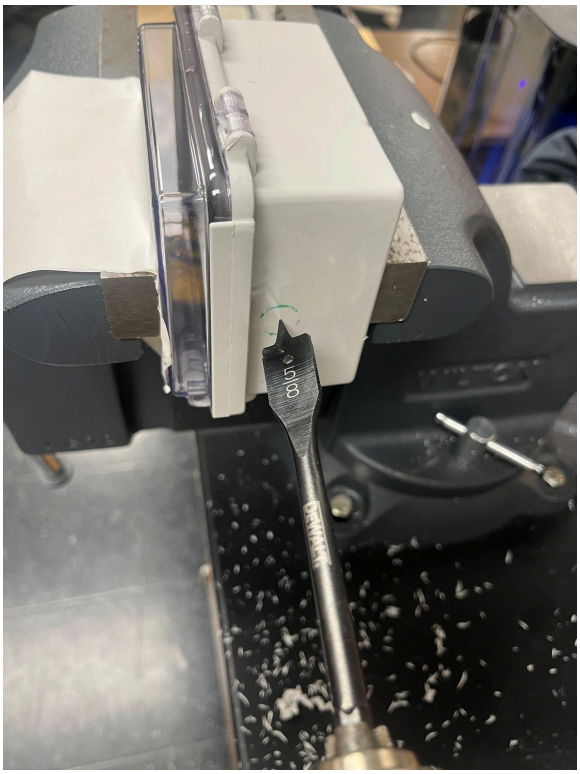
Materials needed

Part	Quantity needed	Picture	Link to item
5W 5V solar system	1		Voltaic
Female 3.5x1.1mm - MicroUSB	1		Voltaic
Polycase WH-02-02 Clear	1		Polycase
PG-9 size cable gland	1		Adafruit
1/4-in x 2-1/2-in Stainless Coarse Thread Hex Bolt	1		McMaster

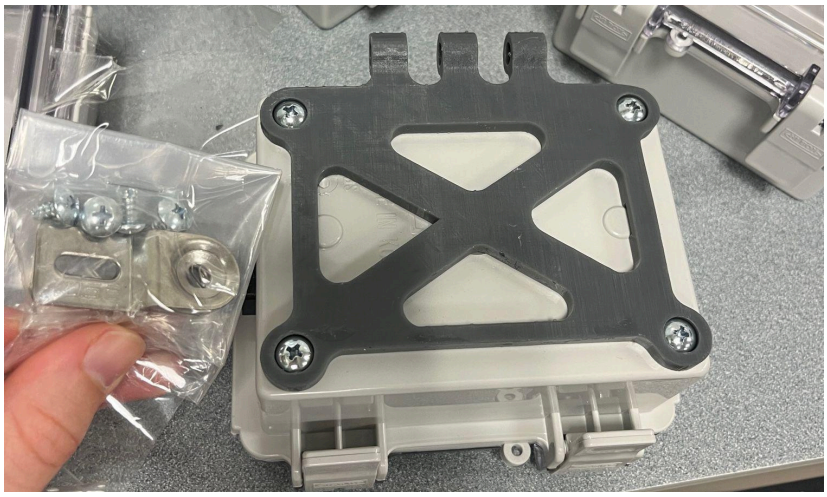
1/4-in x 20 Stainless Steel Stainless Steel Nylon Insert Nut	1		McMaster
18-8 Stainless Steel Socket Head Screw, M5 x 0.8 mm Thread, 20 mm Long	7		McMaster
18-8 Stainless Steel Hex Nut M5 x 0.8 mm Thread	7		McMaster
#8-16 x .31” screws	4		Comes with the polycase
3D Printed Parts:			
XY Bracket	1		3D design
Extender	1		3D Design

Polycase mount	1		3D Design
Solar panel mount	1		3D Design
*Note all 3D printed parts were printed on a FormLabs Form 3 3D printer, using Tough 2000 resin.			

1. Take the polycase and secure it for drilling. Trace around the cable gland as a guide for your drilling. Drill the hole in the upper half of the right side of the polycase, if the hinges are pointed upward and the clear front is facing you. Use a ⅝” spade bit to drill out the hole.



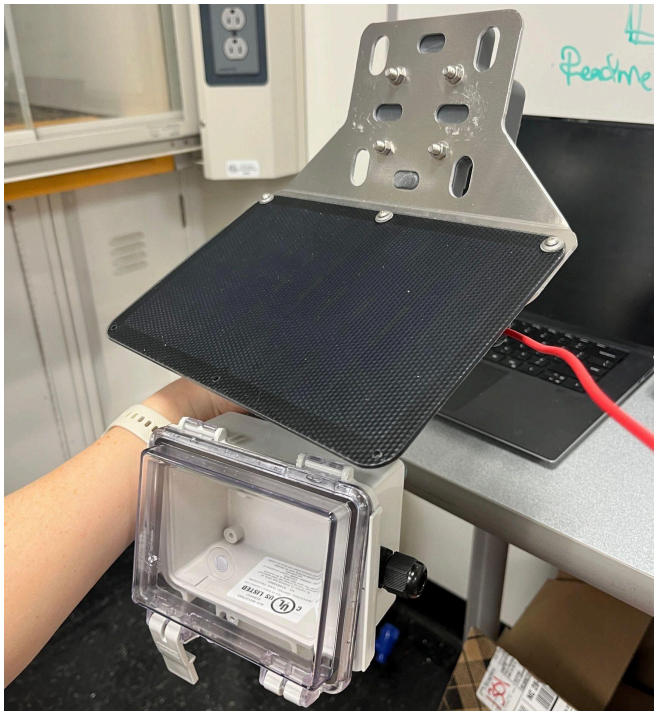
2. Deburr the hole and install the cable gland. Remember to include the o-ring and use o-ring grease. Tighten both sides of the cable gland, so it is securely fastened.
3. Using the four #8-16 screws that came with the polycase, install the polycase mount to the back of the polycase, being careful not to overtighten.



