

TinyCamML build guide

Last edited: June 23, 2025

<https://github.com/TinyCamML/>

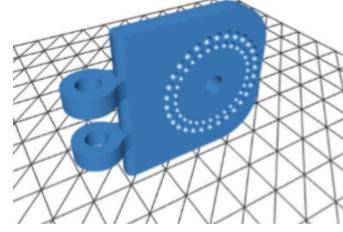
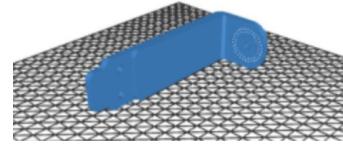
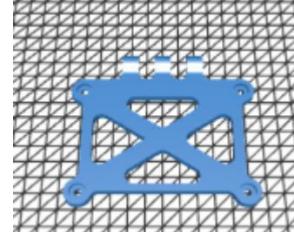
Table of Contents

Building the housing	1
Materials needed	1
Putting together the electronics	7
Materials needed	7
Prepping the microcontrollers	10
OpenMV	10
Boron	12
Mounting electronics	13
Creating the webhook integration	17
Deploying the TinyCamML	17
Materials needed	17

Building the housing

Materials needed

Part	Quantity needed	Picture	Link to item
5W 5V solar system	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 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	 <small>Sold as bags of 50 sci (1.6kg + 50 screws)</small>	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 $\frac{5}{8}$ " 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.



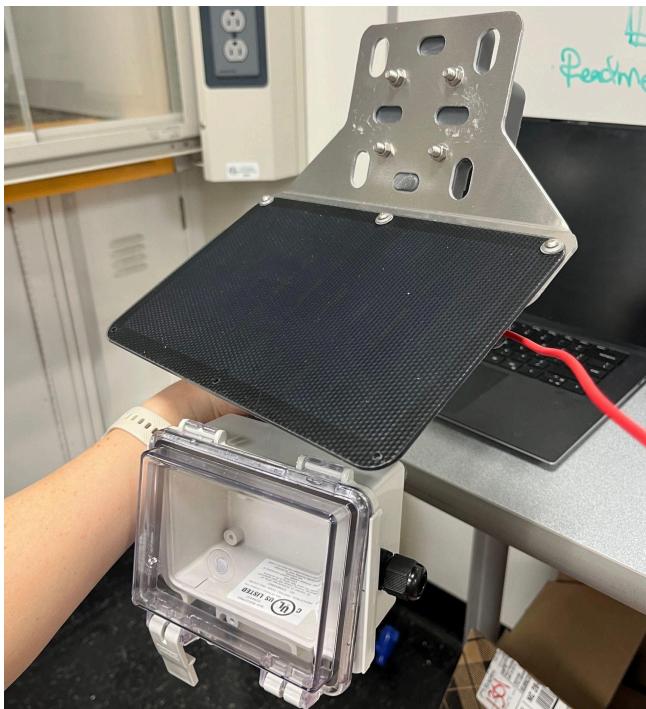
4. Install the XY bracket to the polycase mount using one 1/4-in x 2-1/2-in Hex Bolt and one 1/4-in x 20 Nylon Insert Nut. Pro-tip: It helps to line up the XY bracket and polycase mount and drill through the circular opening beforehand with a 1/4 in drill bit to make inserting the hex bolt easier. You 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.

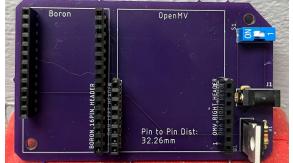
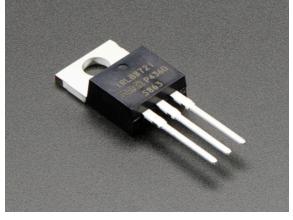


7. Fit the output cable of the solar panel through the cable gland

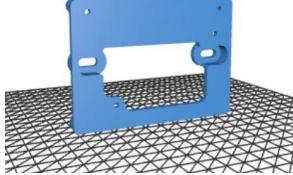
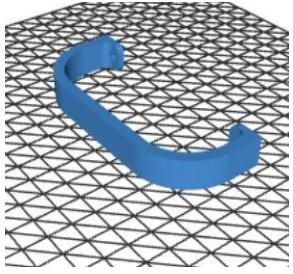


Putting together the electronics

Materials needed

Part	Quantity needed	Picture	Link to item
Custom PCB Board	1		Github
OpenMV Cam H7 Plus	1		OpenMV
MicroSD card	1		Amazon
Particle Boron with cellular antenna	1		Particle
N Channel MosFET switch 30V/60a	1		Adafruit

