Accurate VO2 Max for Zwift and Strava

by rabbitcreek

Human bodies are engines that utilize oxygen to burn fuel. Humans are not built with gauges and checking to see how the human engine is running is usually based on things we find easy to measure--pulse, Blood pressure. These do not directly tell us how the engine is burning its fuel. Measuring the amount of oxygen used with exertion provides better information on how hard the bodies engine is working--the equivalent of looking at the gauges on your cars dashboard. My previous work on measuring the components of exercise physiology were encumbering and not elegant enough to make them easily portable: https://www.instructables.com/Real-VO2Max-Measure... So after being introduced to the joys of **Zwift** bicycling with a **Wahoo** trainer I built a device that enables you to accurately add VO2 Max to the bluetooth information displayed on the **Zwift** screen animation while you are peddling along. The device also easily pairs with the **Strava** app for providing VO2 Max data for all your trail excursions. It substitutes VO2 max information for the heart rate sensor data in both of these popular exercise programs. The device is portable and lightweight with Wifi and Bluetooth capabilities and is easily worn with a modified 3M mask designed to be comfortable for long term use. I have tested the device for other sports including cross-country skiing and skating but its portability and wireless transmission capacity make it amenable to just about any sport other than swimming. The device also communicates with an App for the iPhone which enables graphing and long term data storage and download for ancillary calculations. The output includes continuous output of VO2, calories consumed, volumes of expired gas, as well as performing such functions as Basal Metabolic Rate. The device is easily made for about $100. Parts are all readily available and the case is 3D printed. It can be assembled in about 10 minutes. We tested the device in a physiology lab against a $60,000 machine and found it gave the about the same results.

Step 1: Gather Your Parts

There are only four main parts to the unit. A differential pressure sensor, an oxygen sensor--both connected by I2C to a TTGO Esp32 microcontroller with screen. A Lipo Battery with switch completes the unit. The original unit included a Laser CO2 sensor and an analogue version of the differential pressure sensor. The digital version of the sensor was found to be much more accurate and with a better range of 1--250 Pa.

1. TTGO T-Display ESP32 CP2104 WiFi bluetooth Module 1.14 Inch LCD Development Board $11

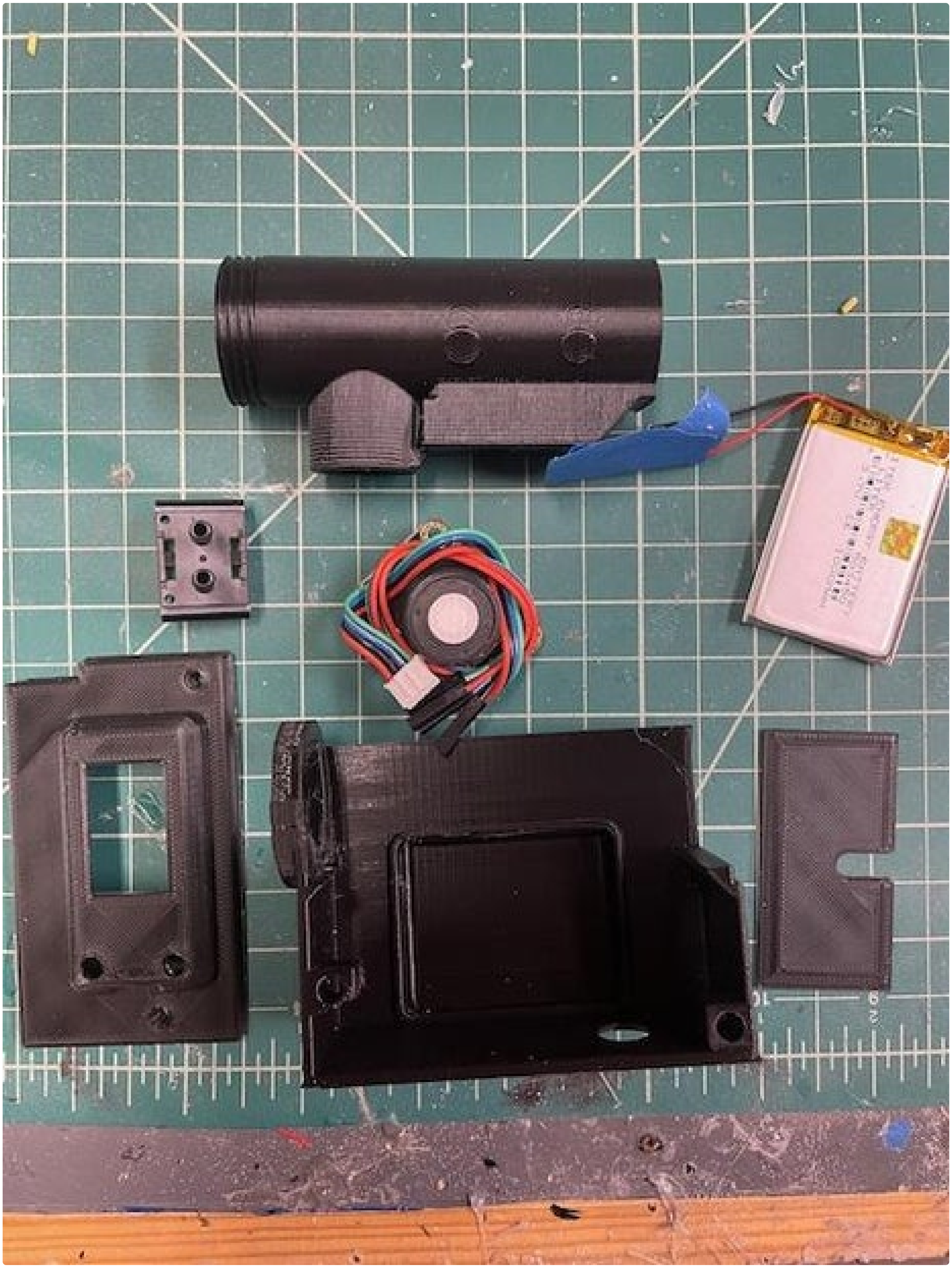
2. Omron--D6F-PH0025AD2--$40 from Digikey

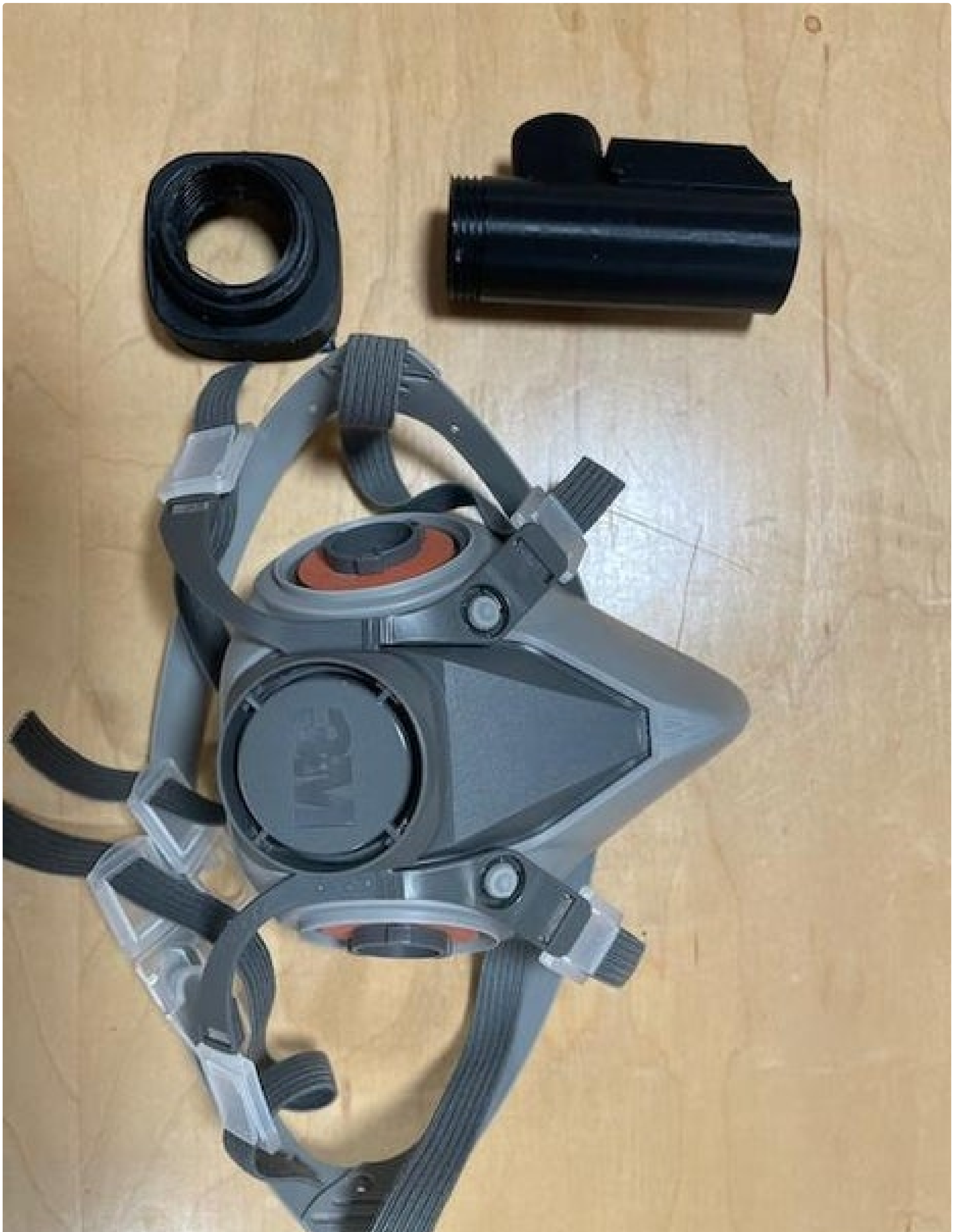
3. Gravity: I2C Oxygen Sensor--$50 from DF Robot

4. Lipo Battery -- $5 1000 Mah

5. Switch ON/OFF ---$1

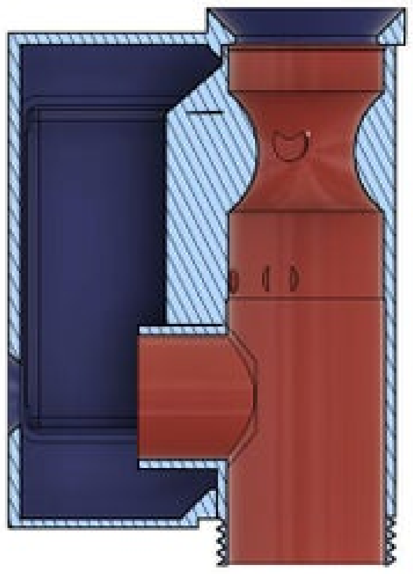
6. PARTICULATE RESPIRATR MASK -- 3M $20 Digikey (you don't really care about the filtration on this mask they are removed)

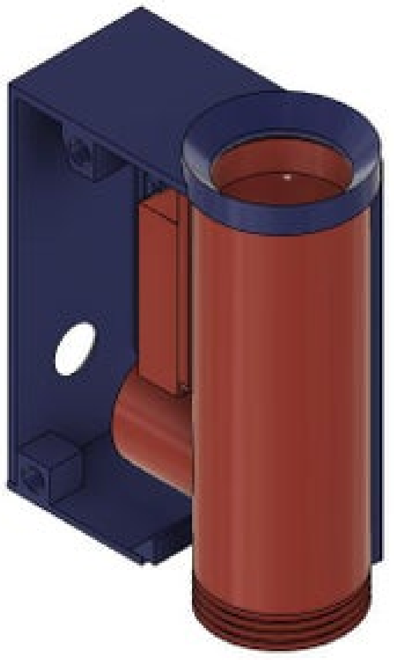


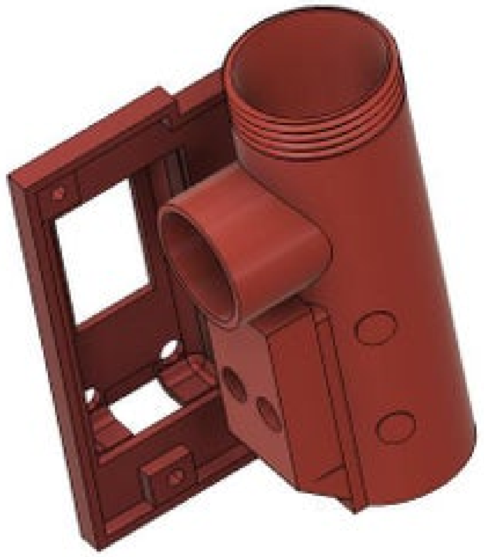


Step 2: 3D Print Your Parts

All parts are printed in PLA. All files are included. No support was used except for the computer housing. The body of the unit consists of three parts--the most important being the venturi tube with ports for both the differential pressure sensor and the oxygen sensor. The measurements of the internal structure of the throat are carefully laid out and measured and cannot be changed or it will drastically effect the results of the output. The venturi tube is nested into the body enclosure. The third part is the access door/ computer housing which is held on by two 3mm screws. A small retaining shield is also included to seal off the computer and make assembly easier.







https://www.instructables.com/ORIG/F6O/KXFY/KY5UAQP7/F6OKXFYKY5UAQP7._3mf Download

https://www.instructables.com/ORIG/F5I/UIMW/KY5UAQP8/F5IUIMWKY5UAQP8._3mf Download

https://www.instructables.com/ORIG/FP9/XRQO/KY5UAQP9/FP9XRQOKY5UAQP9._3mf Download

https://www.instructables.com/ORIG/F20/24J8/KY5UAQPK/F2024J8KY5UAQPK._3mf Download

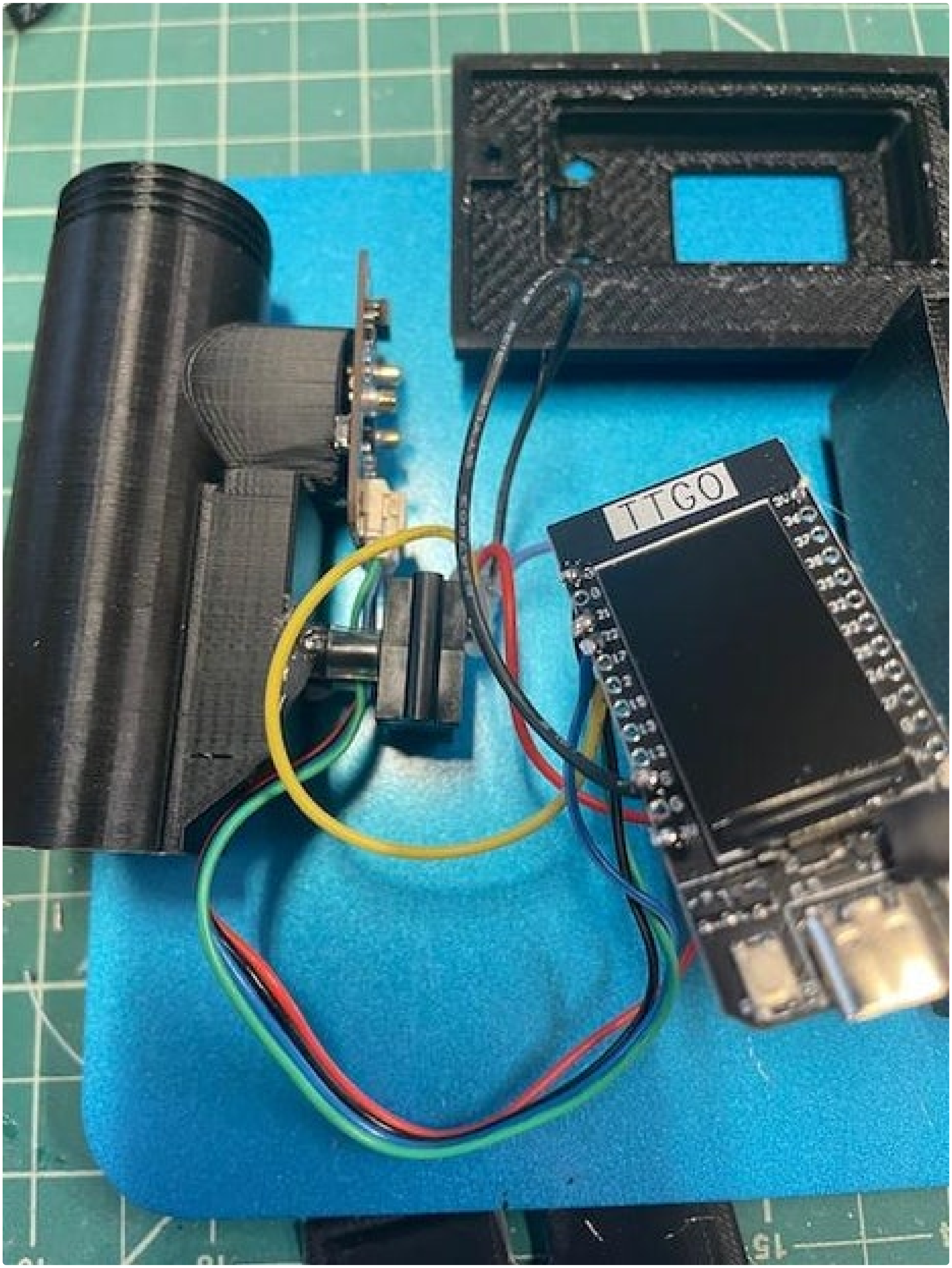
https://www.instructables.com/ORIG/F8W/AO96/KY5UAQPL/F8WAO96KY5UAQPL._3mf Download

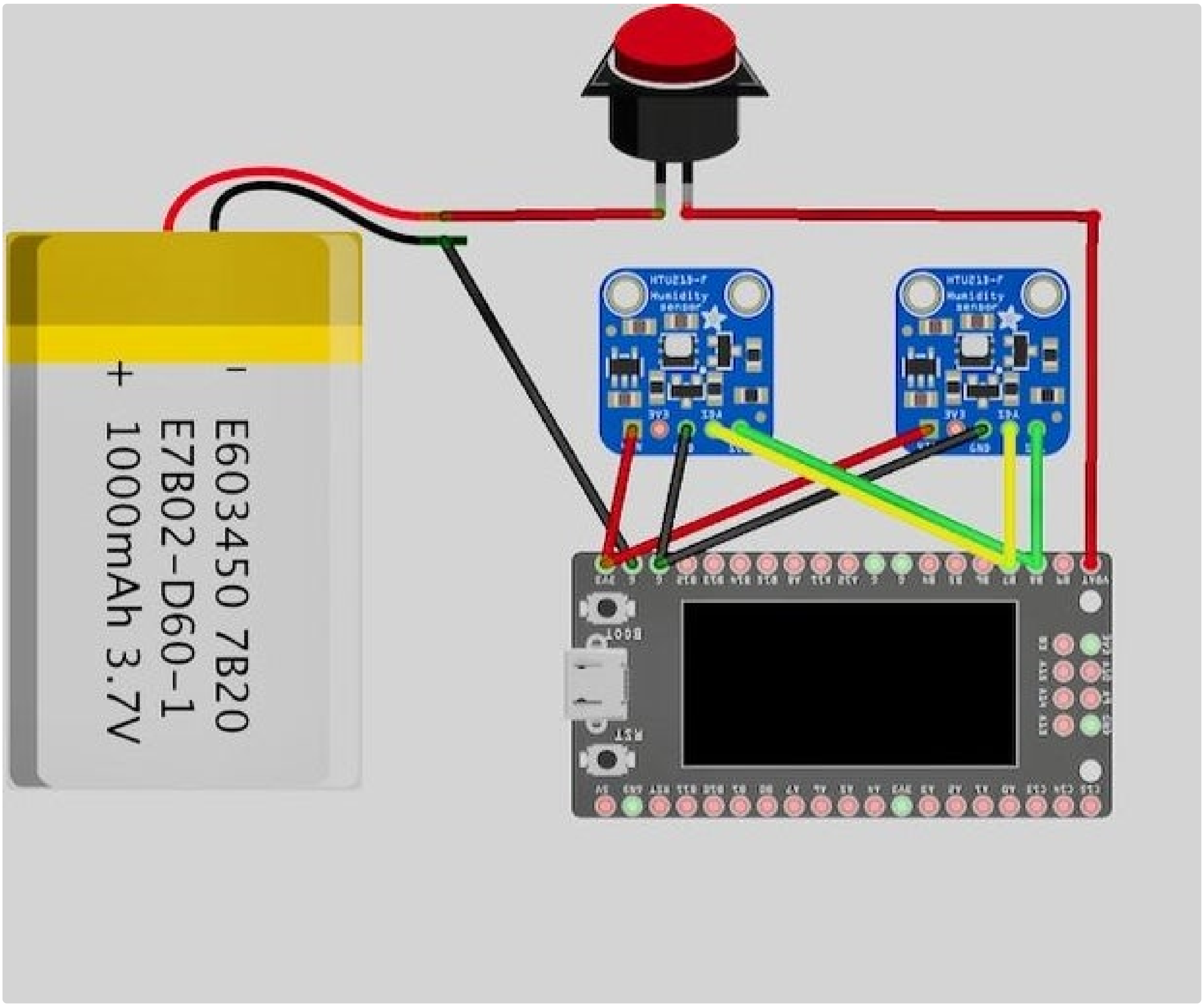
https://www.instructables.com/ORIG/F6D/O38R/KY5UAQPH/F6DO38RKY5UAQPH._3mf Download