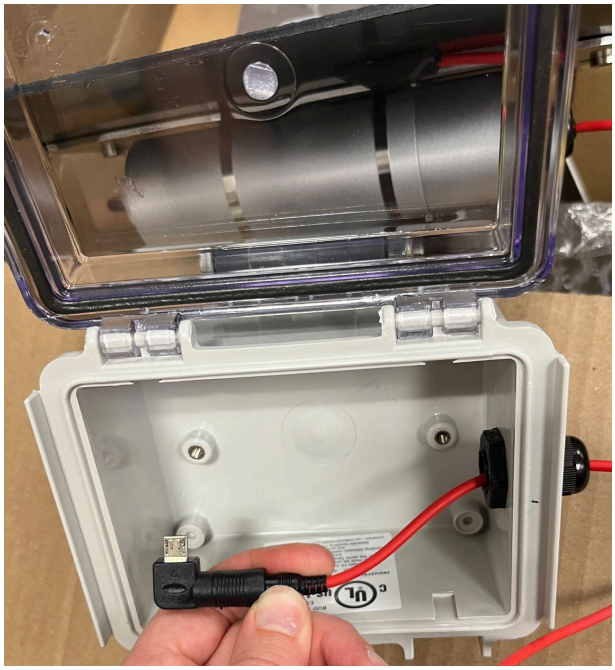
4. Install the XY bracket to the polycase mount using one 1/4-in x 2-1/2-in Hex Bolt and one 1/4-in x 20 Nylon Insert Nut. Pro-tip: It helps to line up the XY bracket and polycase mount and drill through the circular opening beforehand with a ¼ in drill bit to make inserting the hex bolt easier. You don't need to fully tighten at this stage because you'll have to adjust it during deployment anyways.

5. Use one M5 x 0.8mm x 20 mm socket head screw and one M5 x 0.8 mm nut to fasten the extender onto the XY bracket. The nut should fit into the recessed hexagon on the XY bracket. Then attach the solar panel mount to the extender using two more M5 x 0.8mm x 20 mm socket head screws and two M5 x 0.8 mm nuts.

6. Attach the solar panel to the solar panel mount using four M5 x 0.8mm x 20 mm socket head screws and four M5 x 0.8 mm nuts.

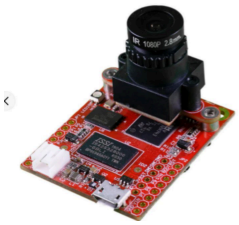
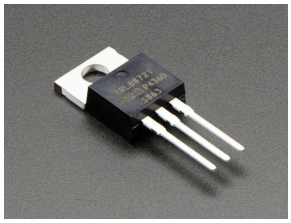
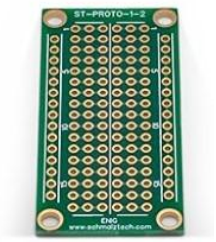


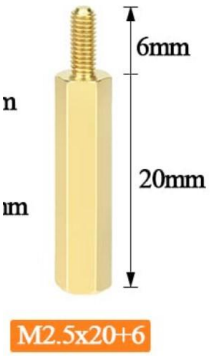
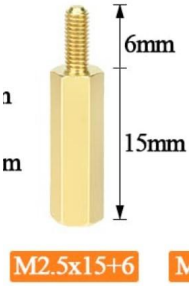
7. Fit the output cable of the solar panel through the cable gland and put the Female 3.5x1.1mm - MicroUSB convertor on the end of the cable.