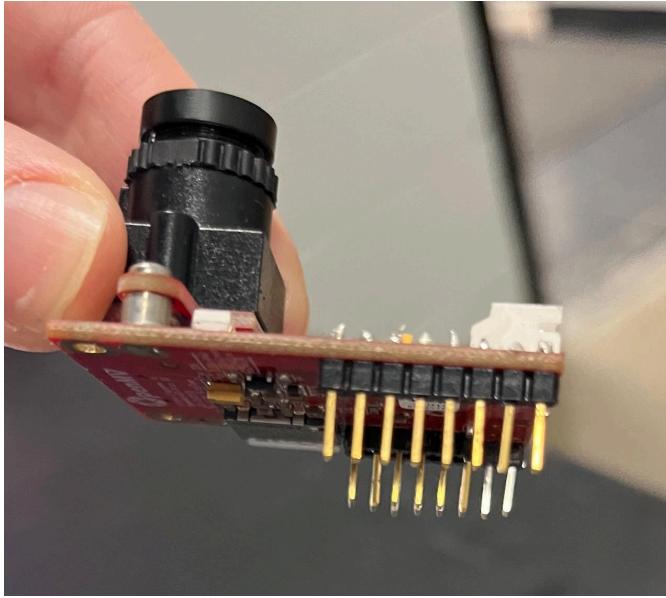
Female-male headers	1 - row of 12 1 - row of 16 2 - rows of 8		Adafruit (12 and 16) (Just cut the 16 pin ones in half for rows of 8)
Female barrel plug	1		Digikey
Switch slide	1		Digikey
10k Ω resistor	1		Sparkfun
M2.5 x 12 mm screws	2		McMaster
M2.5 nuts	2		McMaster

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 3/8" screws	4		McMaster
4-40 nuts	4		McMaster

Prepping the microcontrollers

OpenMV

1. Solder on the male header pins that came with the OpenMV onto the backside of it, so the longer pins are facing down. 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.

```

PublishOpenMv.py - OpenMV IDE
File Edit Tools Window Help
PublishOpenMv.py ×
Line: 85, Col: 1 Frame Buffer Record Zoom Disable
1 # Untitled - By: ebgoldstein - Wed Feb 14 2024
2 ## Adding edits by combining senior design teams and EBG's - Liz Farquhar 6/18/2024
3 # EBG clean up 7/23/24
4 # LF turned off the external pin interrupt 8/1/2024
5
6 import sensor, image, time, utime, pyb, tf, machine, gc, os, uselect
7 from pyb import UART, Pin, ExtInt
8 from machine import LED
9
10 # Set the threshold for detecting dark images
11 brightness_threshold = 10 # Adjust this value as needed
12
13 #Load the TF lite micro file
14 net = tf.load('MN2xFlood.cat_Sept2024.tflite', load_to_fb=True)
15 labels = ['flood', 'noflood']
16
17 #make directory to save images
18 #if not "images" in os.listdir():
19 #    os.mkdir("images") # Make a temp directory
20
21 uart = UART(1, 9600) # UART1, adjust baudrate as needed
22
23 def callback(line):
24     pass
25
26 def get_today_folder():
27     ts = int(curr_time) #time from boron
28     tsdt = list(time.localtime(ts)) #convert from tuple to list
29     tsdt[0] = (int(tsdt[0])-30) #subtract 30 years bc micropython starts timestamps at 2000
30     date_str = f'{tsdt[0]}-{tsdt[1]}-{tsdt[2]}' #extract date
31     folder_path = f'{date_str}'
32     if not folder_path in os.listdir():
33         os.mkdir(folder_path)
34     return folder_path
35
36 while(True):
37     #Reinitialize the sensor after sleep
38     sensor.reset() # Initialize the camera sensor.
39     sensor.set_pixformat(sensor.RGB565)
40     sensor.set_framesize(sensor.QVGA)
41     sensor.skip_frames(time = 2000)
42
43     #TAKE PIC
44     img = sensor.snapshot()