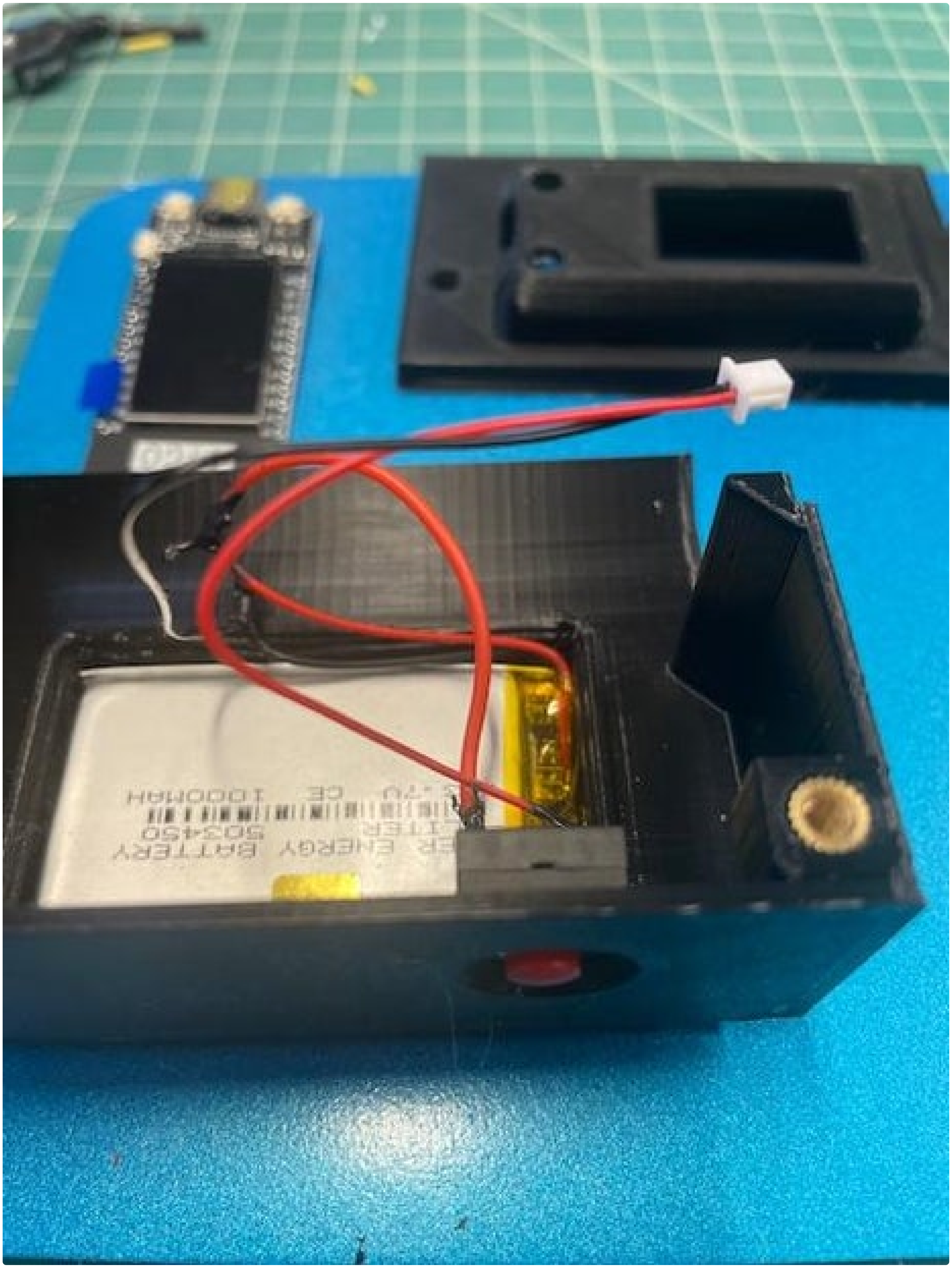
https://www.instructables.com/ORIG/F84/AIJ5/KY5UAQPG/F84AIJ5KY5UAQPG._3mf Download

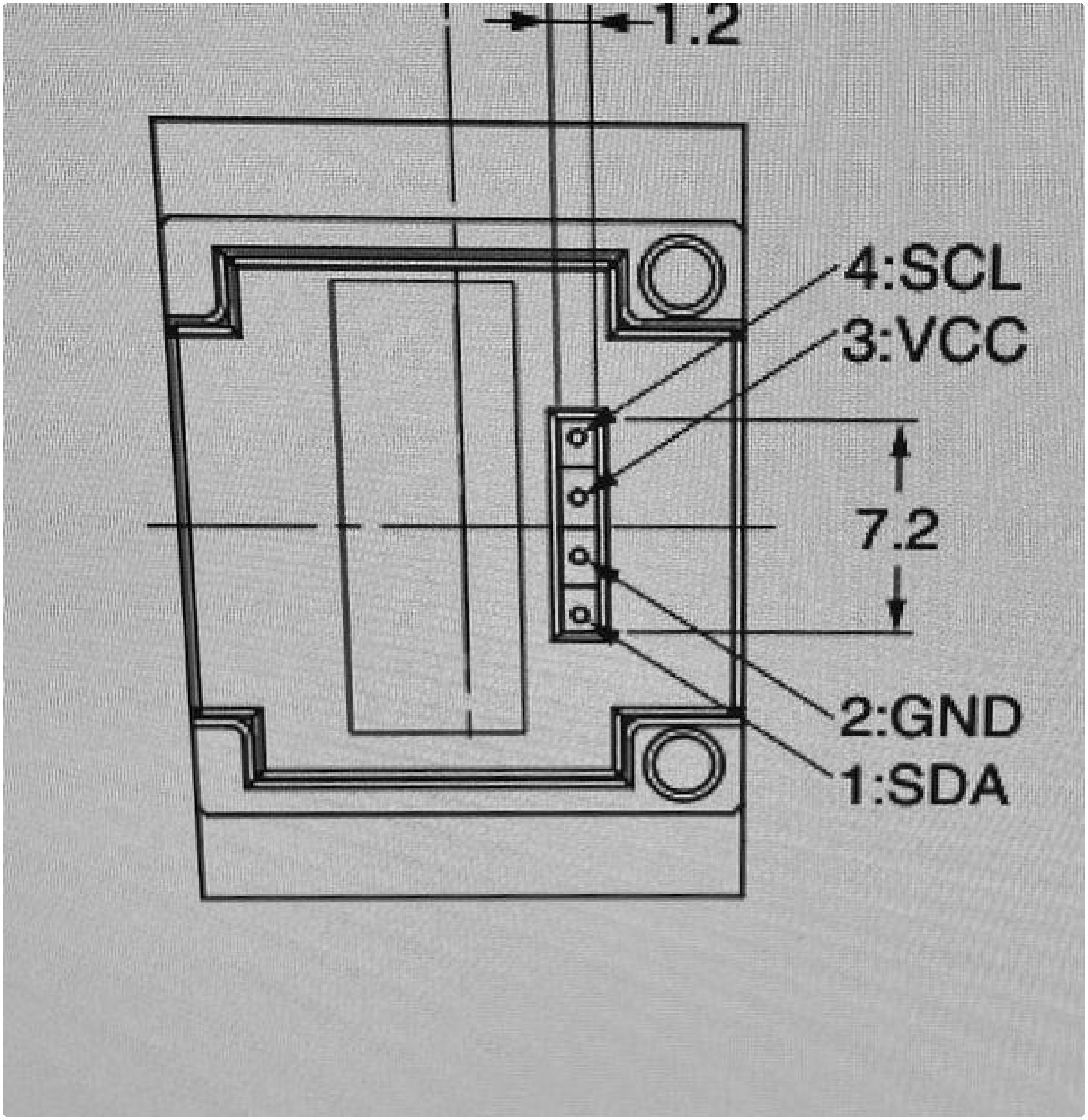
Step 3: Wire It

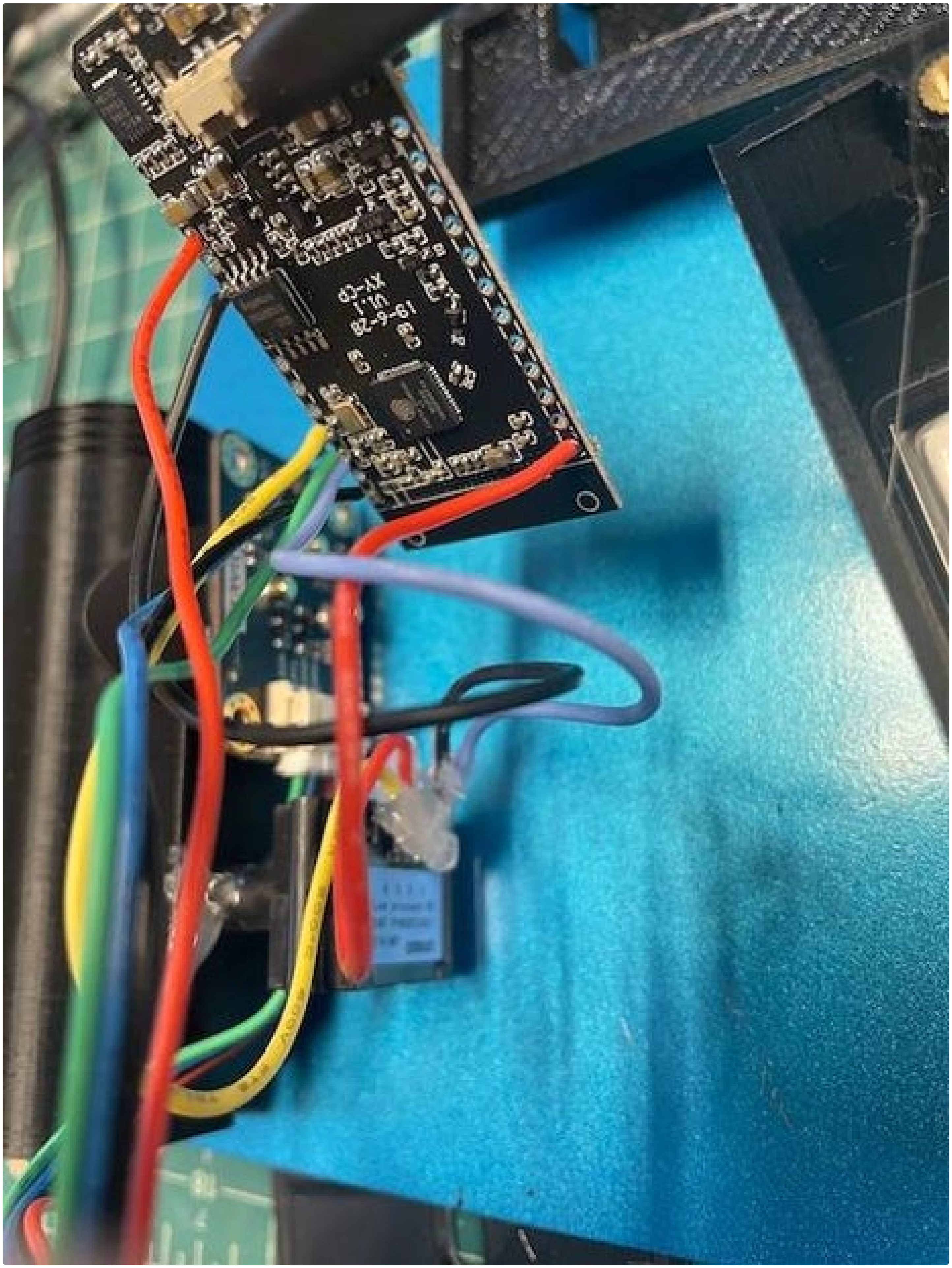
The actual wiring for this project is minimal. It consists of connecting two I2C devices to the computer and supplying them with power and ground. They each have different I2C addresses and the pull-up resistors are included in the DF Robot O2 sensor. The O2 sensor is carefully marked for which wires go where, however, pay carefully attention to the wiring diagram included above for the Omron sensor before wiring. (Other Omron sensors of the same type have different wiring patterns!) Both of the sensors take 3 Volts which is obtained off of the TTGO board. This ESP32 board has I2C inputs on pins 21(SDA) and 22(SCL). Multiple options on the board are available for G and Power (3V). The battery supply for voltage is delivered to the small battery connector on the back of the board interrupted by a simple on/off switch. To enable charging you must have the button in the on position and provide power through the USB-C connector. I used a 1000Mah battery which easily powers the unit for several hours. The computer body is carefully designed to fit the battery snugly and you might want to measure it before ordering--there is some variation in size.

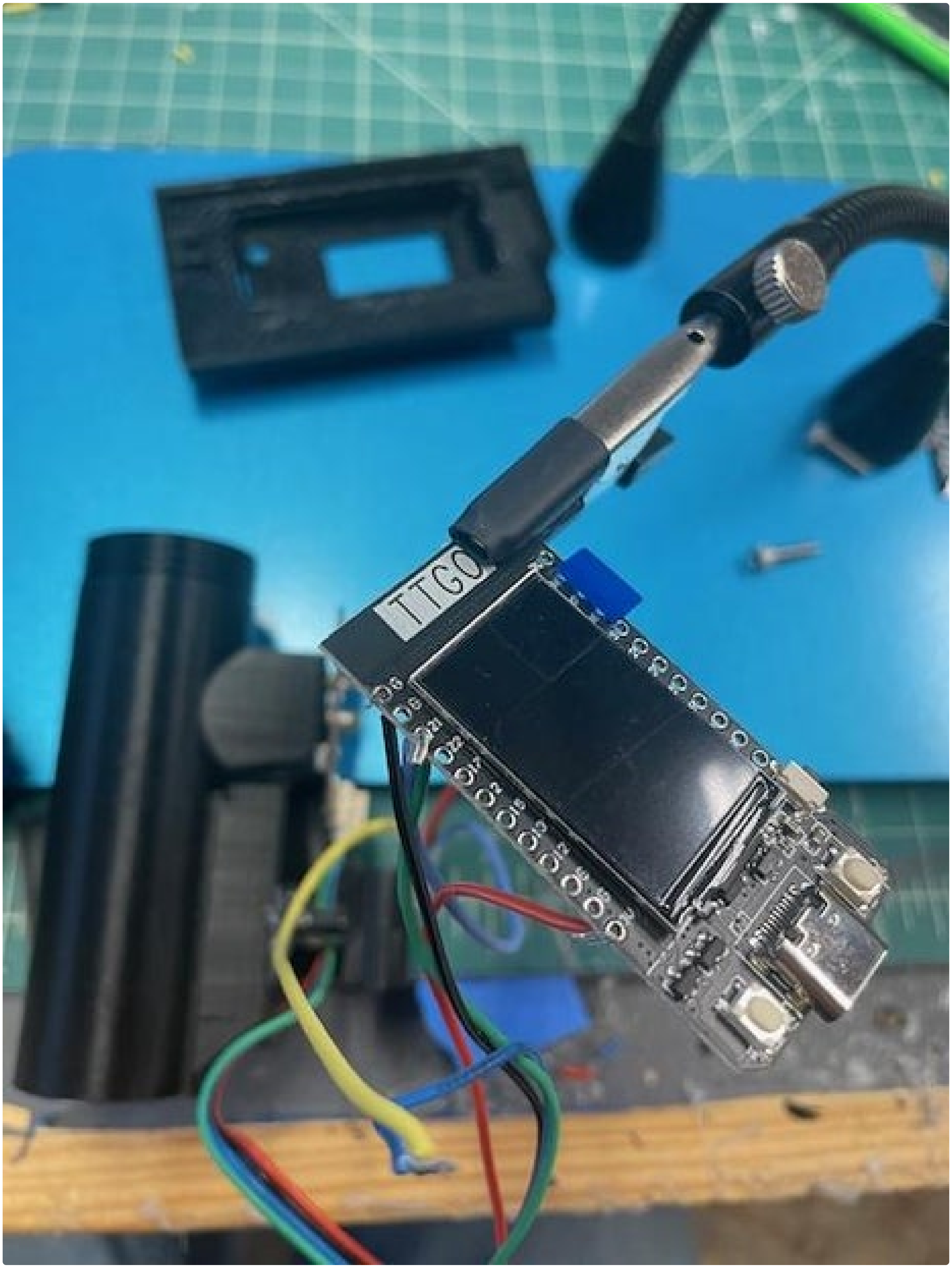

TTGO


E603450 7B20
E7B02-D60-1
1000mAh 3.7V
+
-
HTU21D-F
Humidity sensor
HTU21D-F
Humidity sensor
BOOT
RST