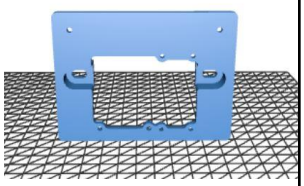
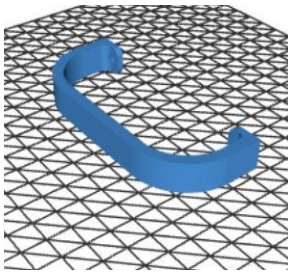


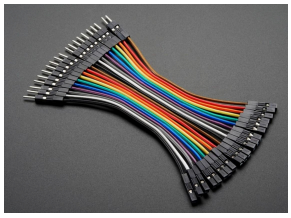
Putting together the electronics

Materials needed

Part	Quantity needed	Picture	Link to item
OpenMV Cam H7 Plus	1		OpenMV
MicroSD card	1		Amazon
Particle Boron	1		Particle
N Channel MosFET switch 30V/60a	1		Adafruit
1” x 2” in protoboard	1		Amazon

10k Ω resistor	1		Sparkfun
M2.5 x 20 mm + 6 mm standoff	2		Amazon
M2.5 x 15 mm + 6 mm standoff	2		Amazon
M2.5 x 6 mm screws	4		Amazon
M2.5 nuts	4		Amazon

M3 x 10 mm + 6 mm standoffs	2		Amazon
M3 x 6 mm screws	2		Amazon
M3 nuts	2		Amazon
Baseplate	1		Github
Antenna mount	1		Github

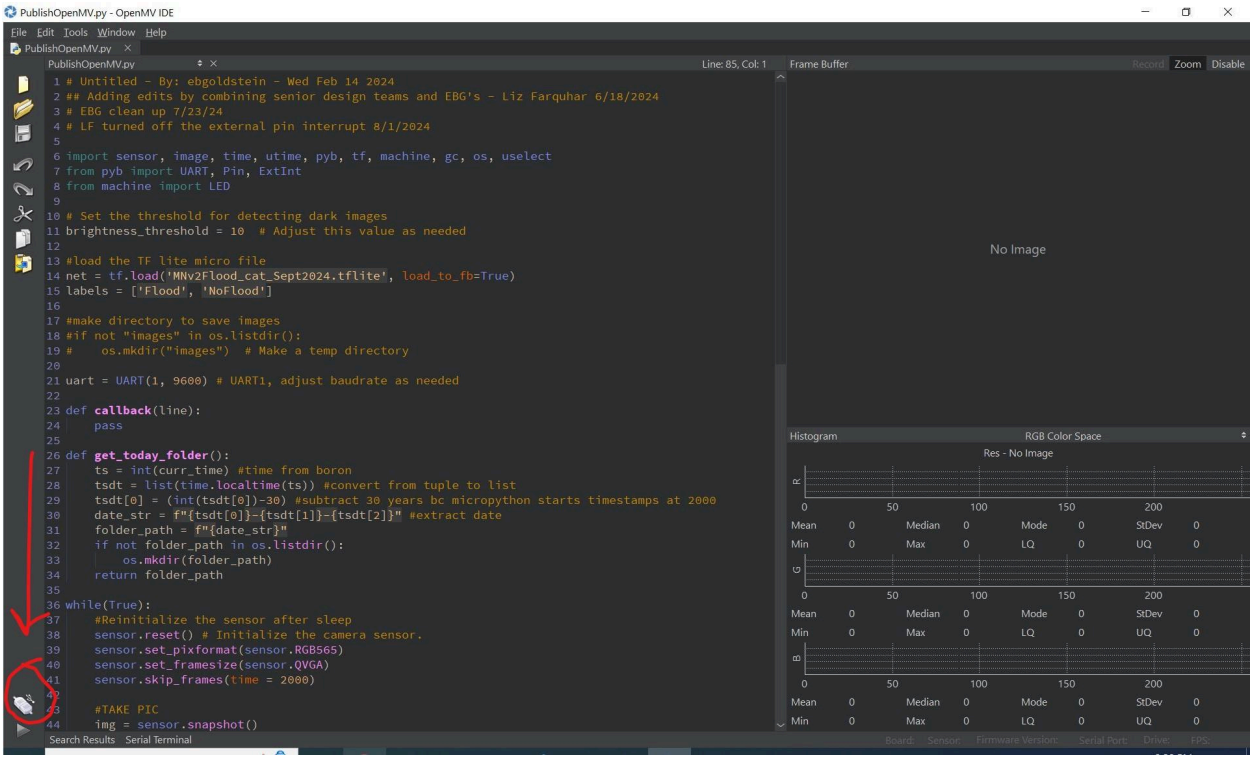
Stainless Steel Pan Head Phillips Screws, M4 x 0.70 mm Thread, 8mm Long	2		McMaster
4-40 ⅜” screws	2		McMaster
4-40 nuts	2		McMaster
Jumper cables	6		Adafruit and Adafruit