```

Search Results: Serial Terminal

No Image

Histogram RGB Color Space

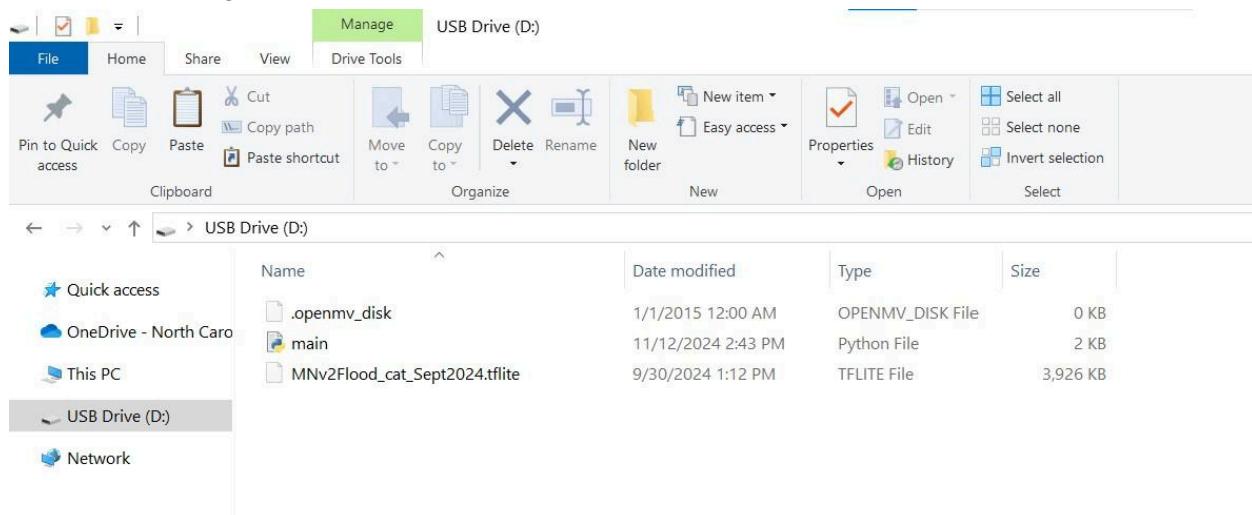
Res - No Image							
	0	50	100	150	200		
Mean	0	Median	0	Mode	0	StDev	0
Min	0	Max	0	LQ	0	UQ	0
	0	50	100	150	200		
Mean	0	Median	0	Mode	0	StDev	0
Min	0	Max	0	LQ	0	UQ	0
	0	50	100	150	200		
Mean	0	Median	0	Mode	0	StDev	0
Min	0	Max	0	LQ	0	UQ	0

Board: Sensor Firmware Version: Serial Port: Drive: FPS:

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>helloworld.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.



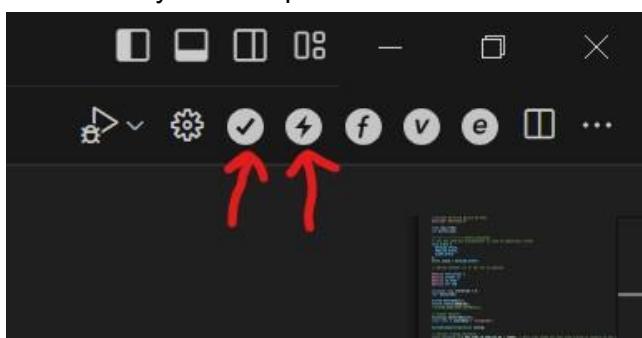
Boron

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.

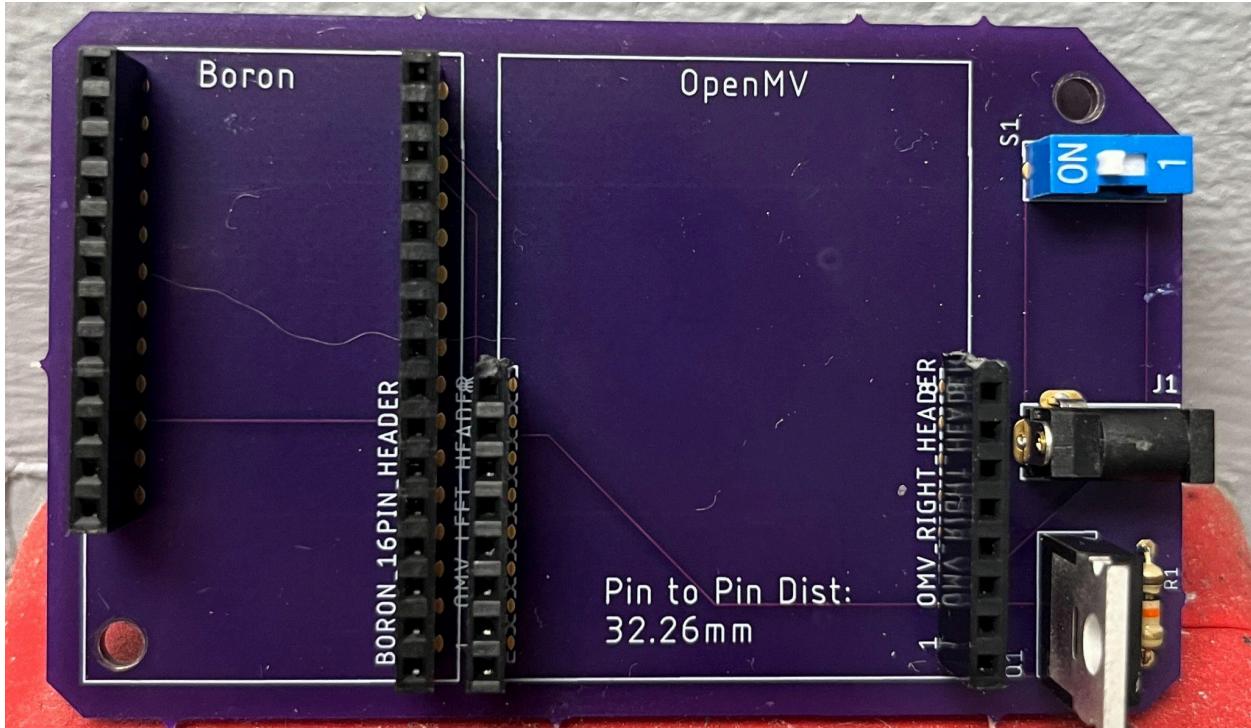
The screenshot shows the Visual Studio Code interface. The left sidebar displays the 'EXPLORER' view with the 'BORON' folder expanded. Inside 'BORON', there are subfolders '.vscode', 'src', and files '.gitignore', 'nul', 'project.properties', and 'README.md'. The 'src' folder is also expanded, showing 'TinyCamPublish.cpp' and 'TinyCamPublish.ino'. The 'TinyCamPublish.ino' file is currently selected. The main editor area shows the code for 'TinyCamPublish.ino'. The status bar at the bottom of the screen shows the target as 'Boron' and the device OS as 'deviceOS@6.1.0'.