1.2
4:SCL
3:VCC
7.2
2:GND
1:SDA

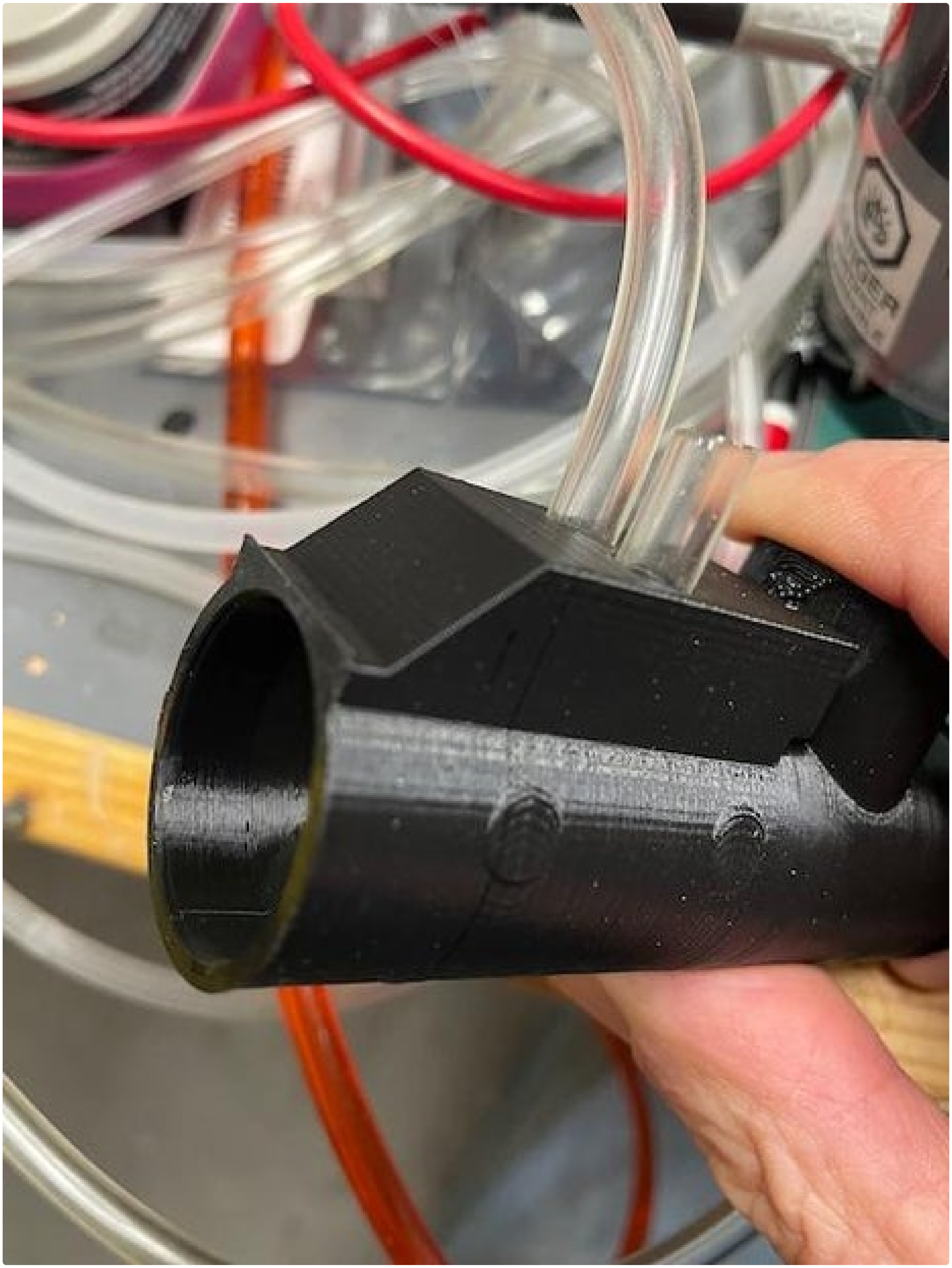


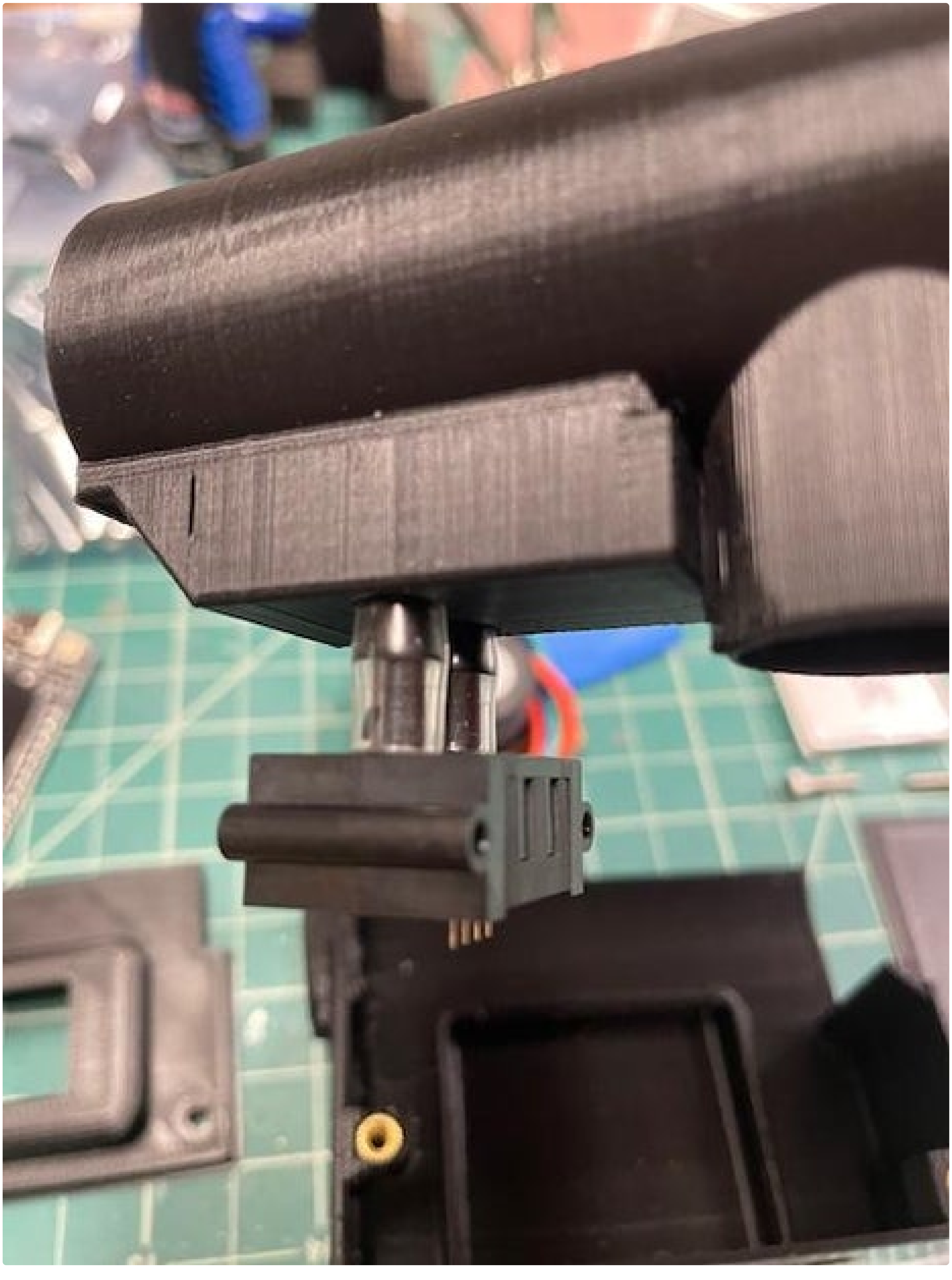

TTGO

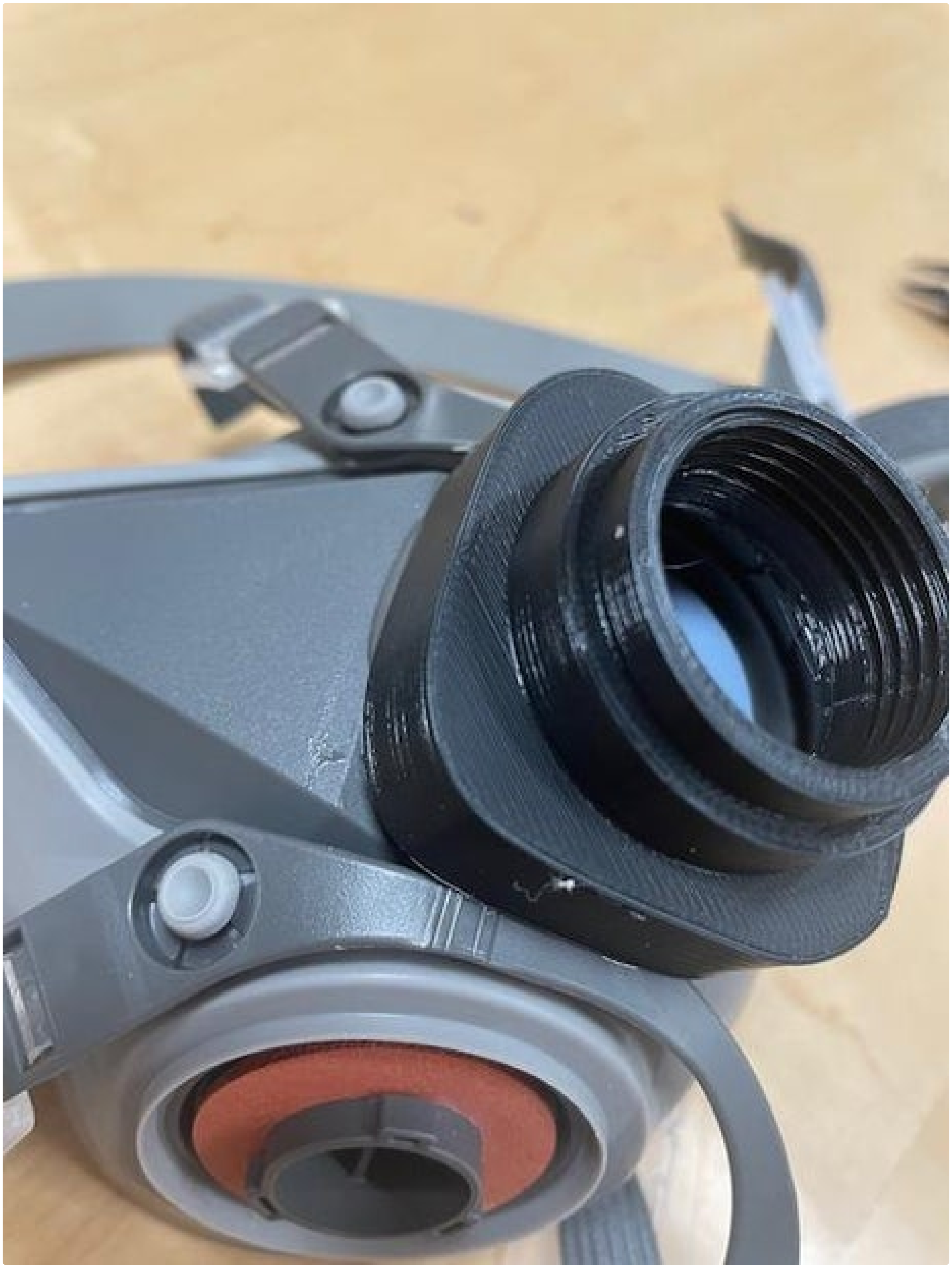
https://www.instructables.com/ORIG/FER/YQQY/KY5UAQPA/FERYQQYKY5UAQPA.cpp Download

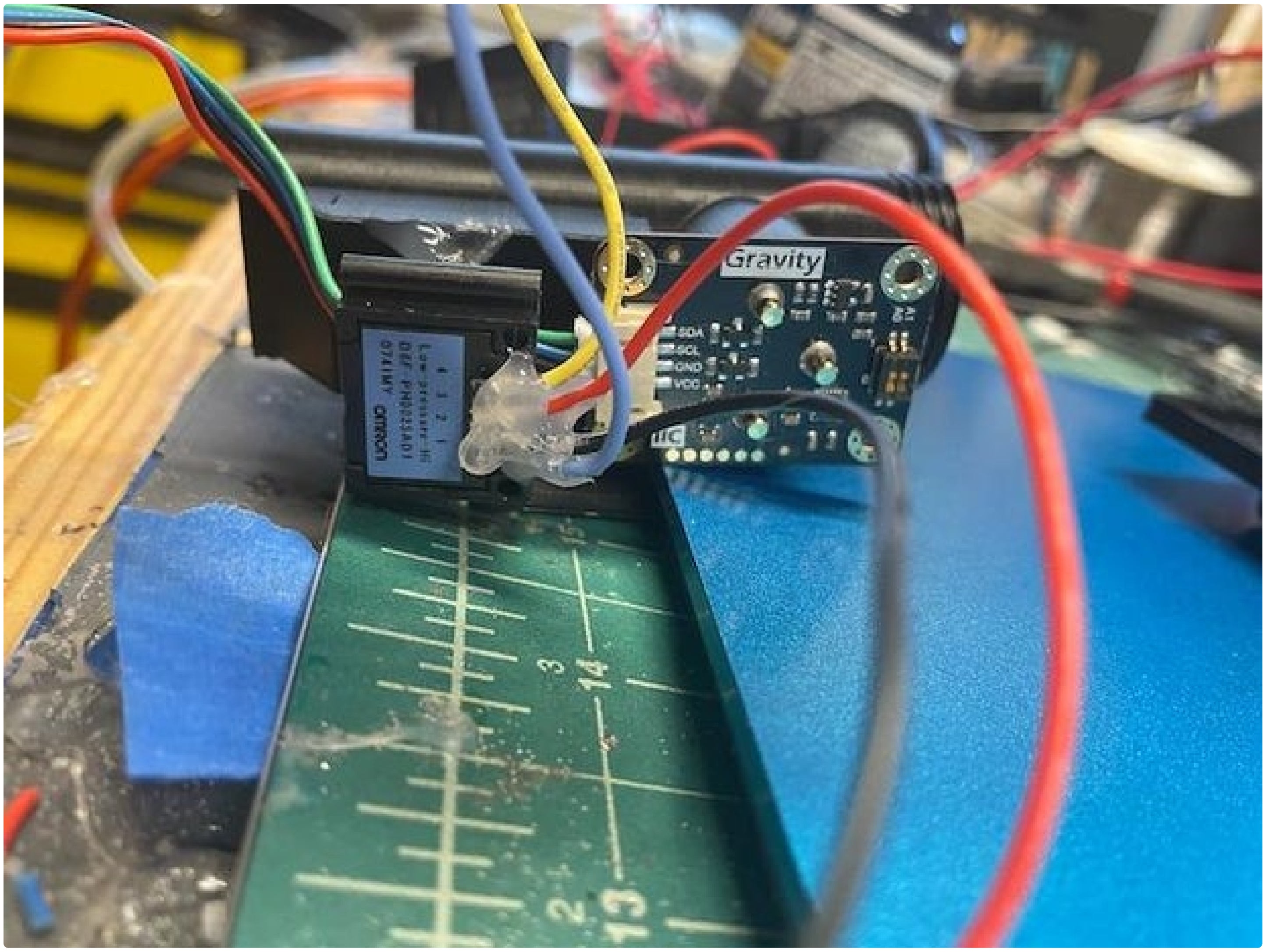
Step 4: Build It

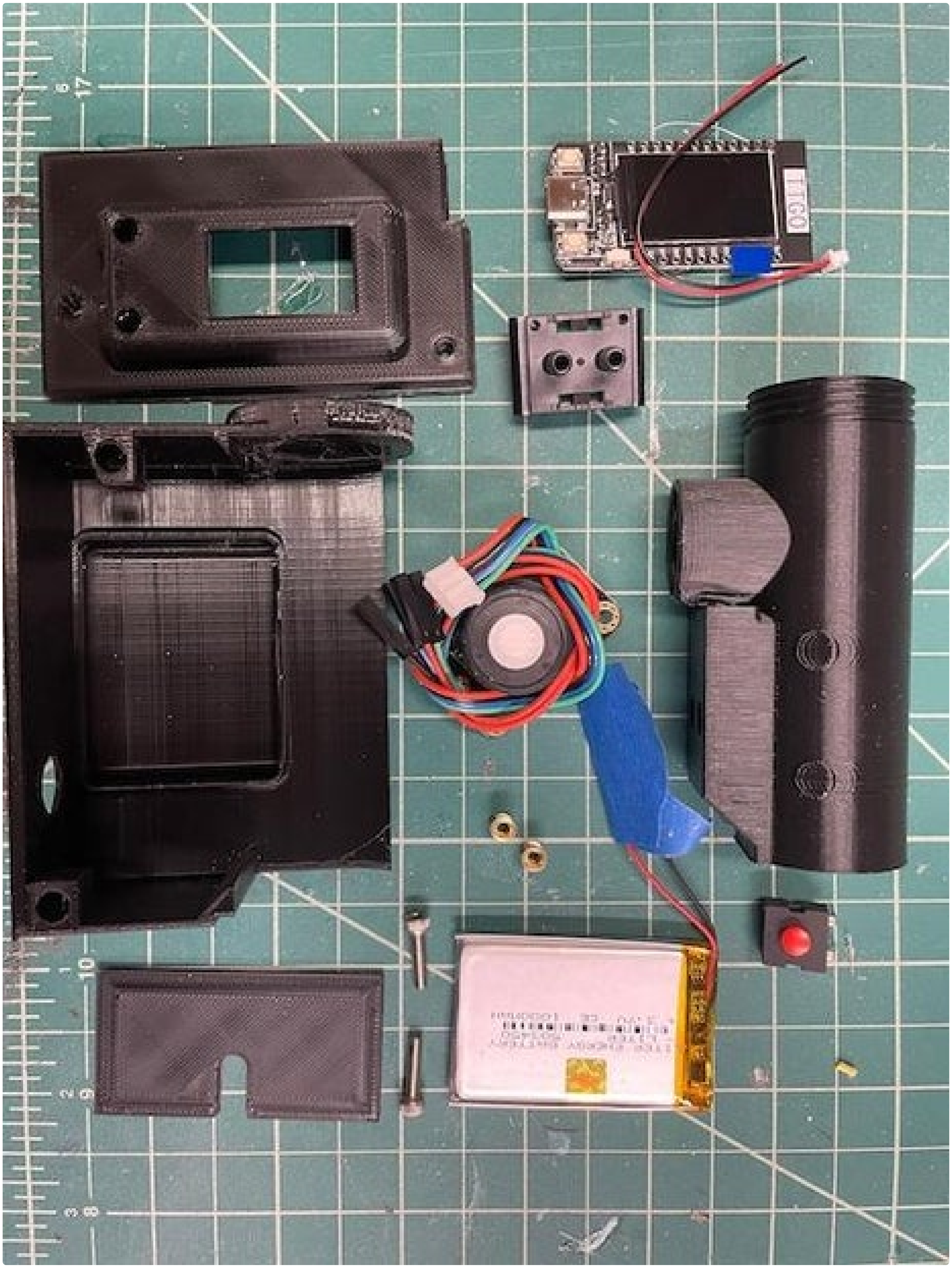
The only unique part of the assembly is mounting of the Omron sensor. Two passages located in the venturi tube connect the sensor with the 3D printed ports within the venturi tube. These were built to accommodate two pieces of standard aquarium tubing with an inside diameter of 4mm. This tubing is designed to have a very snug fit into these holes. The tubes sink to a predetermined depth and are then trimmed to accommodate the two fluted ends of the sensor. Please see detailed pictures to clarify the description. There is also a polarity to these input tubes to the sensor--the sensor must be mounted per the photo to have a low and high pressure sensor input to be correct. A bead of hot glue on the tubes will hold them in position. The Oxygen sensor tube friction fits into its holder in the venturi tube--you might want to put a bead of hot glue on the unit to keep it in position. A defect of this sensor allows the chemical unit to separate very easily from the digital measuring unit. This is done on purpose as the O2 sensor head--like all O2 sensors must be replaced about every two years. Threaded inserts for 3 mm screws are heat inserted into the 3D printed holes in the body of the unit. The battery is placed into position in the wall of the main body. The On/Off switch is superglued into the hole in the bottom of the main body. When the two sensors are mounted into correct position in the venturi tube it is hot glued into the main body as shown. The TTGO board with attached wiring is hot glued into position in the side wall. Make sure the buttons are visible through their respective holes and the USB-C connector is open in the hole. The cover is placed over the wiring and sealed with hot glue. The wires are carefully organized and the case closed with two 3mm screws. The button housings are superglued over their respective holes after inserting the flanged buttons. The 3M mask is modified by removing the filters from the side holes which are input. The output valve at the center of the mask has a small plastic cover over it. This is clipped off and removed. The adapter unit that you printed is then glued to the front of the mask.