Prepping the microcontrollers

OpenMV

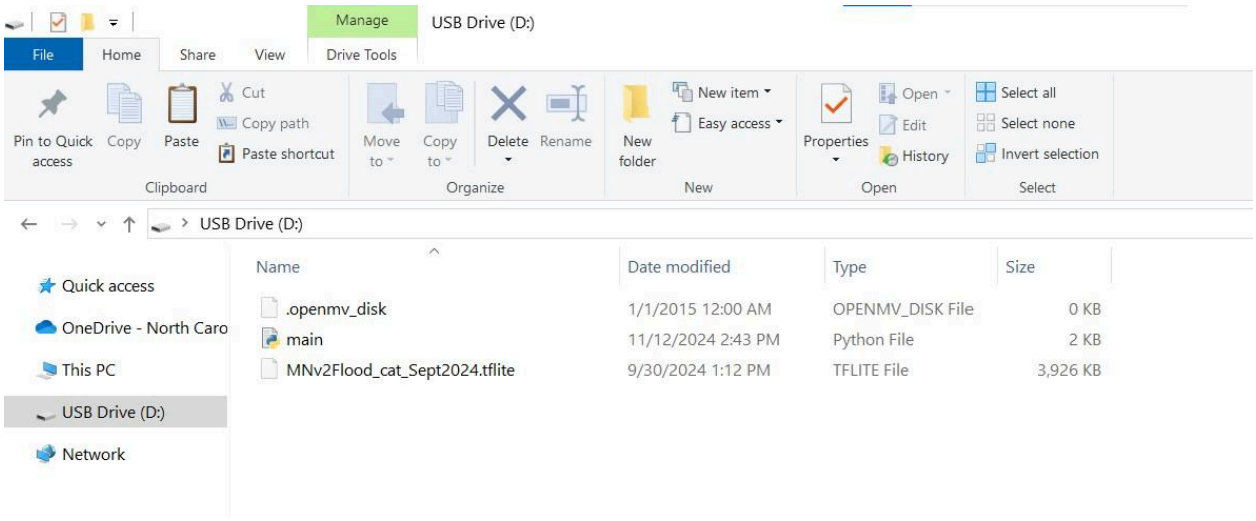
1. Solder on the female header pins that came with the OpenMV onto the backside of it. Insert the micro sd card as well.

2. Open up the OpenMV IDE, and use a micro-usb cable to connect the OpenMV to your laptop. Once in the IDE, navigate to File > Open File, and open the source code file, which you should have downloaded from here.
3. Click the connection button in the bottom left corner of the screen.



4. You will likely be prompted, “Your OpenMV Cam’s firmware is out of date. Would you like to upgrade?” Select Yes. It’ll then ask you if you want to erase the internal file system. Select Yes.

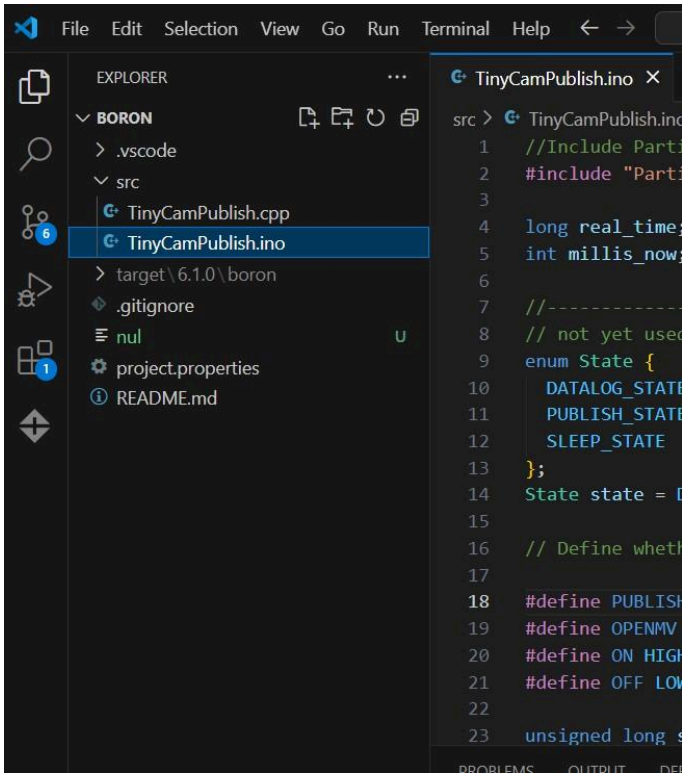
5. To make sure the camera is in focus, navigate to File>Examples>HelloWorld>helloword.py . This will open up a new tab of code. Press the green play button in the bottom left corner of the screen to run this code. The camera image will show up on the right side of the screen, so you can check to make sure it looks okay. Most of the time, the OpenMVs come pre-focused, but if you find yours is not in focus, follow the steps here to focus it.
6. Now, exit out of the helloworld.py tab. Navigate to Tools> Save open script to OpenMV Cam (as main.py). It will ask if you want to strip comments and spaces. Select yes. This saves the code onto the OpenMV's sd card, so it knows what to run.
7. To make sure your OpenMV's source code saved properly, you can navigate to your file explorer, where the OpenMV should be mounted as a usb drive and check that the main.py and the openmv_disk files are both in there.
8. Copy the most updated machine learning model .tflite file, found here, into the OpenMV's usb storage.



Boron

1. For the Boron microcontroller, download the Particle App on your smartphone and login to your Particle account. If you do not have a smartphone or have issues with the setup, you can also set up the device using the information found here. Attach the antenna that came with the Boron onto it and plug the boron into a power source. A microusb cable to your computer works. If it's not automatically blinking blue when you plug it in, hold down the mode button until it is.
    a. Pro-tip: If it is stuck on blinking green for a very long time, you may just have to let it run overnight to have it connect for the first time.
2. Go onto the Particle app after logging in and hit the + sign in the upper right corner of the screen and hit "Set up a new Argon/Boron." From there, the app will walk you through setting up the Boron.