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.



Mounting electronics

1. Solder on the female header pins onto the purple custom PCB board. Additionally, solder the slide switch, the n-channel MosFET, and the female barrel plug onto the board until it looks like this:



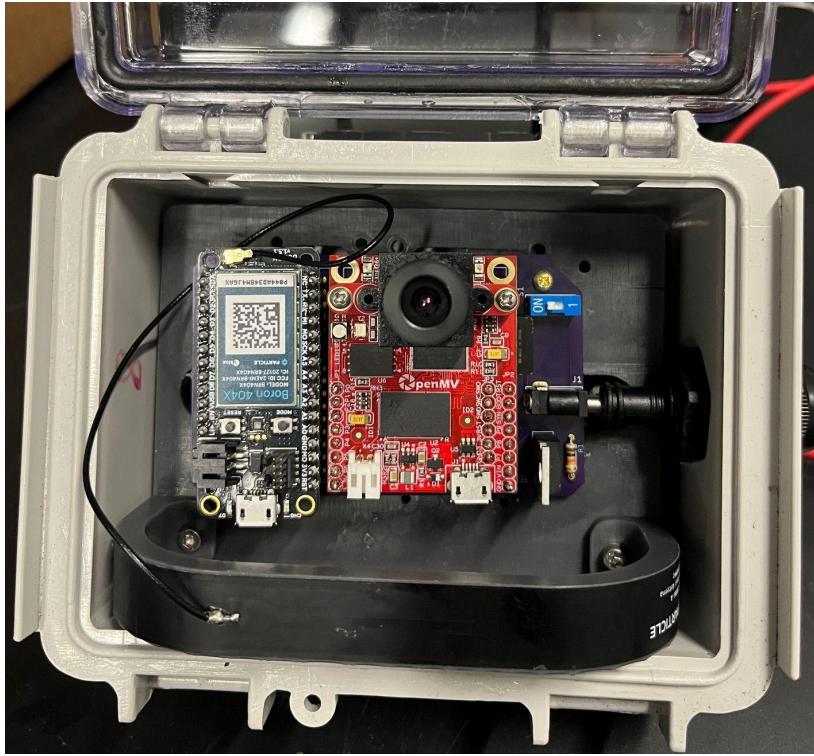
2. Attach the antenna mount to the baseplate using two 4-40 3/8th in screws and two 4-40 nuts. Peel the adhesive off the antenna and stick it onto the antenna mount.
3. Use the 2 4-40 3/8ths screws and 2 4-40 nuts to secure the PCB board onto the baseplate.



4. Secure the baseplate to the inside of the polycase using two M4 x 0.70 x 8 mm pan head screws. You may find it easier to secure the left screw down before the above step 3.



5. Plug the male barrel plug from the solar panel into the female barrel plug on the PCB board. And plug the Boron and the OpenMV into the female headers on the PCB board. Attach the end of the cellular antenna to the Boron. ****Note: you should not plug a micro USB into the Boron at the same time that the slide switch is “On” and receiving power from the barrel plug.**



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.

- 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

****Note: you should not plug a micro USB into the Boron at the same time that the slide switch is “On” and receiving power from the barrel plug.**

- 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.
 - 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.
- 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.
- 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).
- 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).
 - 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.
- Neaten up the loose solar panel wires and secure the extender to the pole with zip ties
- 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.
- 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.