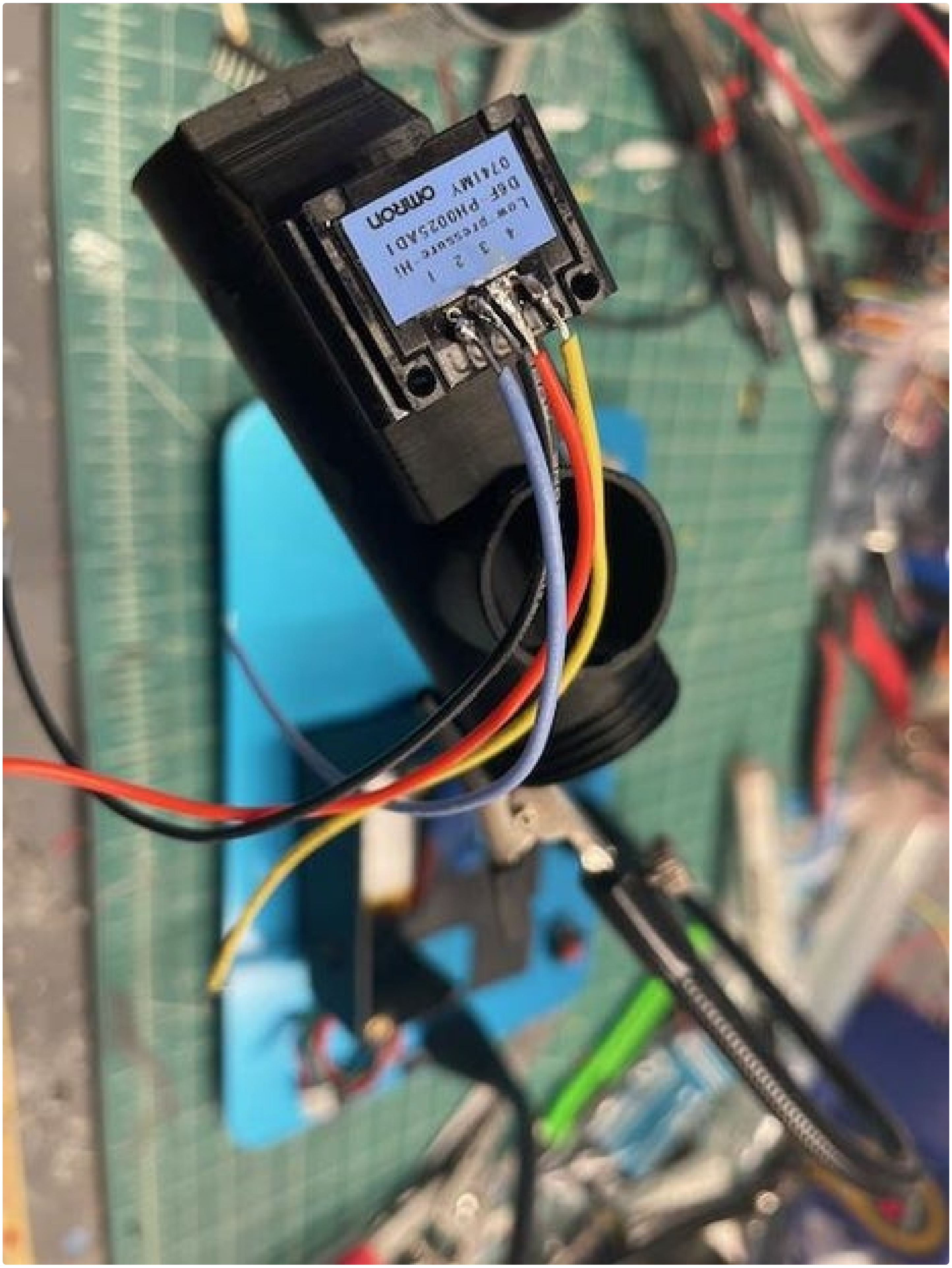


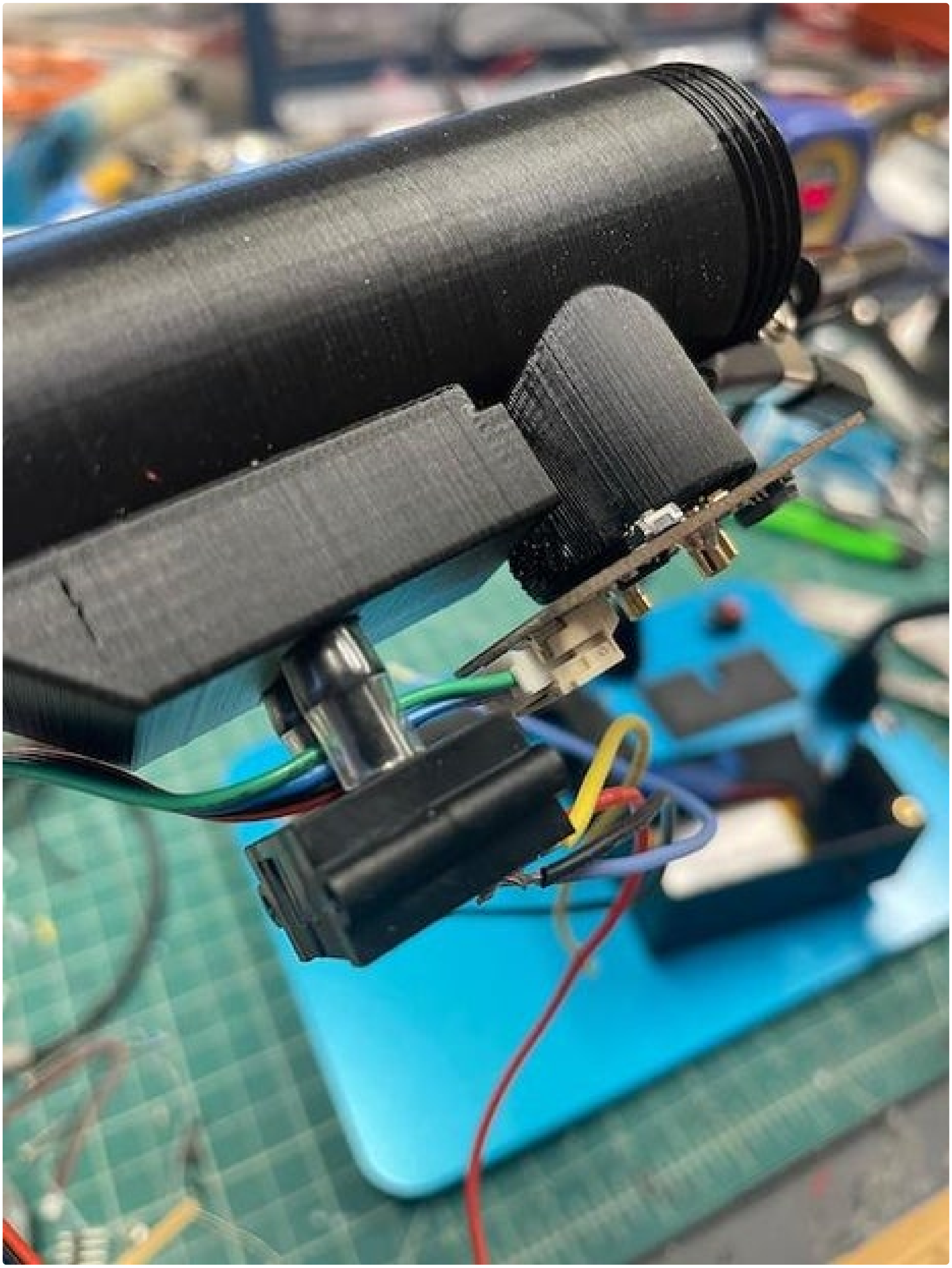


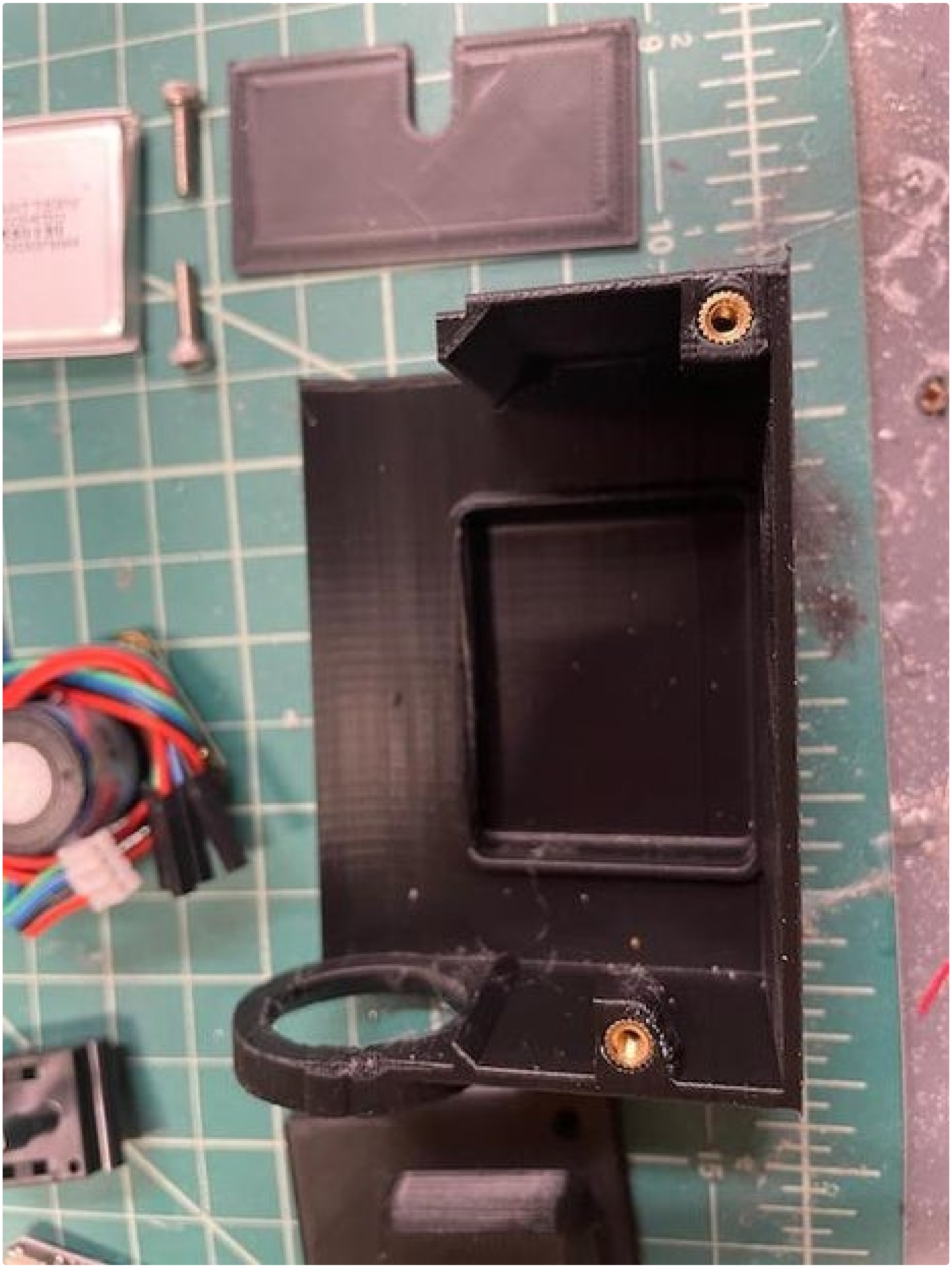


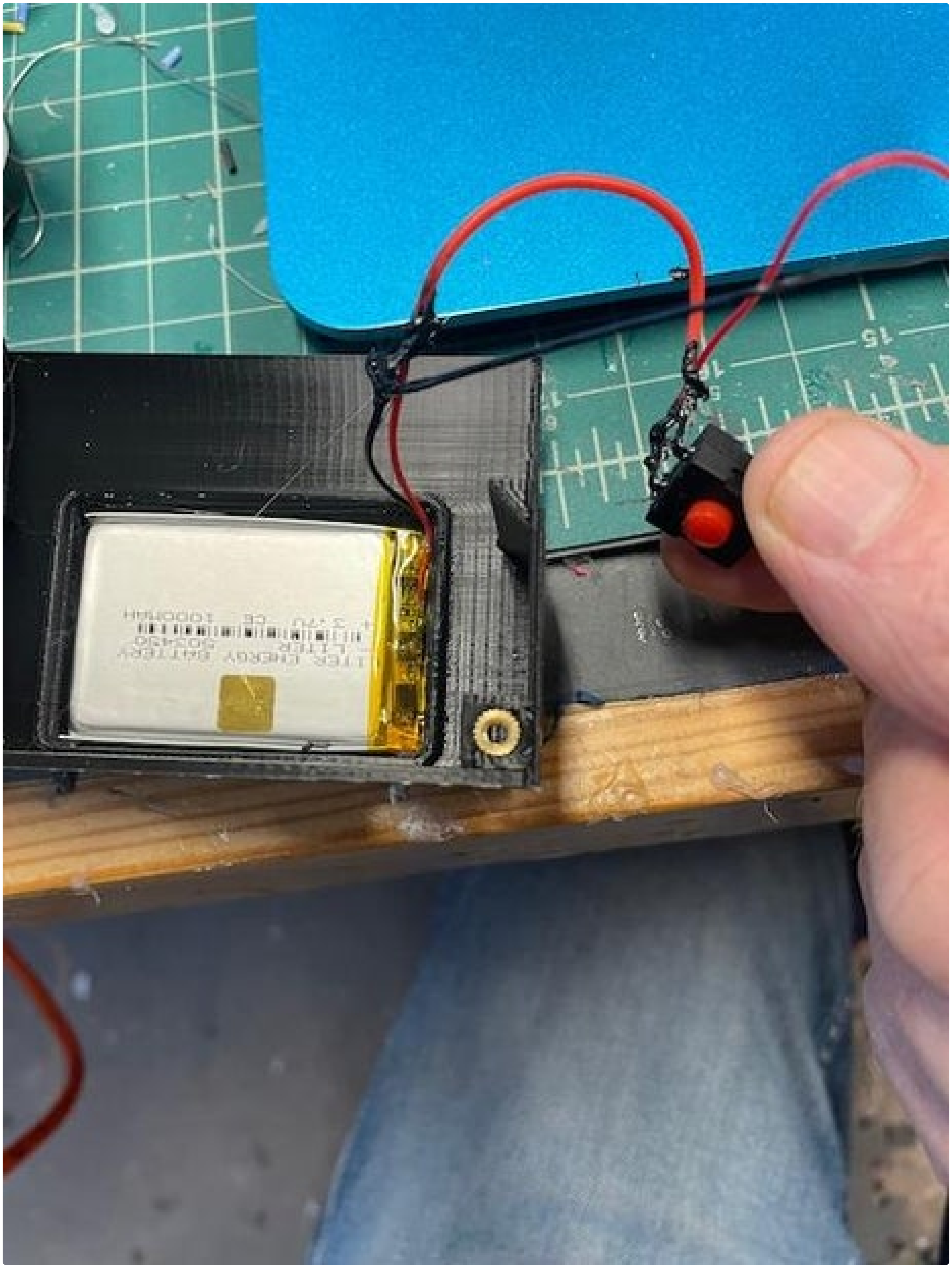

Gravity
SDA
SCL
GND
VCC
IIC
omron

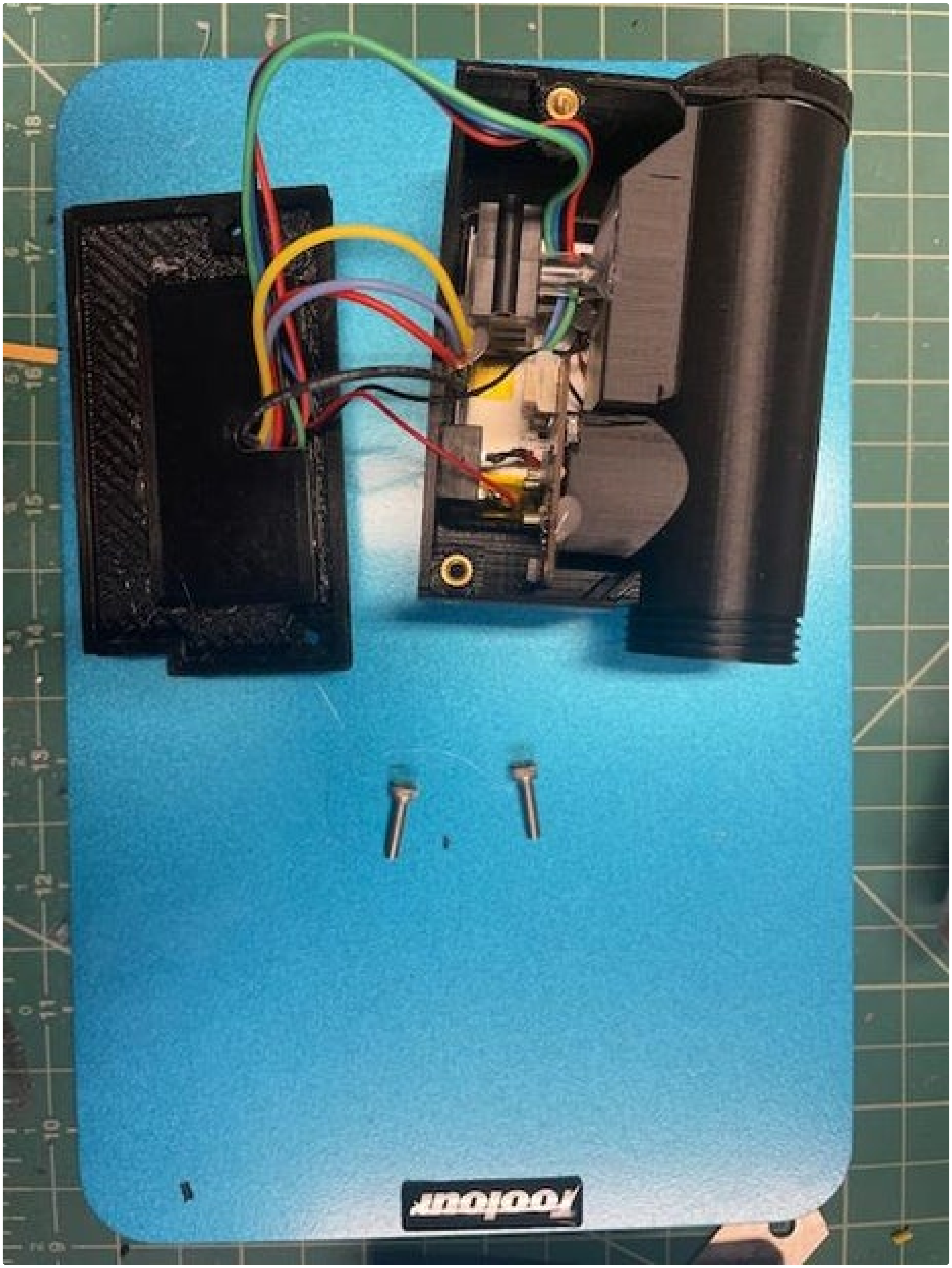


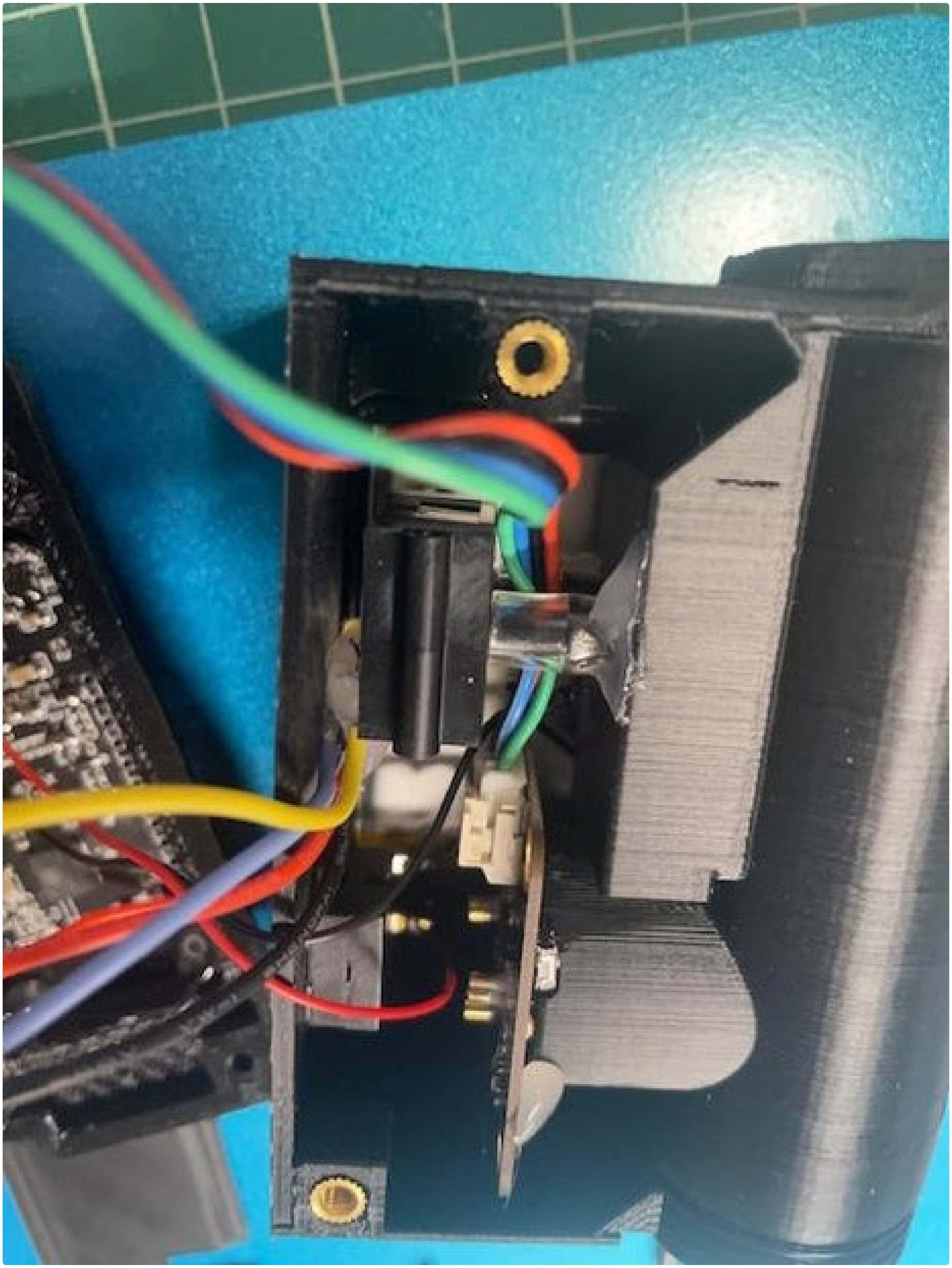

omron
0741MY
D6F PH0025AD1
Low pressure Hi
4 3 2 1

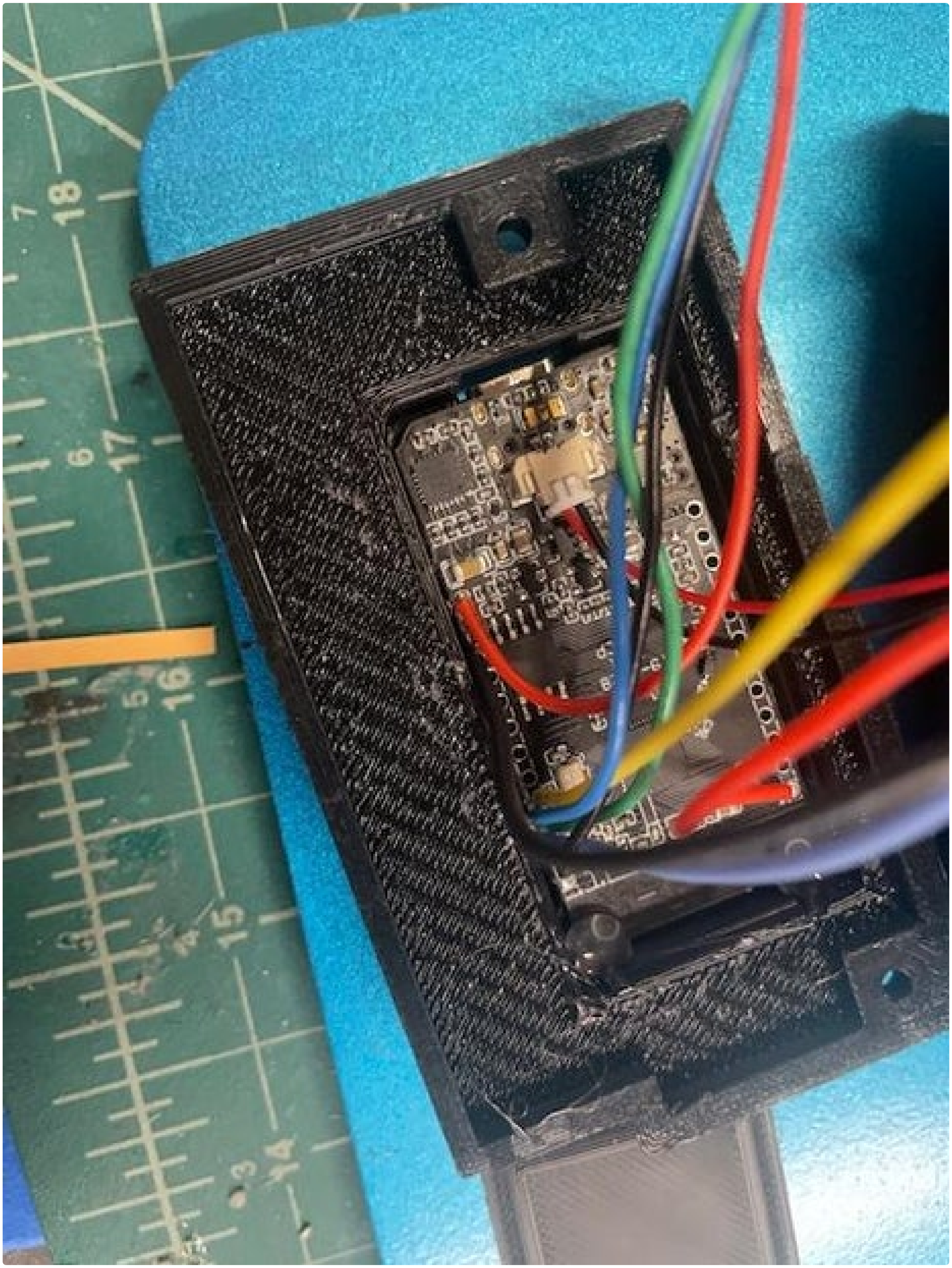








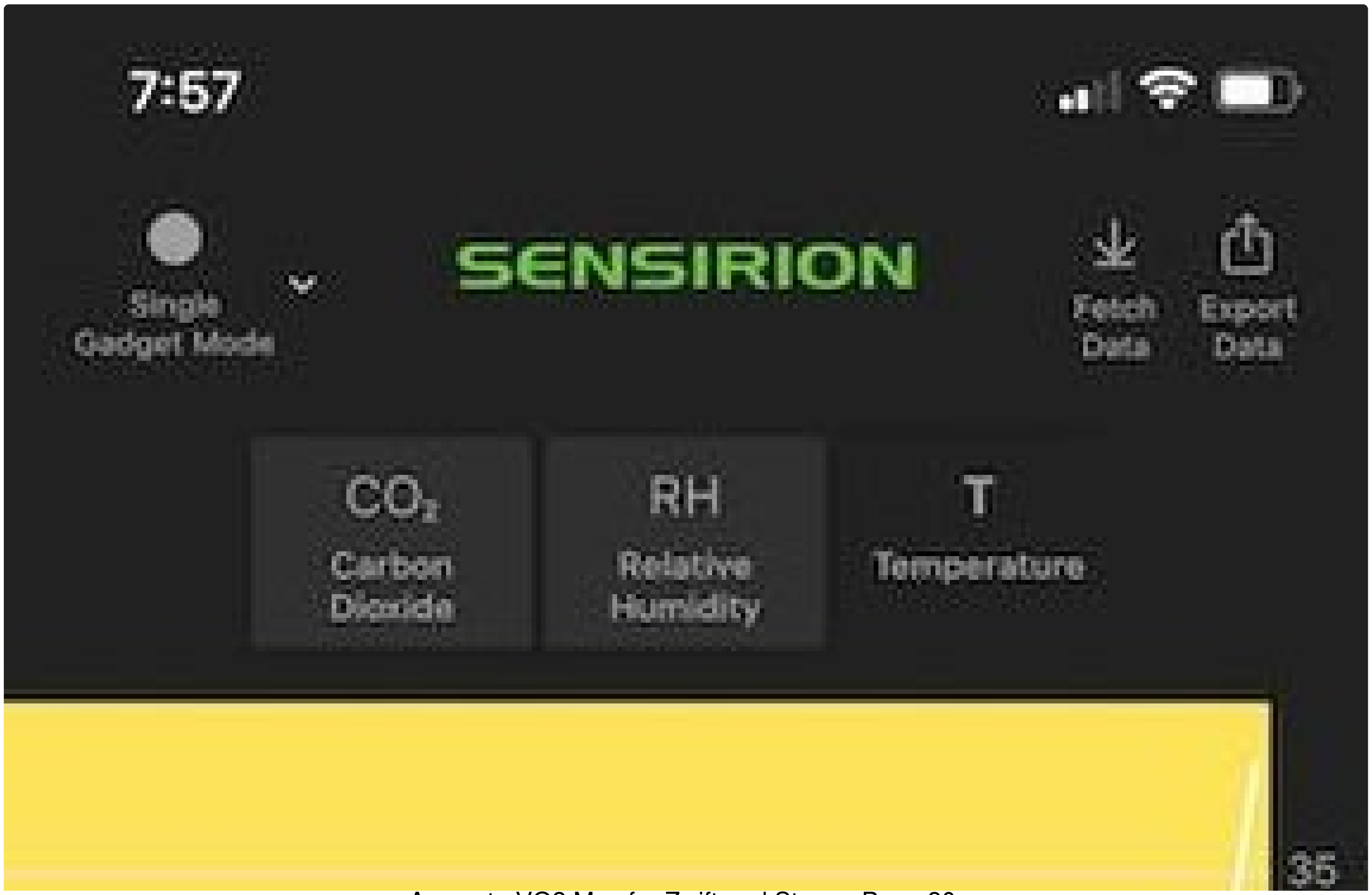


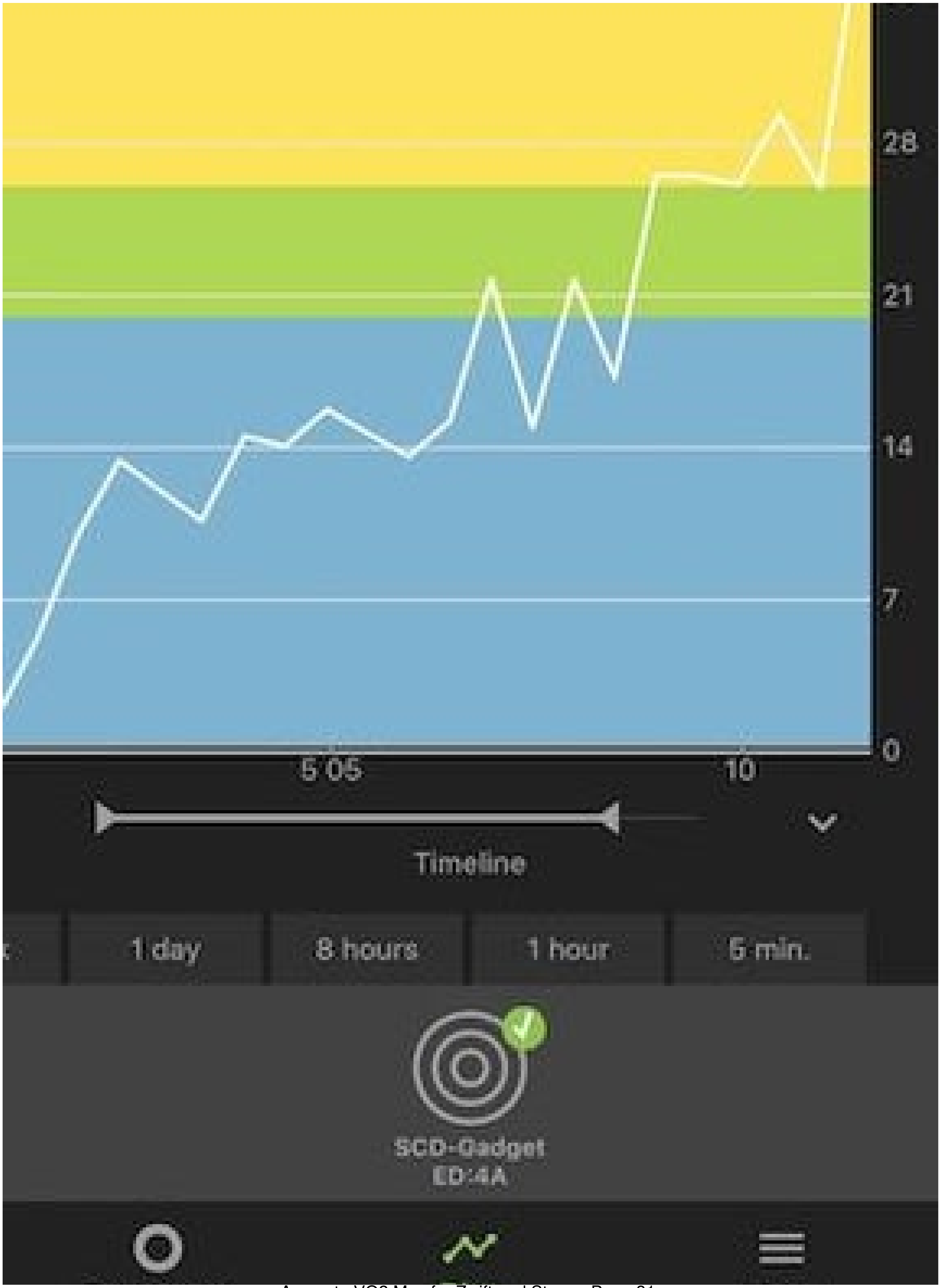


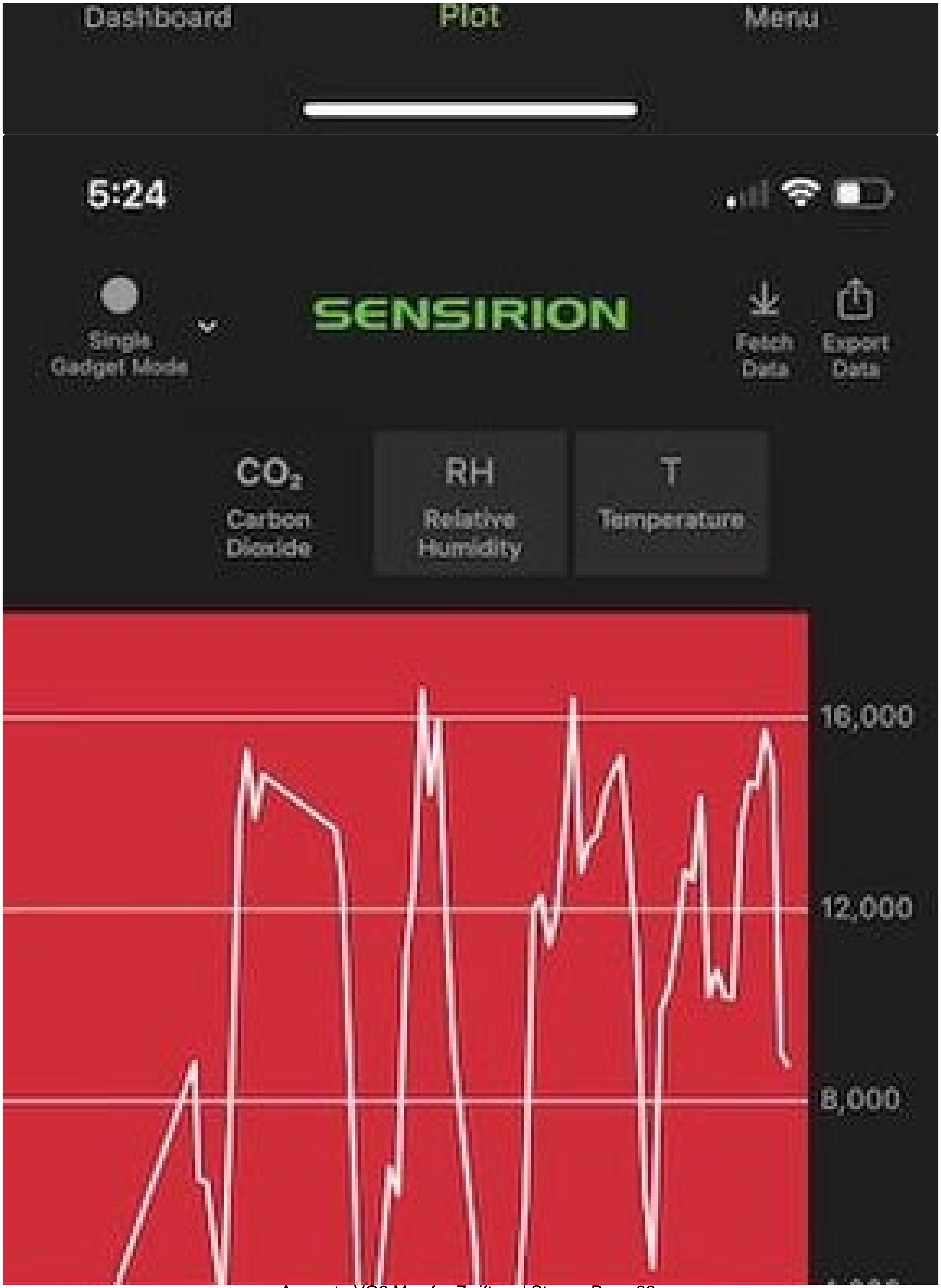
Step 5: Program It

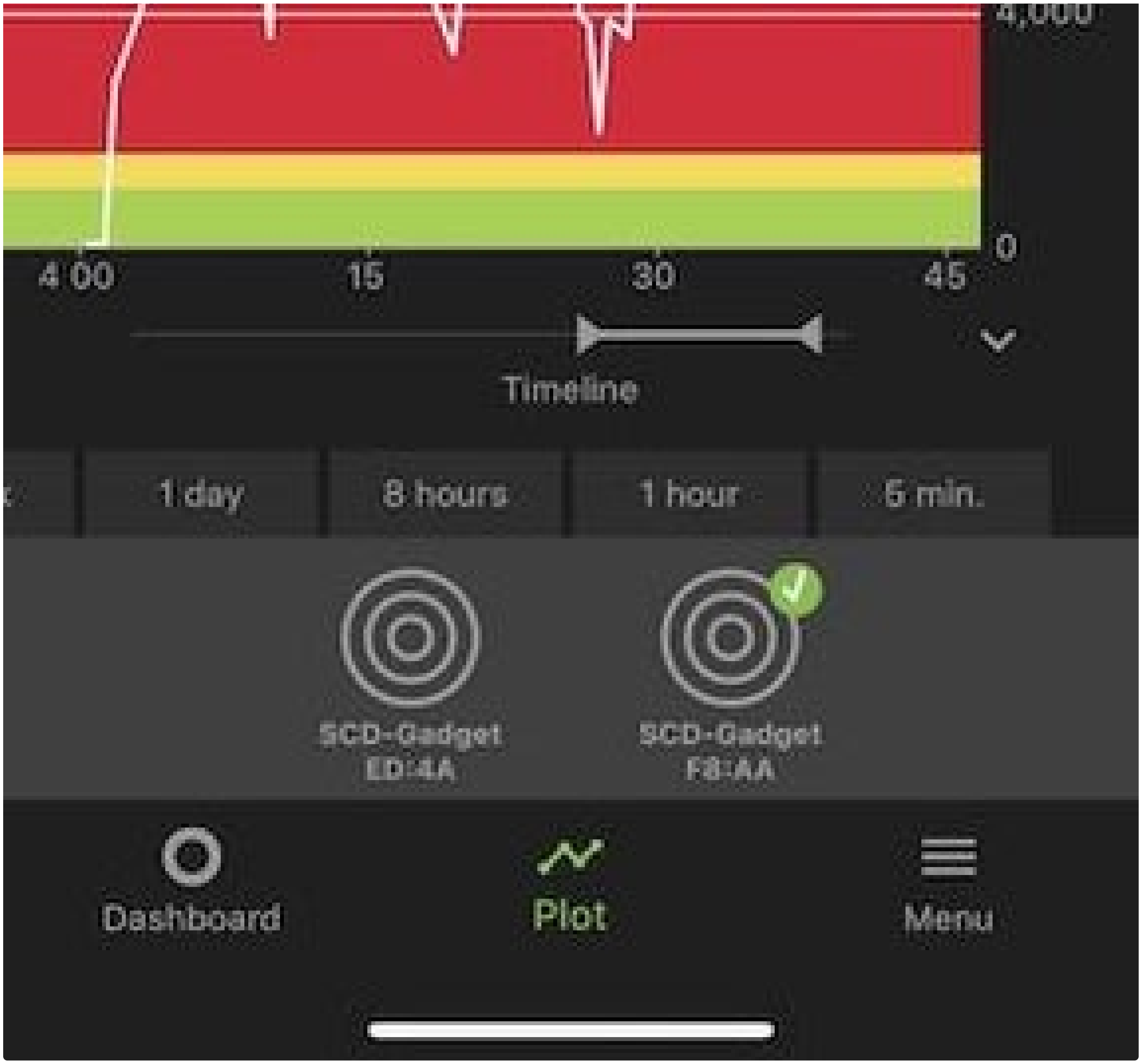
There are two programs that run the unit. Both are Arduino IDE based. FinalZwiftConnect with files DFRobot_OxygenSensor.cpp and DFRobot_OxygenSensor.h are used when the unit is broadcasting VO2 max to either Zwift or Strava. This is based on Andress Spiess work in YouTube #174. After loading, the initial screen will request the users wt in pounds. Another screen with Zwift will appear and the button with "go" will initiate the session. The loop function checks the Omron sensor for a pressure drop and initiates a time function to calculate the total volume of air moving past the sensor using Bernoulli's equation. Every five seconds if there is no breath the seeO function is called to check the O2 level. At thirty second intervals the goFigure function calculates the minute volume of expired O2 and calculated levels of CO2 and VO2 Max. The program sends the bluetooth characteristic to the Zwift receiver on either your Mac or Apple TV masqueraded as the Heart Rate which approximates a typical VO2 max level. The ESP32 screen then presents alternatively Time, VO2, VO2Max, Cal burned, Max Cal/Day, O2 level and Volume(Liters of O2 used).