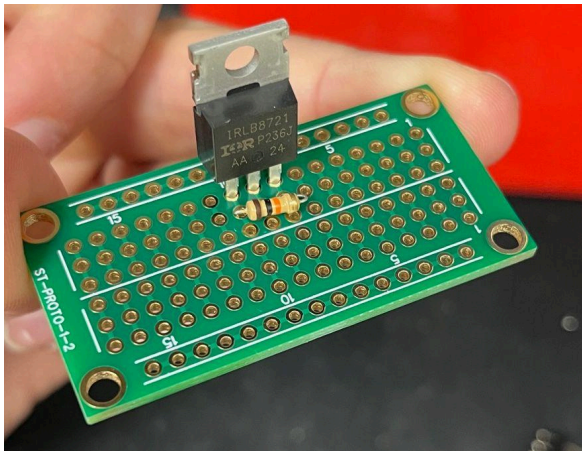
3. Now open up Visual Studio Code. Navigate to the Extensions button on the left side of the screen and type in “Particle” and download the Workbench option. You may need to restart Visual Studio after downloading the extension.
4. Navigate to File>Open Folder, and open the entire Boron folder that can be downloaded here. If it opened properly, you should be able to click on the src subfolder and open the .ino file. Make sure at the bottom of your screen you select what device you are targeting (e.g., Boron) and select the device.os firmware version you want (deviceOS@6.1.0), which will prompt the program to download that firmware package.



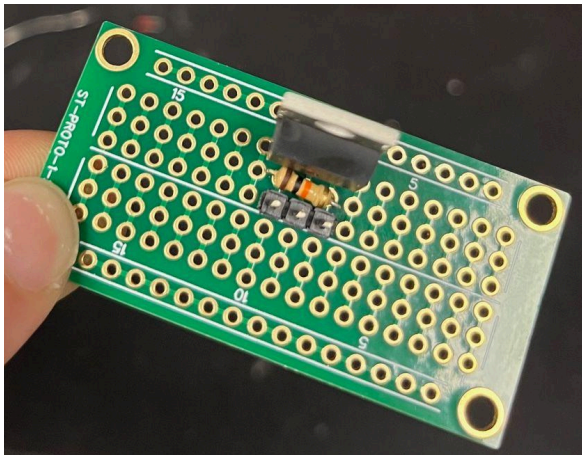
5. From here, hit the check mark button in the upper right corner of the screen that says “Compile.” The first time you compile something, it may take a few minutes. After it compiles successfully, ensure your Boron is plugged into your computer via microusb cable and hit the lightning bolt icon that says “Flash.” Now the Boron will run the code every time it’s powered on.

Creating the Mosfet switch

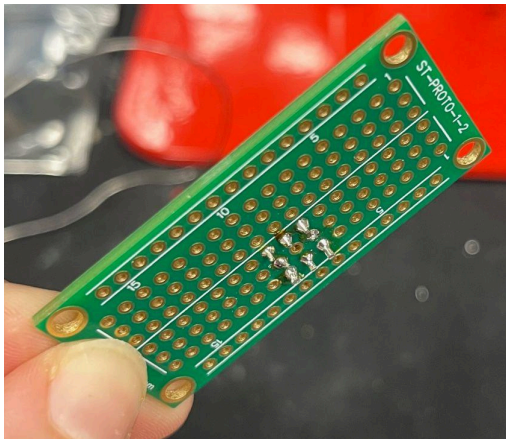
1. Take the 1x2 in protoboard and solder on the N Channel MosFET switch about halfway down the protoboard on the edge, so each of its prongs are on a different row and its tall side is facing away from the board. Solder on the 10k ohm resistor, so that the resistor is on the same rows as the first and third prongs of the mosFET.



2. Of the male header pins that came with the OpenMV, break off a row of 3 of them and solder those onto the board, so each pin is in the same row with the mosFET's prongs.

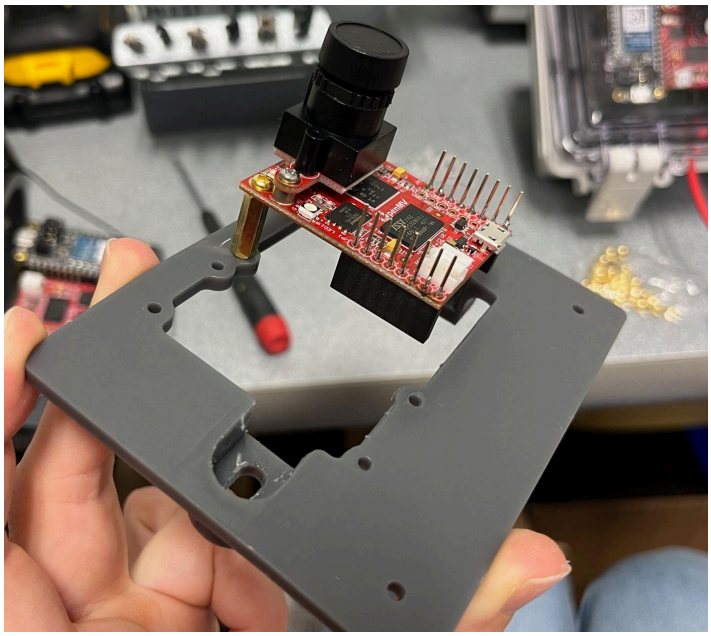


3. Trim the excess length of the resistor and mosFET prongs on the backside so they cannot bend and touch each other.

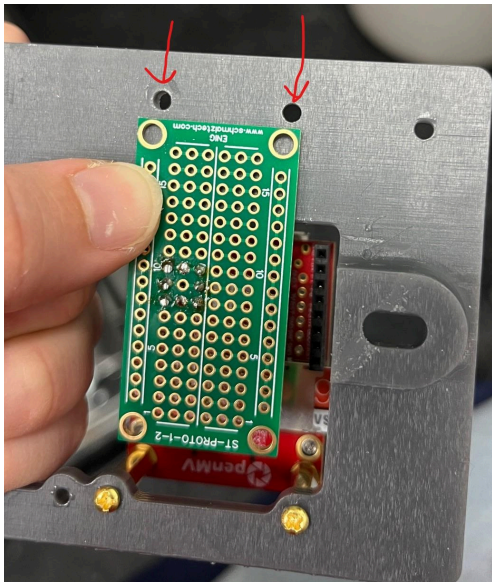


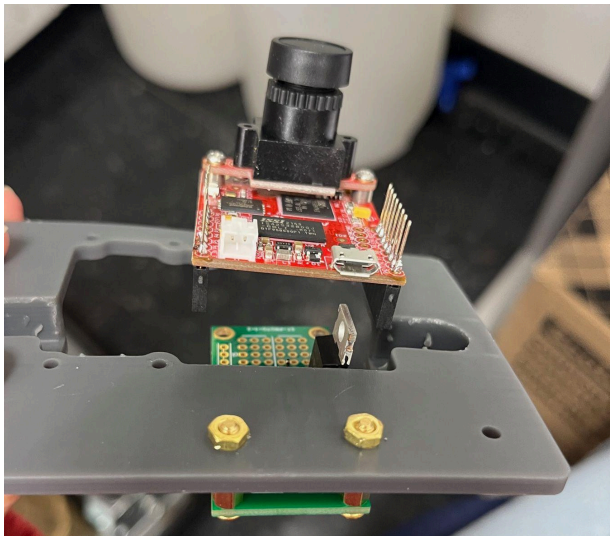
Mounting electronics

1. Mount the OpenMV to the baseplate using two M2.5x15 mm standoffs, two M2.5 nuts, and two M2.5 screws.

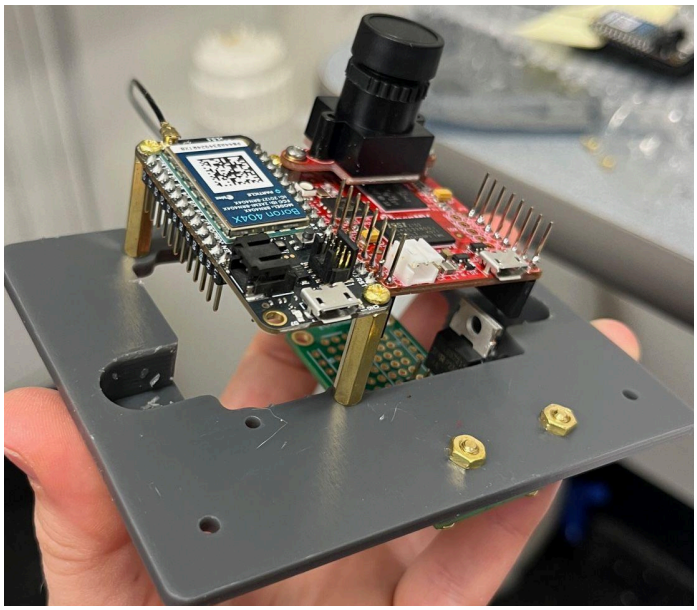


2. Drill M3.5 holes to add the mosFET switch onto the baseplate, ensuring that the actual mosFET will not be in the way of any connections to the OpenMV. Then mount the mosFET switch onto the backside of the baseplate using two M3x10 mm standoffs, two M3 nuts, and two M3 screws.