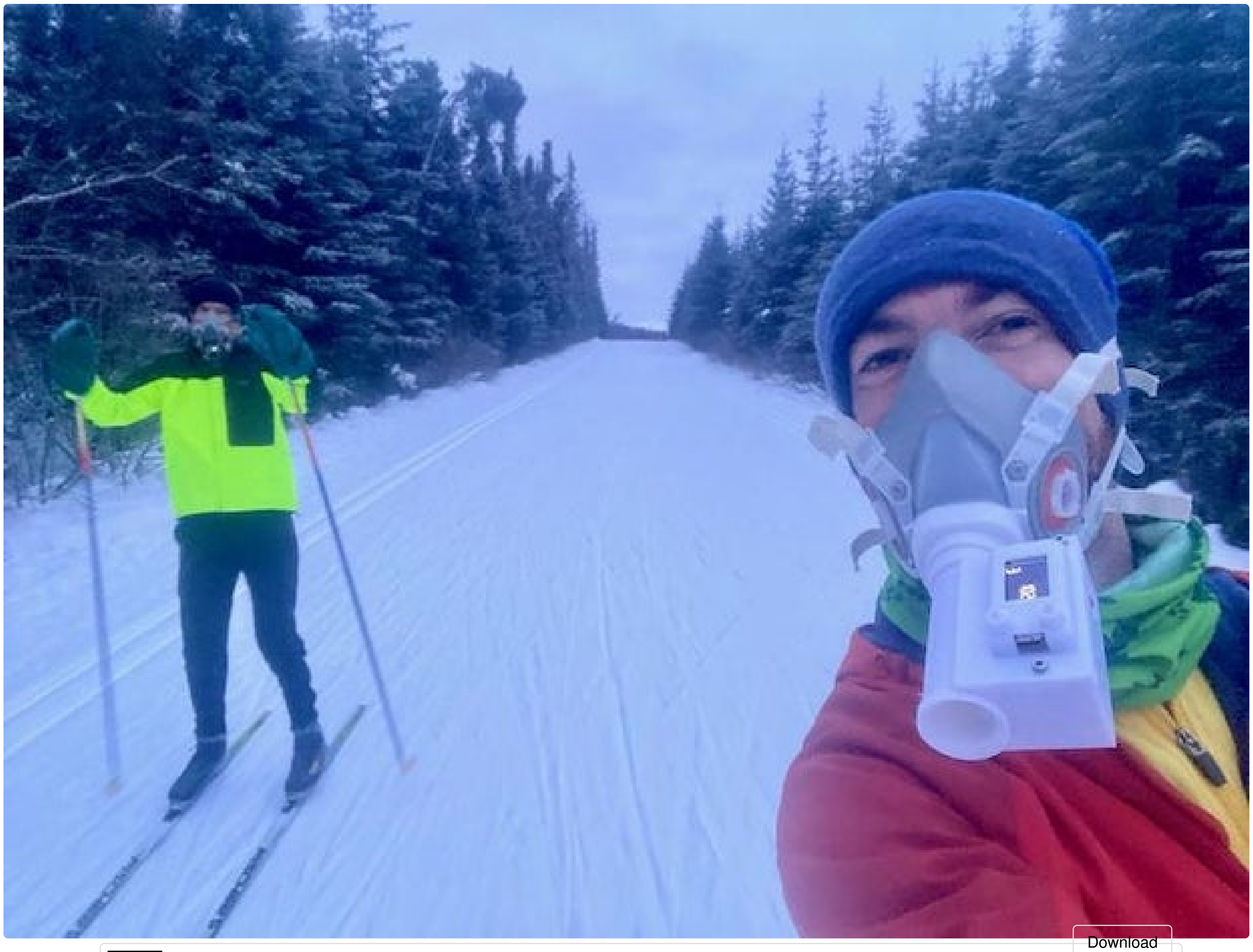
The alternative program is FinalSensirionScreen with above DFRobot_OxygenSensor.cpp and DFRobot_OxygenSensor.h as well as Sensirion_GadgetBle_Lib.cpp and Sensirion_GadgetBle_Lib.h. This is a wonderful App(**https://apps.apple.com/us/app/sensirion-myambience/id1529131572**) that allows you to connect the sensor to your iPhone and collect data, graph it and distribute the data by text or email. The App is designed for collecting data from a CO2 sensor so you have to spoof it by sending the Volume Minute of O2 to the CO2 level screen, the VO2 max to the Temp screen and the O2 level to the Humidity screen. These three pieces of data can then be saved and then downloaded to a spreadsheet program of your choice. The function of the two programs other than their output is identical including the screen output on the ESP32. The App sends out the data in a .edf format which is problematic from a Mac standpoint. Relabel the file as a ".csv" file and this allows you to import it in a regular fashion. Turn on the app only after the unit has been running otherwise it will tend to collect enormous amounts of empty data sets that have to be culled. The two graph screen shots above are from the Sensirion App. The first is treadmill output(VO2 max) the second is X-country skiing(Cal/Day).




28
21
14
7
0
5 05
10
Timeline
1 day
8 hours
1 hour
5 min.
SCD-Gadget
ED:4A


Dashboard
Plot
Menu
5:24
Single Gadget Mode
SENSIRION
Fetch Data
Export Data
CO₂ Carbon Dioxide
RH Relative Humidity
T Temperature
16,000
12,000
8,000


4,000
0
4:00
15
30
45
Timeline
1 day
8 hours
1 hour
6 min.
SCD-Gadget ED:4A
SCD-Gadget F8:AA
Dashboard
Plot
Menu



Download

https://www.instructables.com/ORIG/FV7/Y9BQ/KY5UAQPD/FV7Y9BQKY5UAQPD.h

Download

https://www.instructables.com/ORIG/F84/K7G1/KY5UAQPE/F84K7G1KY5UAQPE.ino

Download

https://www.instructables.com/ORIG/F6U/R7UE/KY5UAQPJ/F6UR7UEKY5UAQPJ.h

Download

https://www.instructables.com/ORIG/F6J/L7UN/KY5UAQPI/F6JL7UNKY5UAQPI.cpp

Download

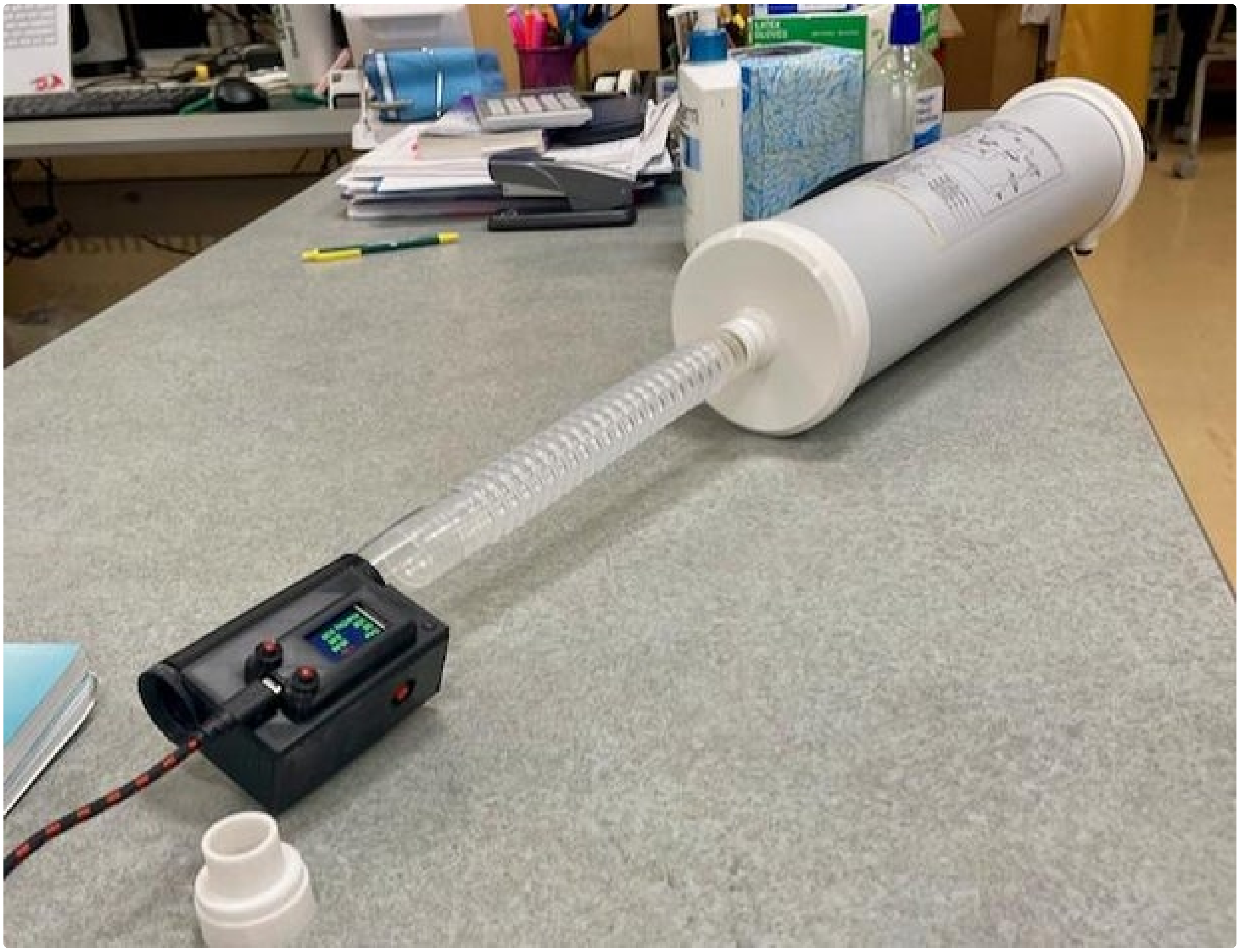
https://www.instructables.com/ORIG/F3H/NNR4/KY5UAQPF/F3HNNR4KY5UAQPF.ino

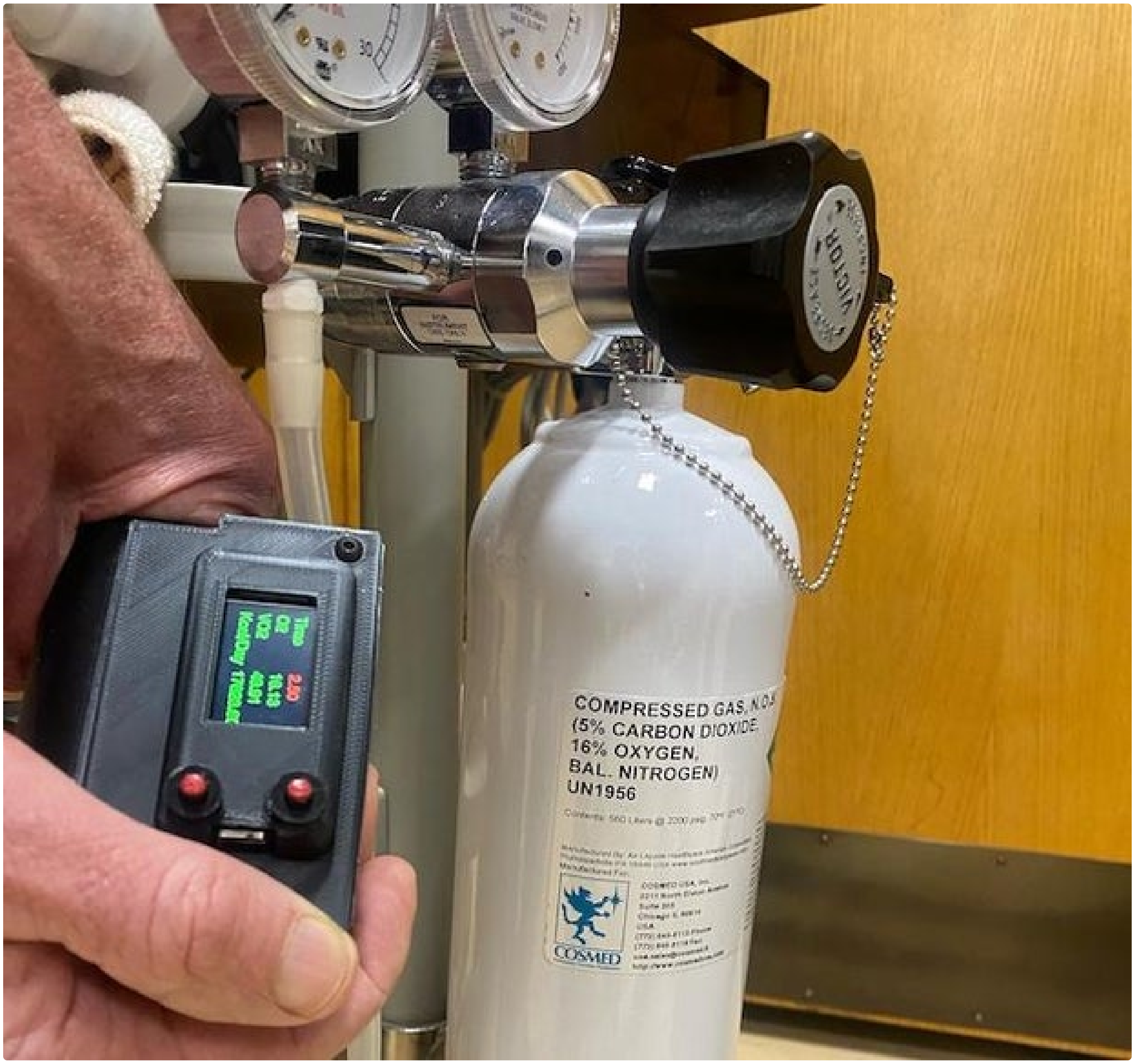
Download

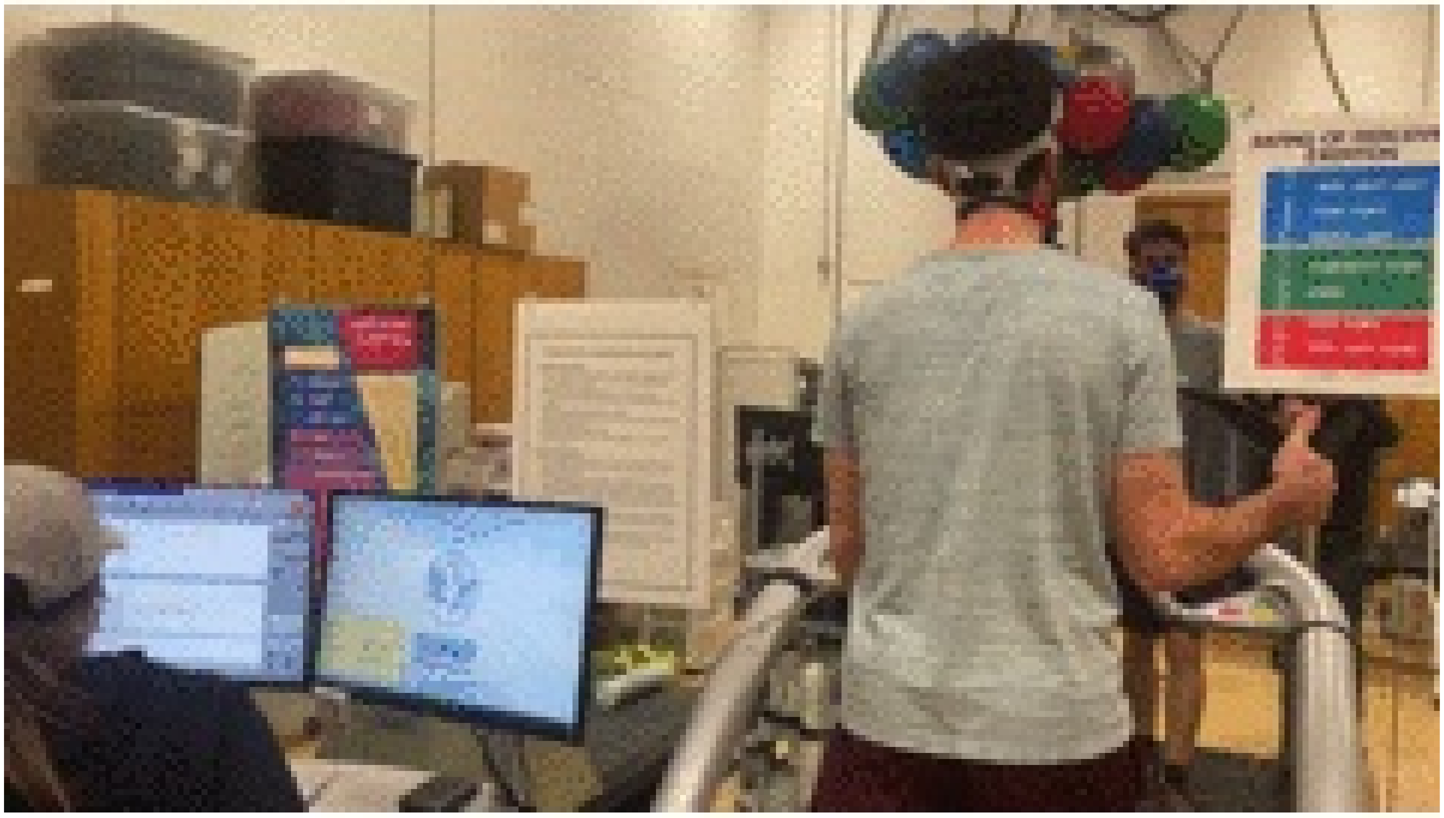
https://www.instructables.com/ORIG/FAL/BOZ5/KY5UAQPC/FALBOZ5KY5UAQPC.cpp

Step 6: Testing It

Several compromises were made in the design of the unit: no CO2 sensor and it's totally compact and weights only 4 oz. Physiology labs that are normally used for testing VO2 max cost upwards of $60,000 and are certainly not portable. They have sensors that measure the same things only on a much finer level. But since what you're trying to study may not really require this degree of granularity perhaps our machine will be adequate. CO2 sensor data helps refine the difference between volume inhaled and exhaled which are not exactly the same. Volumes of these two gasses relative to unchanging amount of Nitrogen in room air makes this possible, but is this a really significant difference? Humans are a fairly subjective test animal and within the limits of testing humans do these things matter? We brought our unit to the University of Alaska physiology lab and compared it. All machines are benchmarked for checking a standardized volume with a 3 liter syringe. The labs unit passed with all measurements being +/- 50 cc. Our unit did very well with all measurements within +/- 100cc. An error this small over liters of air/min is not that significant. Standardized gas (O2 16%) was then passed through our O2 sensor and found to be off by only 0.16%. We then ran our test subject through two levels of testing on the treadmill. The results from both O2 volume and VO2 readings were nearly identical. While there was significant minute to minute variability in the VO2 output from the lab machine the averages per minute turned out to be about the same with each instrument. A more thorough test would involve more test subjects and extended periods of testing of each instrument.



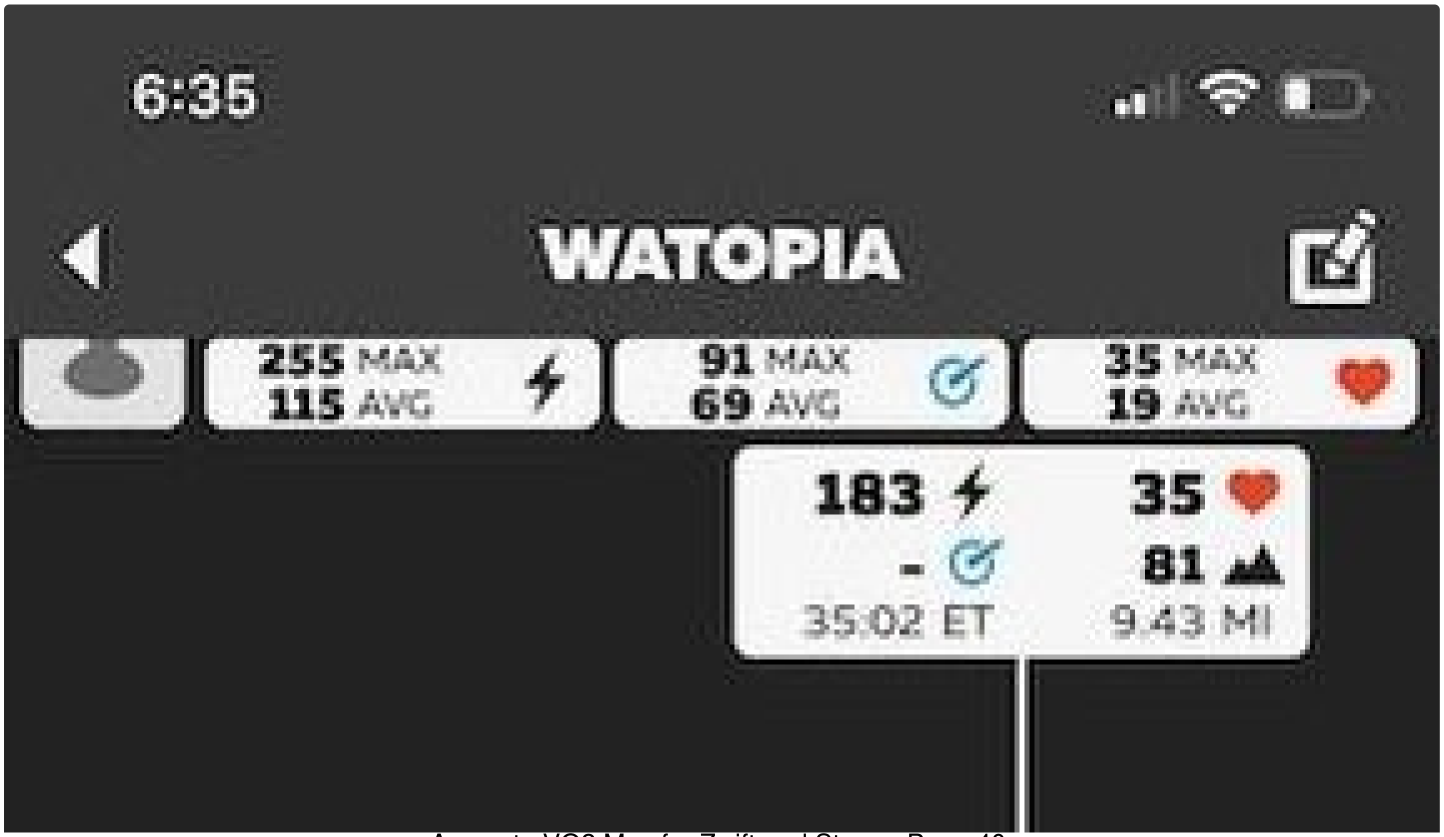

COMPRESSED GAS, N.O.S.
(5% CARBON DIOXIDE,
16% OXYGEN,
BAL. NITROGEN)
UN1956
COSMED
VICTOR

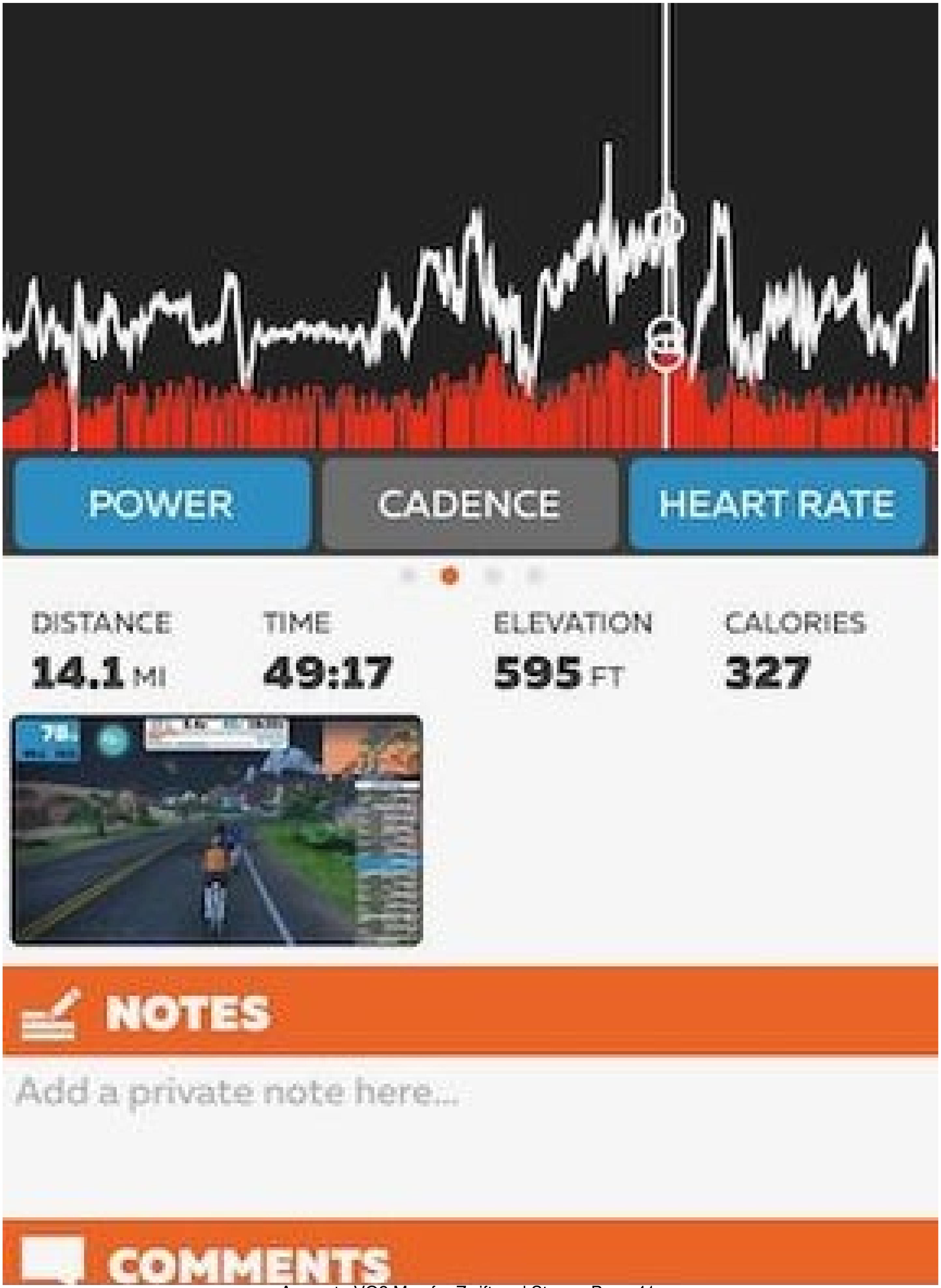


PERCEIVED EXERTION
0 NOTHING
1/2 LIGHT
1 VERY LIGHT
2 LIGHT
3 MODERATE
4 SOMEWHAT HA
5 HEAVY
7 VERY HEAVY
9 VERY-VERY HEA
COSMED
The Metabolic Company
brother

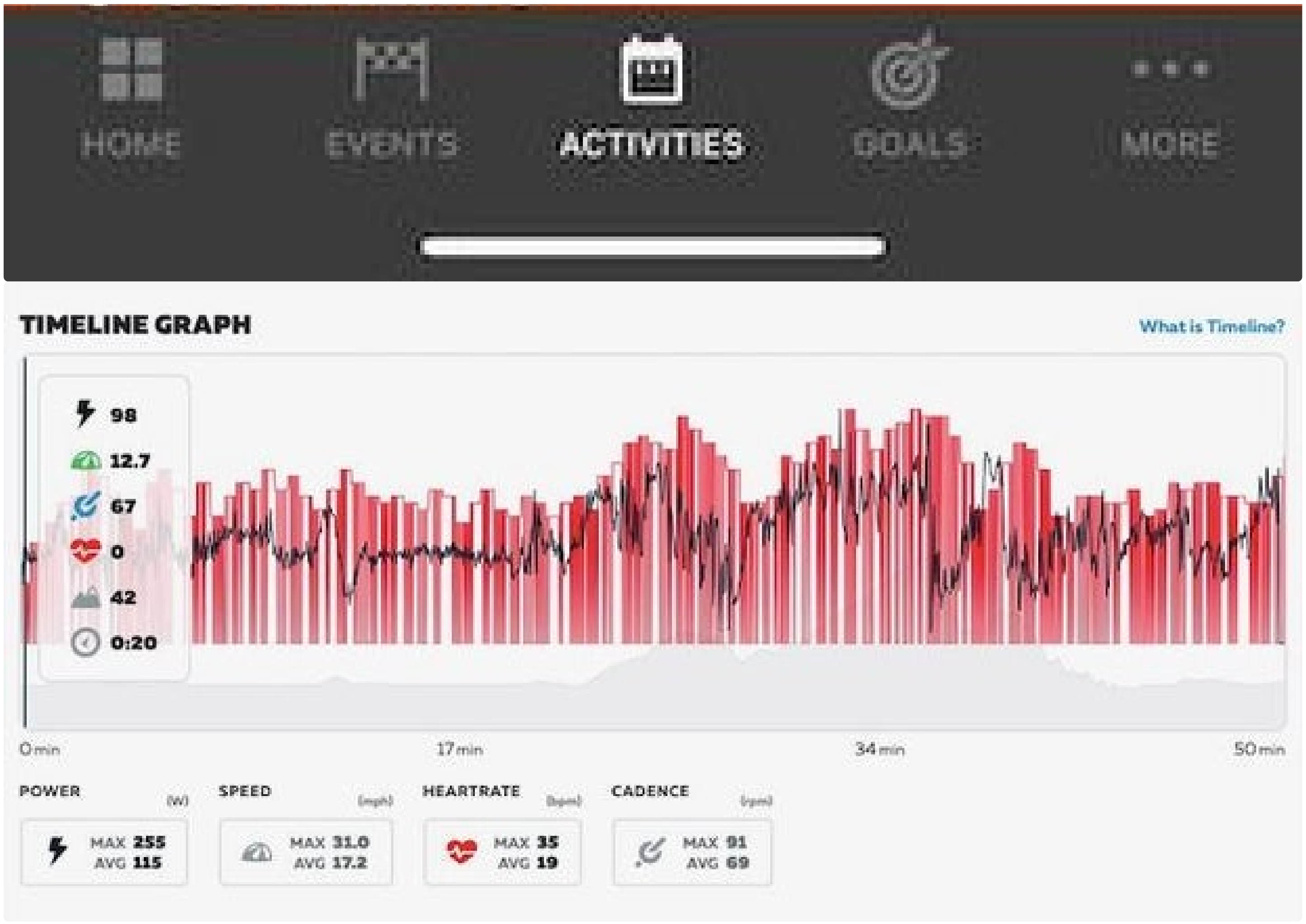
Step 7: Using It

The adapter unit on the mask is threaded and allows the corresponding portion of the unit to be screwed into it. It is designed so that the computer unit and the oxygen sensor are off to the right when looking at the mask front on. This prevents fluid from dripping down into the O2 sensor head. Programing is done through the USB-C connector. Charging the battery is accomplished by turning on the unit and then plugging it in. When using the unit with either **Strava** or **Zwift** you have to pair the unit with the software that you are using. Make sure bluetooth is enabled on the phone that you have Strava on then go to the record function at the bottom of the screen. Once you push it a Heart icon will appear at the bottom--just push it to bring up the bluetooth pairing screen and find the unit. It will then pair. For the **Zwift** unit the pairing is just as simple. When the screen that pairs the bike trainer cadence and power meters appears the hear rate monitor will also appear. This will pair the unit with the screen output. On subsequent pairings it should automatically synch. The data from your bike ride can then be graphed and played with in the same manner as your power output and cadence. The graphs above are the VO2 max output graphed on the **Zwift** App. When using the unit with the Sensirion App for x-country skiing, biking and skating just turn on the device add your weight and push the go button. Then turn on the Sensirion App and it will automatically pair with the unit and start recording data. This is a way of studying many aspects of the human physiology engine that is now portable and cheaply done. How hard is any participant in a sport working? Soft data like pulse rate may offer some clues but the actual amount of energy being consumed will reveal much more about the function of the inner machine. Most VO2 max equations currently available on watches and trainers are based on these soft findings--now they can be based on real data. Since this type of testing has not been widely available due to limitations in size and expense this unit will open up the potential for studying how people improve with conditioning or reveal if they're truly working near their maximum output. The unit is also useful for studying basal metabolic rate. This is the amount of energy that is used by the body in doing nothing. The cost of just being alive. It is hard to study because it should be done with the participant sleeping or resting. Modern studies using doubly labeled water have revealed incredible incites to energy usage among hunter gatherers and office workers. These units could also be used to provide more insight to exertion than the counting of steps. I have found that with the calculations for energy expended from the Oxygen utilization algorithm is slightly higher than that calculated by the work/watt output of the Zwift App algorithm. So good news: more pizza slices from your ride!

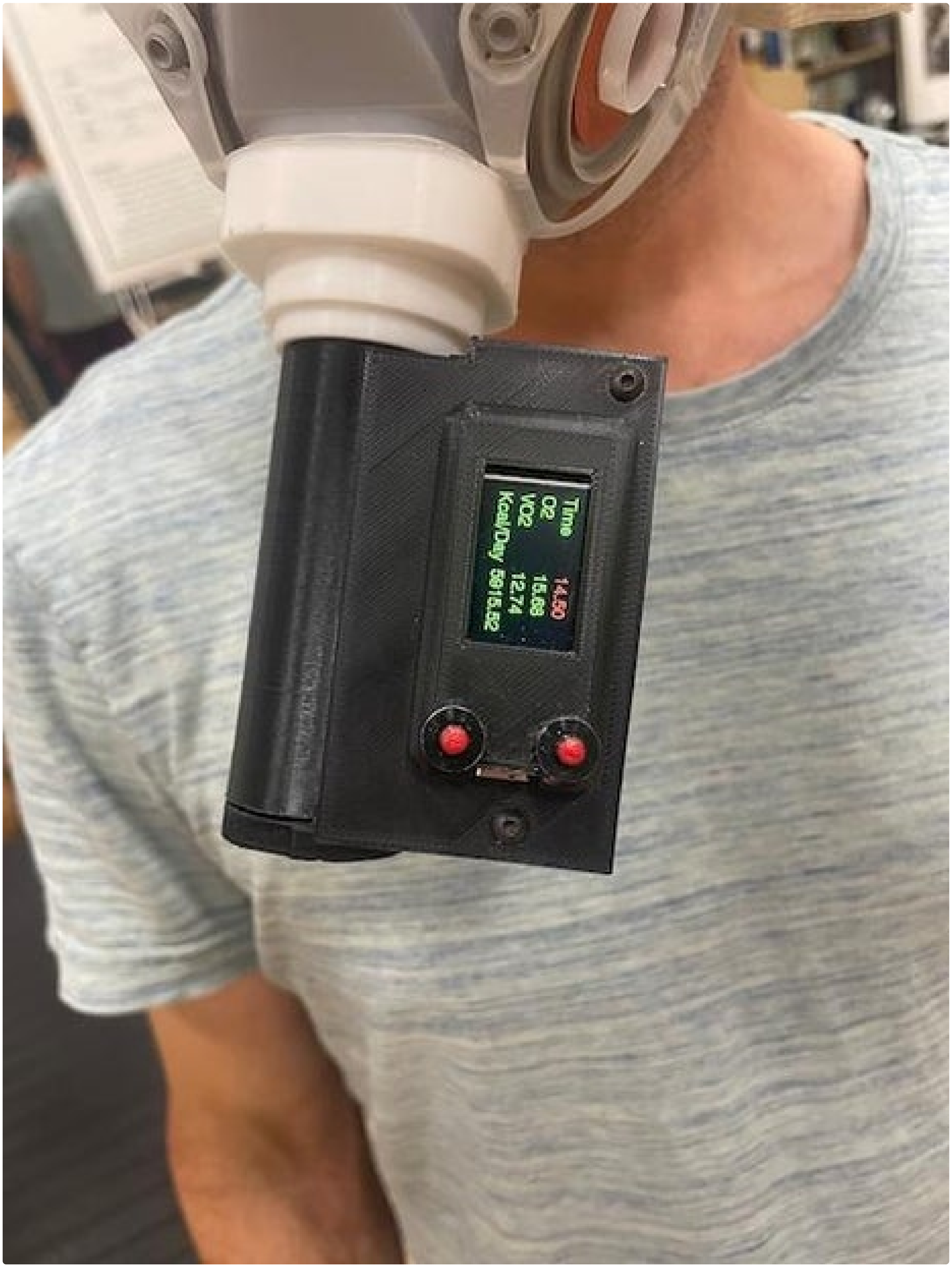




DISTANCE	TIME	ELEVATION	CALORIES
14.1 MI	49:17	595 FT	327



POWER (W)	SPEED (mph)	HEARTRATE (bpm)	CADENCE (rpm)
MAX 255	MAX 31.0	MAX 35	MAX 91
AVG 115	AVG 17.2	AVG 19	AVG 69


Time 14.50
O2 15.68
VO2 12.74
Kcal/Day 5915.82

Step 8: Additional Upgrades

A wonderful innovator from Bielefeld, Germany has worked with his son to develop the hardware and the software to improve the design of the VO2 max instrument. I have included the photos of his new output and modifications as well as his much superior software. His name is Ulli Rissel and while I encouraged him to write up a whole new Instructable he wanted me to just add it to this one. Here is a description of his changes:

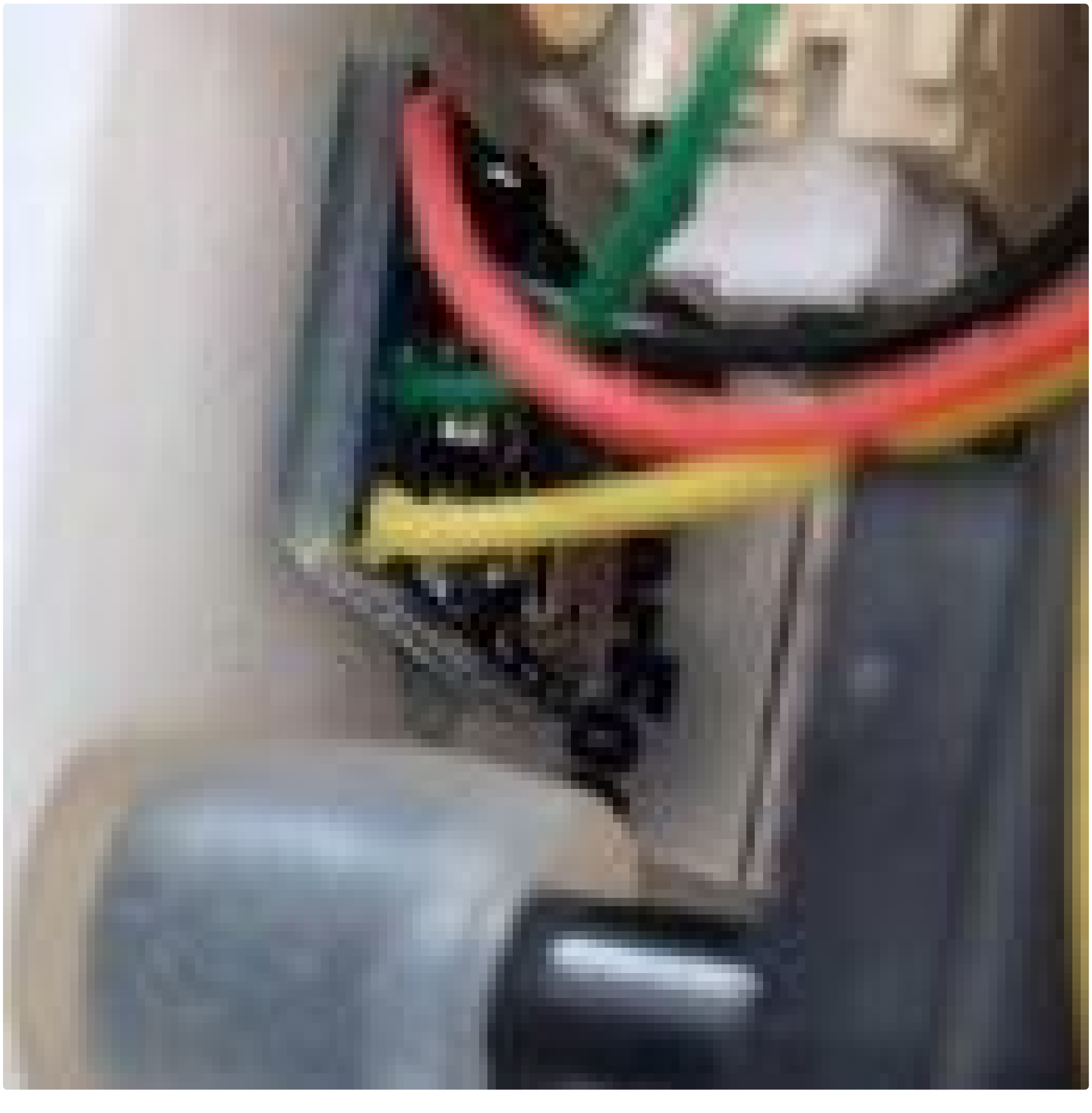
Great project!Here is my version with numerous modifications:

1.I added a barometric sensor (BMP180) to calculate the air density. The calculation of VO2 should also be correct in the mountains thanks to the barometric sensor. The BMP180 sensor is connected to the I2C bus like the other sensors. I glued it between the two sensors on the Venturi tube. In Arduino, the library "Adafruit_BMP085" must be added.2.The VO2max and VEmin calculation is carried out with moving averages and measurement of each individual breath. This allows current measured values every 5 seconds. 3.The displays of the values can be switched during the measurement.4.If the sensor reaches its limit (at approx. 4.5L/s corresponding to a VEmin of 136L/min!) this is displayed. However, the error should only be minimal. If the limit is displayed too often, the Venturi nozzle would have to be modified. 5.A warning appears if the O2 sensor measures less than 20% VO2% at startup. Then the sensor (or room) needs to be ventilated before moving on. If "Continue" is pressed anyway, the value is set to 20.9%.6.Battery indicator: The voltage is displayed with background color to the state of charge. White: USB or charge, Green: Battery full, Yellow: Battery approx. 50%, Red: Battery approx. 20%. The display takes place at the top left. A 1100mAh Lipo lasts about 10h!7.Kcal are now calculated correctly with their own integral timer.8.DEMO mode when the top button is pressed when powered on. It simulates about 28L of respiratory minute volume with respiratory rate of 12 at 4% CO2. This allows you to test the data transfer.9.Reset by pressing both buttons. This works after the 3L check.10.The volume measurement was measured with a 3L calibration pump and a calibration factor was programmed. The volume measurement can be checked in the first 10 seconds with a real-time display. I tested 2 sensors and for practical reasons set the average calibration factor 0.92 for volume measurement. You are welcome to customize this in the program itself (is easy to find).11.The data is sent every 5 seconds as csv via USB cable and Bluetooth serial. Data transmission with cable and BT starts immediately after "Press to start". With Excel Datastream, a display can be made in real time.BT on Excel works really well, even with instant graphical representation. You can also send the data to the APP Bluetooth Serial Monitor on an Android mobile phone and process or send it from there.

As you can see his additions are truly great. But wait they get even better:

I've made some further changes:
Optionally, there is a Venturi nozzle with 20mm for trained athletes. The 16 mm version generates errors due to a too low maximum flow. The current software is set for 16mm and can also be used for 20mm. The necessary adjustment can be found in the commentary at the top of the program. Vo2 is sent via BLE as a heart rate, for example for Zwift, Strava, Wahoo or Garmin. With prolonged operation and high performance, O2 sensor failures occur caused by condensation. The sensor then displays too low O2 concentrations and VO2 is calculated too high. It gets a little better when the sensor is covered with a piece of an FFP2 mask. For this purpose, the diameter for the O2 sensor has been extended to 21mm. You can cut out a 5cm circle from an FFP2 mask, place it on the opening and push the sensor in. It holds without glue. Some thoughts on the calculation: The common spirometers measure the inspiration volume Vi in the supply line, not Ve. However, the concentration of O2 and CO2 is measured expiratory. Therefore, Ve must be calculated in order to correctly record the O2 consumption. Three factors come into play: 1) The volume shrinks minimally if less CO2 is produced than O2 is absorbed (i.e. at quotient <1). This can be corrected via the Haldane formula. However, the deviations are minimal. 2) Much more important is that BTPS conditions prevail in the expiration air (35°C, 95% humidity, actual ambient pressure). 3) The measurement has to be converted to STPD conditions (0 °C, 1013.25 hPa, 0% humidity). VO2 is defined under STPD conditions. (This was the missing part!) According to my research, the conversion is nothing more than the ratio of the density of BTPS (actually measured pressure, 35°C, 95% humidity) to STPD. The density of STPD is constant = 1.2922 kg/m^3. The density of BTPS can be easily calculated, whereby instead of the gas constant 287.058, the constant for moist air is used for BTPS (about 292.8). The constant fluctuates in our temperature range depending on pressure only by

a few percent, the density only in the per mille range, if 292.8 is not adjusted. The density indicates how many molecules are in the air. And now it becomes relatively simple: We measure Ve directly, so no further conversion is required. This results in: VO2 = Ve * O2diff% * Density BTPS / 1.2922. Done! By the way, according to the spirometry instructions, VO2 is only calculated from the O2 difference. CO2 is only used to calculate the quotient and for the Haldane formula, which we do not need.



1000

Download

https://www.instructables.com/ORIG/FQU/60J7/L2DAYLSQ/FQU60J7L2DAYLSQ.ino

Download

https://www.instructables.com/ORIG/F7M/SW7A/L2DAYLSR/F7MSW7AL2DAYLSR._3mf

This is the coolest nerdiest thing I've seen! I love that it can be integrated with Zwift. It really makes me wish I had a 3D printer. FWIW, you should take a look at GoldenCheetah too. Should be easy to integrate.

I know this comment was a while ago... but FYI I've just got my build working with GoldenCheetah by sending out the VO2 master pro bluetooth messages instead of heartrate.

Hi, can you publish your code?

Many public libraries nowadays have a 3D printer for use, you can call around and find one close to you. The other option is a 3D printing service, not free but fast and convenient. Simply send the files for a quote and many ship next day. Hope that helps jumpstart your project!

It's a good thought but I have an autistic child with epilepsy at home so my free time is virtually non existent. Even if I'm on the trainer doing Zwift I have to jump off at a moments notice. At best I could order parts if I get time to get the other parts needed.

Thanks so much for your encouragement! GoldenCheetah looks amazing...thanks for the information I didn't know there was that much interest in the data!

Thanks

Is there any online store where you recommend me to get the TTGO T-Display quickly? In all the places I have found, like ebay or aliexpress, takes more than a month to arrive.

Just wanted to add congratulations for the project, that was brilliant fun to build.
I used a 3d printing service to get the parts and I'm amazed by the quality of those.
Had some minor troubles getting the right settings in arduino for the ttgo board (in the end I used the esp32-dev setting, corrected the SDA in the header and downloaded the driver from (https://github.com/Xinyuan-LilyGO)
Also some frustration with the wiring for my AD2 being different to the picture (took me hours to spot that!)
After an initial play I found a BMP85 in my parts "bucket", so added that too and switched to Ulli's firmware (adding back in the sensiron output)
Again.. brilliant project, will have fun playing with this.

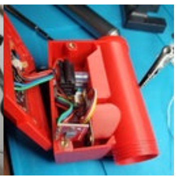
The parts list should be updated to include DF-CABLE3 (from digikey) to go with the Omron--D6F-PH0025AD2 or the recommendation for the part should be switched to the D6F-PH0025AD1, the part from the pictures.

The DF-PH0025AD2 is a connector-type model so it really needs that connector. See the datasheet here:
https://media.digikey.com/pdf/Data%20Sheets/Omron%20PDFs/D6F_Series_Cat.Ver2.pdf

I forgot to order one... but couldn't stomach the cost (with additional international shipping) just to get the cable so I made a DIY one for the AD2 using a thin printed pcb I scavenged out of my "bin" and soldered wires to that. The AD2 is a push in edge connector rather than pins so this works well.

Totally a good idea...thanx!

Superb project, looking forward to integrating this in my telemetry tracker - https://blog.ivor.org/2022/02/between-rak-and-swee...
Bit of frustation that my Omron wiring was different to the guide - that took me hours to spot!
Found a barometer in my drawer, added and switched to the Ulli firmware.



Great project!
Here is my version with numerous modifications:

1.I added a barometric sensor (BMP180) to calculate the air density. The calculation of VO2 should also be correct in the mountains thanks to the barometric sensor. The BMP180 sensor is connected to the I2C bus like the other sensors. I glued it between the two sensors on the Venturi tube. In Arduino, the library "Adafruit_BMP085" must be added.
2.The VO2max and VEmin calculation is carried out with moving averages and measurement of each individual breath. This allows current measured values every 5 seconds.
3.The displays of the values can be switched during the measurement.
4.If the sensor reaches its limit (at approx. 4.5L/s corresponding to a VEmin of 136L/min!) this is displayed. However, the error should only be minimal. If the limit is displayed too often, the Venturi nozzle would have to be modified.
5.A warning appears if the O2 sensor measures less than 20% VO2% at startup. Then the sensor (or room) needs to be ventilated before moving on. If "Continue" is pressed anyway, the value is set to 20.9%.
6.Battery indicator: The voltage is displayed with background color to the state of charge. White: USB or charge, Green: Battery full, Yellow: Battery approx. 50%, Red: Battery approx. 20%. The display takes place at the top left. A 1100mAh Lipo lasts about 10h!
7.Kcal are now calculated correctly with their own integral timer.
8.DEMO mode when the top button is pressed when powered on. It simulates about 28L of respiratory minute volume with respiratory rate of 12 at 4% CO2. This allows you to test the data transfer.
9.Reset by pressing both buttons. This works after the 3L check.
10.The volume measurement was measured with a 3L calibration pump and a calibration factor was programmed. The volume measurement can be checked in the first 10 seconds with a real-time display. I tested 2 sensors and for practical reasons set the average calibration factor 0.92 for volume measurement. You are welcome to customize this in the program itself (is easy to find).
11.The data is sent every 5 seconds as csv via USB cable and Bluetooth serial. Data transmission with cable and BT starts immediately after "Press to start". With Excel Datastream, a display can be made in real time.
BT on Excel works really well, even with instant graphical representation. You can also send the data to the APP Bluetooth Serial Monitor on an Android mobile phone and process or send it from there.

@rabbitcreek: How can I publish the modified software?
Have fun

Hi Robert
i posted a mail with the modified software to publish it on your project site.
Ulli

Not sure if you got this reply. My direct email is Rabbitcreekatgmail.com….I totally loved your modifications to the instrument! You can either write a new Instructable or you can edit my original one to add the new upgrades…either way thanks so much for working on the project
Robert

This is a very neat design. Out of curiosity why do you use an oxygen sensor and not a CO2 sensor? Do they exist cheaply enough to measure exhaled breath CO2 concentration, or is it compatibility with the Sensirion app? Did you consider using an orifice plate instead of ~Venturi restriction? If you went the orifice plate route would it compact the design?

Thanx very much! Really good question--my original design did have both sensors and the formula that determines vo2 max is more accurate but suffers from size increase. I initially in this design used a new smaller CO2 sensor SCD 41 but found it lagged behind the O2 sensor and didn't have the range for exhaled gas. I wanted to see if just using the O2 sensor came close enough to make it accurate and it did. I built the portable spirometer project last year and found it gives very accurate results with the curved tube so have used the exact same design ever since. Would love to try the orifice plate actually. I keep thinking of ways to make it smaller...

I ordered all parts for 2 masks ;-) Hope they will arrived soon from China. My 3D printer is already working.
Could You please provide the Fusion 360 files for modifications? I would like to modify one mask with an additional CO2 sensor. As you posted before the SCD41 is to slow and the range is too low. So I ordered an SCD30 for the modifications.
I would like to measure the respiratory quotient and the CO2 level during ramp test. My son in law and I are very exited to build and test the masks. Thanks a lot for this really cool project!

Hi! Sorry, I'm kind of new in this and was just wondering if you suceeded on adding the co2 sensor and if you could share the required changes? I assume there may be some changes on the printable parts, as well as on the calculations code. Also thanks to the creator of this awesome project!

In the meantime I build this really cool and awesome project in the original version without a co2 sensor. Actually I am modifiing the software for my needs (added a barometric sensor for correct calculations in higher altitudes, added a bluetooth connection to stream data to excel). It is still work in progress. Integration of a co2 sensor will be the next modification. I will let you know when it's successfull.

Adding the barometer compensation was really our next thoughts about this project…I'm really glad u are doing it! Please keep me informed on how your doing.
Thanx
Robert

Here is the public link for the Fusion 360 design files:
https://a360.co/3J7HAHG
have fun!

That's so great! But I have two little questions. Can I use the SENSIRION SDP810 instead of the Omron one? It is much cheaper and easier for me to buy. Both of them are using I2C, how many codes do I need to change?

The sensation sdp810 is rated for sensing -500 to +500 so its accuracy within the range you are interested in which is about 0 to 250 for most large breaths will probably be reduced but I can't tell if this will be significant. The only way to check is to try it out. The Senseiron will have a different library to make it work but is readily available on the web.

Very cool, even though the tech stuff goes way above my head :)
Just wondering whether by 'encumbering' you meant cumbersome?

Had to look up both definitions to be sure but they both apply! Thanx for looking at my work.

This is such an awesome project and I bought all the parts to it, loaded code on the TTGO and nothing shows up on the screen. I thought maybe I bricked the TTGO however I can load the TFT_eSPI Factory Test Example and everything works.

Did you have any problems with the screen not displaying anything? I have also tried this with both the Sensirion and Zwif .ino files.

Also what are the OxygenSensor 2.cpp and OxygenSensor 2.h files for?

Cool that you are doing this! I will help you any way I can. The initialization checks to see if the oxygen sensor is online and it stops the program until its I2C connection is ok. Finish wiring the oxygen sensor to the TTGO board and monitor the serial channel to make sure that it is ok. Those 2 files are required to make the oxygen sensor go. You can breadboard it out to make sure everything is working before building it. Hope this helps.

Success!!! I apparently miss read the pins and soldered to the wrong pin. I didn't know about the Serial Monitor in Arduino, that helped tons! Once that was resolved I could see the sensor reading but still no display.

I found out that the built in Arduino library TFT_eSPI was having issues with my TTGO, I used the below library and the screen worked!

https://github.com/Xinyuan-LilyGO/TTGO-T-Display

On to finishing the actual construction (I need to go buy some aquarium tubing) and reprint the buttons...

On the weight is this in the units of KG or LBS? I assume LBS since I see this in the code " (float(wtTotal)/2.2)"

Another note when I try to compile FinalSensirionScreen.ino I get the below error. I can compile if I just comment out that line as I don't see this variable used anywhere but I wasn't sure if this was supposed to be there or not:

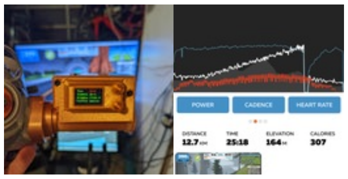
```
FinalSensirionScreen.ino: In function 'void goFigure()':
FinalSensirionScreen:241:15: error: 'vo2CalMaxMax' was not declared in this scope
tft.println(vo2CalMaxMax);
```

very sorry that error is mine ...stick this in the list of variables at the beginning of the program:
float vo2CalMaxMax = 0.0;
and it should work fine...made last minute change to output screen when I posted the software

Success I think! Is there a simple way to confirm calibration?

I just did a ramp test and got a 38.14 using the zwift code.

My guesstimate Garmin watch claims a 58.



The oxygen sensor is calibrated by leaving it on and measuring for a long while and then pushing the button near the base for over 4 seconds until it flashes...the measured O2 level on the serial monitor will fall for some reason. On restarting the machine the O2 level will change to correspond to presumed calibrated level. The vol sensor can also be calibrated by forcing a known volume of air through the system and adjusting a correction factor in the program. You can see the "correction" value is one in the program because I found the volumes from the sensor to be fairly accurate. But because of the variabilities of these systems you may have to compensate this. All of the expensive physiological units are tested prior to testing to with 3 liter volumes and a adjustment factor is put in. I used a 2 liter air pump from my raft to test it as well as the unit at the university. Watch the oxygen level through the output screen and the serial monitor and the volumes coming out to make sure everything is working.

I'm totally going to make this but would love it if you could expand the parts list a bit with more specifics? Things like the threaded inserts, model of battery and the type of on off switch and all the details would be greatly appreciated. This is so awesome! Thank you so much for sharing this with the world.

Thanks for the encouragement ... I try to make these Instructables fun.
Details:
1000mAh 3.7V lipo rechargeable Battery 503450 polymer lithium Li-Po USA
this is the ebay selection for the batteries...hard to get them any other way these days..
Should fit the recess in the case exactly.

AJOYIB 300Pcs Brass Threaded Insert Nuts M2 M3 M4 M5 Female Thread Knurled Nut Inserts Embedment Nut for 3D Printing Project
This is the Amazon order for the 3D inserts...use the 3 mm of any length. Use a rounded 3mm bolt or it will interfere with inserting the USB- C cable.
Here is the Adafruit part list for the On/Off button: https://www.adafruit.com/product/1683 $2
These plastic small buttons are nicely Superglued to the case and fit the recess perfectly.
If you need any additional details just text back.

I get no benefits from any companies mentioned...

I came here to compliment the build, but also to ask the same question as pzadzzzz. I'd love to build this, but have concerns about breathing through a home-made plastic tube. I have no experience with 3D printing.

I think all the concerns were raised regarding small particles created during printing. The finished, printed object is probably just a big piece of plastic.

The design is such that you only breath out through the instrument so the effects would be minimal. Also those bio-disposable drinking straws are all made out of the same material---corn! Of course they are not really biodegradable anyway....

Innovative solution! I have thought this was possible for a while, and now you have confirmed it. Question - why did you place the flow sensor perpendicular to the flow of air? Would it not be more accurate if placed in line with exhalation?

The sensor has a high pressure port and a low pressure port that correspond to the openings in the 3D printed housing. The low pressure port is at the peak of the Venturi narrowing and the high pressure one is the opening prior to this port. The channels are then led out to holes that match the configuration of the sensor--it looks perpendicular but it really is in line like you suggested.

Hey there. Awesome idea. Looking at building this myself. Is there an Android interface option? Or is iOS it? I'm not too fussy about seeing it on screen in place of HR, but would love to export the data afterwards. Also, while using it, I can't also use HR correct?

If you just download the data to the Sensirion app you will be able to link your HR monitor to the screen without any problems just use the other version of the software.
Thanks again. If you need any further help with the build just email.

Here is the link for the Android version of the Sensirion App:https://play.google.com/store/apps/details?id=com....

I haven't used it but I assume it will work. Yea the Bluetooth only takes data from one source.

Amazing project, great job! I was wondering about the safety of breathing heavily through a device made from PLA material. Is there any data on that? I know a while back that it was discovered that 3D printing does pose some risk in that the powders used pose an inhalation risk and that, while working around 3D printer raw materials, safety gear should be worn.

Our world is literally filled with residual particles of everything that is manufactured around us...PLA is supposed to be one of the more innocent ones...made from polymerized corn residue...the rest of the 3M mask that you in inhale through has been in use for decades so I assume has gone through some governmental prior approval. Thanx so much for kind comments. Truly enjoyed that you liked it!

As an old-school physiological ecologist, I am amazed that you can do this. Bravo.

Thanx...this was a fun build and using the Physiology lab was a real treat.....