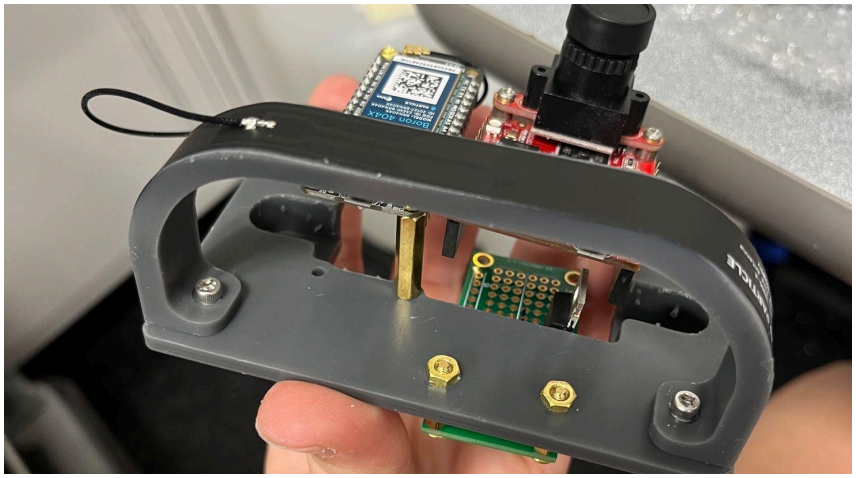




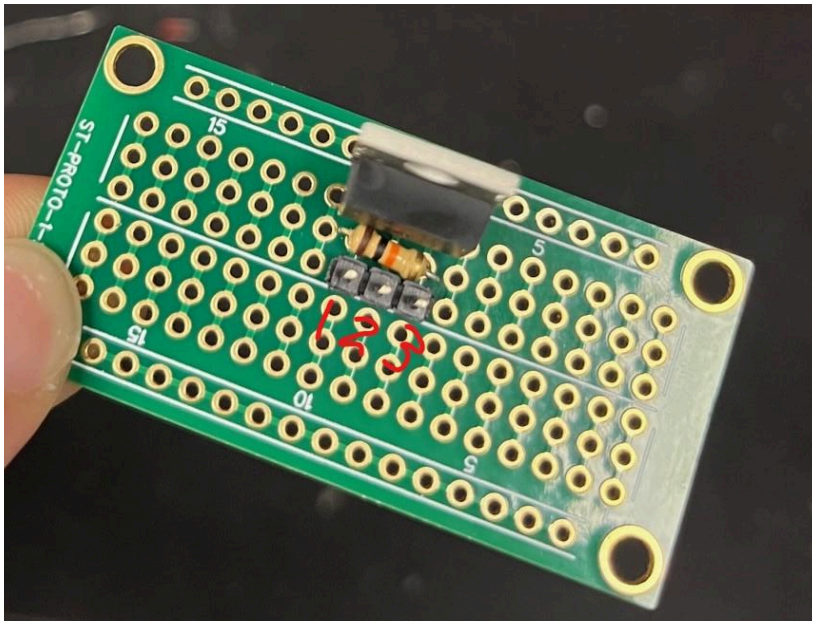
3. Mount the Boron to the baseplate using at least two M2.5x20 mm standoffs placed diagonally from each other, two M2.5 nuts, and two M2.5 screws.

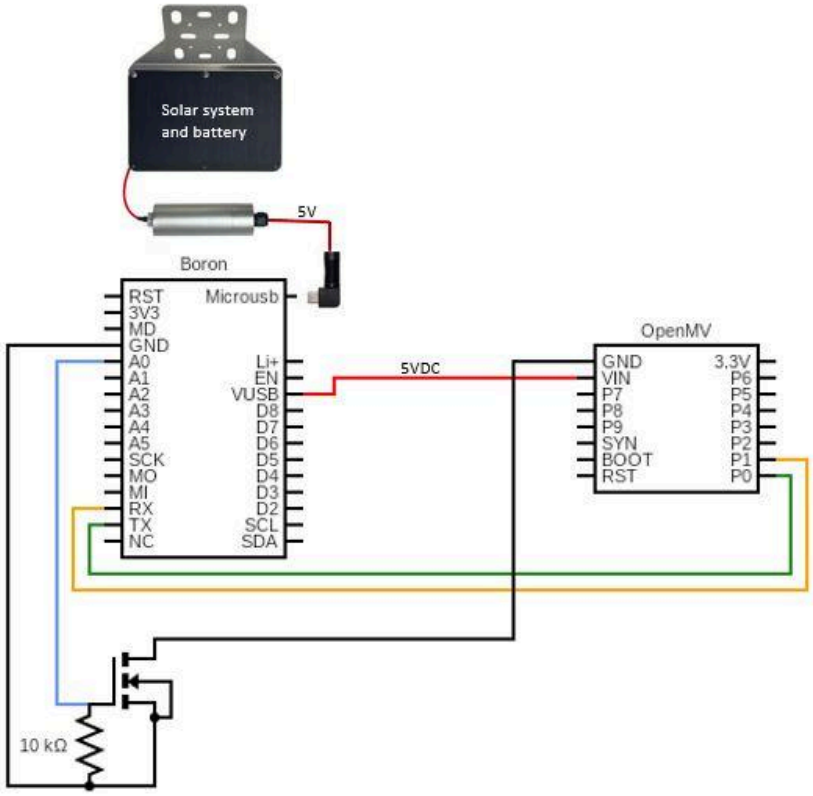


4. Attach the antenna mount to the baseplate using two 4-40 3/8th in screws and two 4-40 nuts

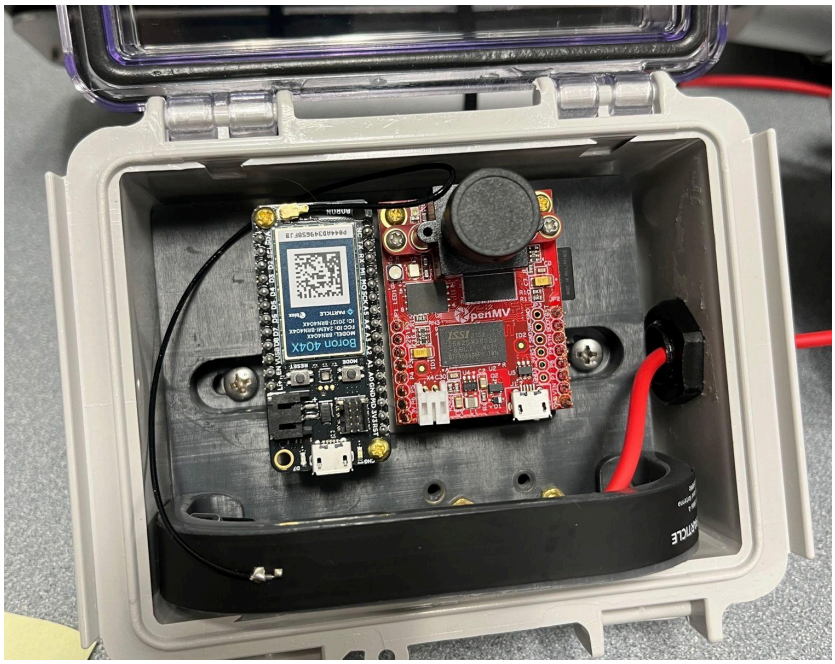


5. Attach the jumper cables to the electrical components using the following electrical diagram:
    a. Note: it's easier to start with the jumper cables that connect to the mosFET: A female-female jumper cable between the Boron's GND to the third prong of the mosFET. A male-female jumper cable between the OpenMV's GND and the middle prong of the mosFET. And then a female-female jumper between the Boron's A0 to the first prong of the mosFET.





6. Secure the baseplate to the inside of the polycase using two M4 x 0.70 x 8 mm pan head screws.



Creating the webhook integration

Before your TinyCamML will successfully publish results to a google sheet, you need to set up the webhook integration in the Particle Console.

1. Navigate to https://console.particle.io/home and login. On the left side of the screen, click “Products” and Create a New Product using the button in the upper right corner of the screen.
2. Once your product is created, you also need to go back to the main Sandbox page and navigate to “My Devices,” on the left side of the screen. You should see all your Borons claimed and named on this screen. Copy the ID of the Boron you are using in your TinyCamML.
3. Open your Product that was created in Step 1, and click “Add Device.” When prompted, paste the Boron ID from Step 2. Now your Boron is specifically associated with the TinyCamML product.
4. Now, follow the steps outlined here to finish the integration process: https://github.com/COAST-Lab/Open-Water-Level/tree/main/Particle_Integration
    a. Note that in the source code for the Boron, we have defined the event name to be “FloodorNo,” so if you change the event name in the integration, you’ll need to change it in the source code as well.
5. Now when you plug in power to the Boron, it should run through the source code and publish to Google sheets in real time. The first publish often will not include “Flood” or “No Flood,” but it will after the first.

Deploying the TinyCamML

Materials needed

- 1-2 stainless steel hose clamps
- O-ring grease
- Kim wipes or something similar to clean the front of the TinyCamML
- Laptop and microusb cord
- Tools for tightening/loosening the XY bracket bolts
- Desiccant pack

1. You should test your TinyCamML for a few days before deploying in a remote area to ensure the OpenMV is saving images properly and the Boron is publishing to Google sheets.
    a. Note that the first time you flash firmware and connect your Boron to the cloud, it probably will run a firmware update (blink magenta) for a few minutes.
2. It may be necessary to pre-charge your solar panel system before deployment. If that’s the case, simply unscrew the side of the battery that has a cable gland and use the

USB-A to DC cable to plug one side into a computer usb socket and the DC port into the other side of the battery. The light on the battery should be red when dead, green when fully charged.

3. To install the TinyCamML in the field, use 1-2 stainless steel hose clamps to secure the solar panel to the pole (or whatever you're attaching to).
4. Make sure the TinyCamML is facing where you want it by loosening/tightening the 2 ½ inch bolt between the XY bracket and polycase mount (to change the up/down tilt of the camera) and the bolt between the XY bracket and the extender (to change the direction it's looking).
    a. You can check the view as you go by plugging a microusb cord into the OpenMV, connecting to it, and running the Helloworld.py example script.
5. Neaten up the loose solar panel wires and secure the extender to the pole with zip ties
6. Plug the solar panel cable into the Boron and ensure that it turns on and connects to the cloud and publishes. Often the first publish doesn't include "Flood" or "No Flood," but it will after that.
7. Ensure you leave a desiccant pack inside and grease the o-ring of the polycase before leaving it, and clean the clear side of the polycase of smudges. Also make sure to mark your TinyCamML with something like "NCSU Research" to deter people from tampering with it.
8. Grease the outside of the cable gland as well to decrease water intrusion.