POC NED

POC comes in two forms, form factor (fully assembled) as well as NED (non-enclosure device).

This page outlines specific functionality and configuration details for the Milan POC NED setup. Please see this wiki page for details on POC form factor.

Getting Started with POC NED

Connections:

The debug board offers several debug functionalities:

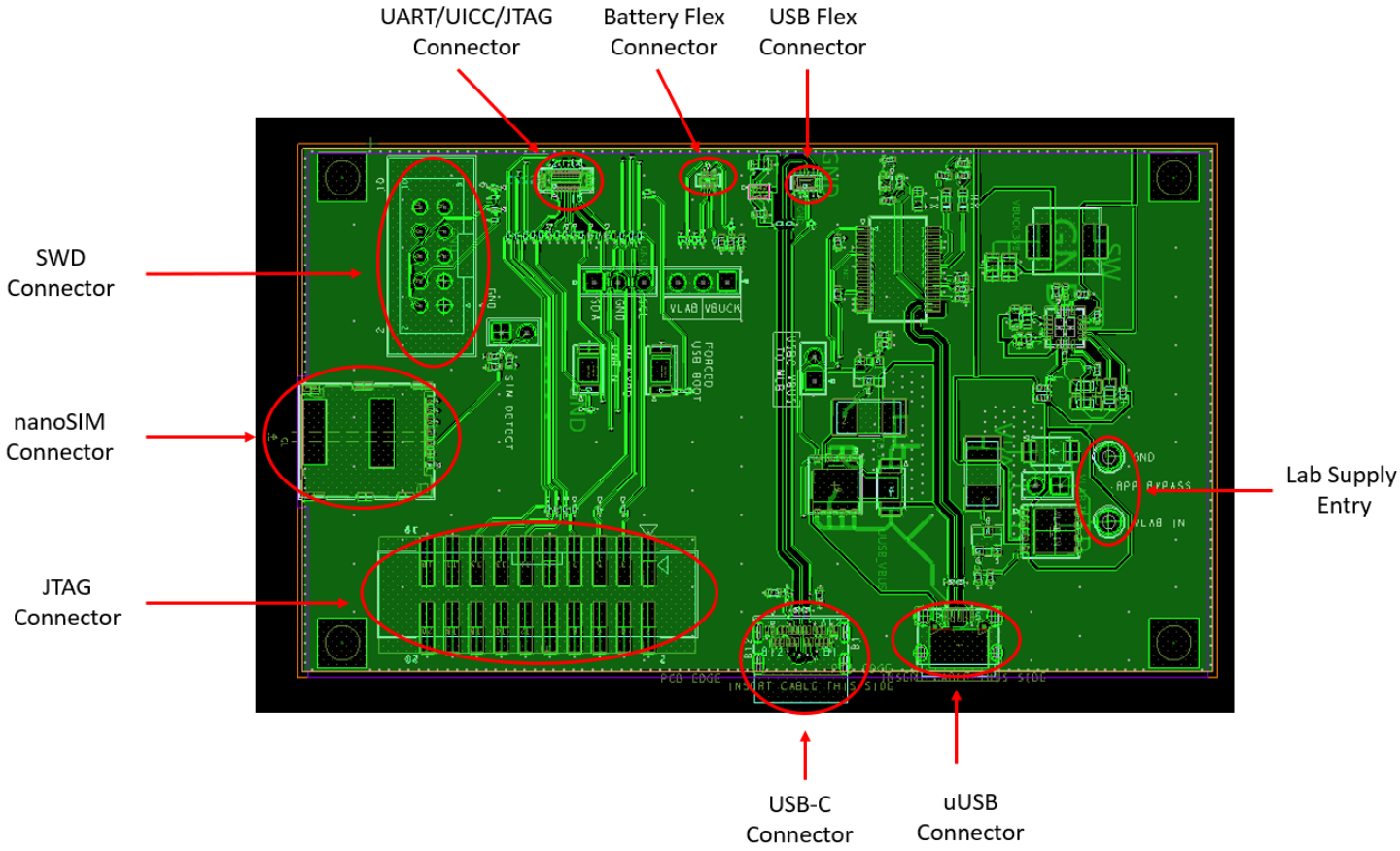
- NanoSIM: **Note please do not use this option**.
- Fake battery options can be selected on J400 : **1-2: uUSB supply OR 2-3: Lab entry supply**
- uUSB provides fake battery as well as UART access to the device
- USBC provides USB VBUS, CC1, and USB DP/DM signals. Note: since there is only CC1, the device only supports charging in **one of the orientations**.
- SWD Conn: Supports SWDIO/SWDCLK for MCU debug

- JTAG Conn: Supports JTAG access for SoC debug
- Battery connection: When using a real battery, please **disconnect the Battery Flex Connector** from the board and directly connect the battery to the Battery Flex Connector flex.

Design Files:

Debug board can be found here.

Rest of the files can be found here.



Initial Bootup:

The first batch of boards did not come with batteries, so one of the fake battery option is required to boot up the board.

The default jumper configuration is set to use fake battery via uUSB. Connect micro USB to device, press KYPD_PWR key and the device should boot up. This enables power as well as UART access.

- Charging should be disabled while using fake batteries at every reboot.
- echo 1 > /sys/class/power_supply/battery/input_suspend

Do not connect USB-C before hand, it will cause protection circuit to kick in and create a reboot loop. After the device is booted, you can connect the USB-C cable.

USB Access:

The USB-C connector provides USB access for the unit, however it also activates the protection circuit that prevents back powering to fake battery sources when USB-C Vbus is present.

- Recommend to use USBC-USBA variant of the cable with the red button unpressed, USBC-USBC will only operate in one orientation.

The recommended configuration is to use **a battery connected directly to the main board (see below)** and communicated with the device via **USB-C** with no uUSB connected.

Note: You could enter a reboot loop when you unplug the uUSB (while USB-C is plugged in). In this case, you may have both USB-C and uUSB plugged in, though the **R404, R420, R421** region might get very hot.

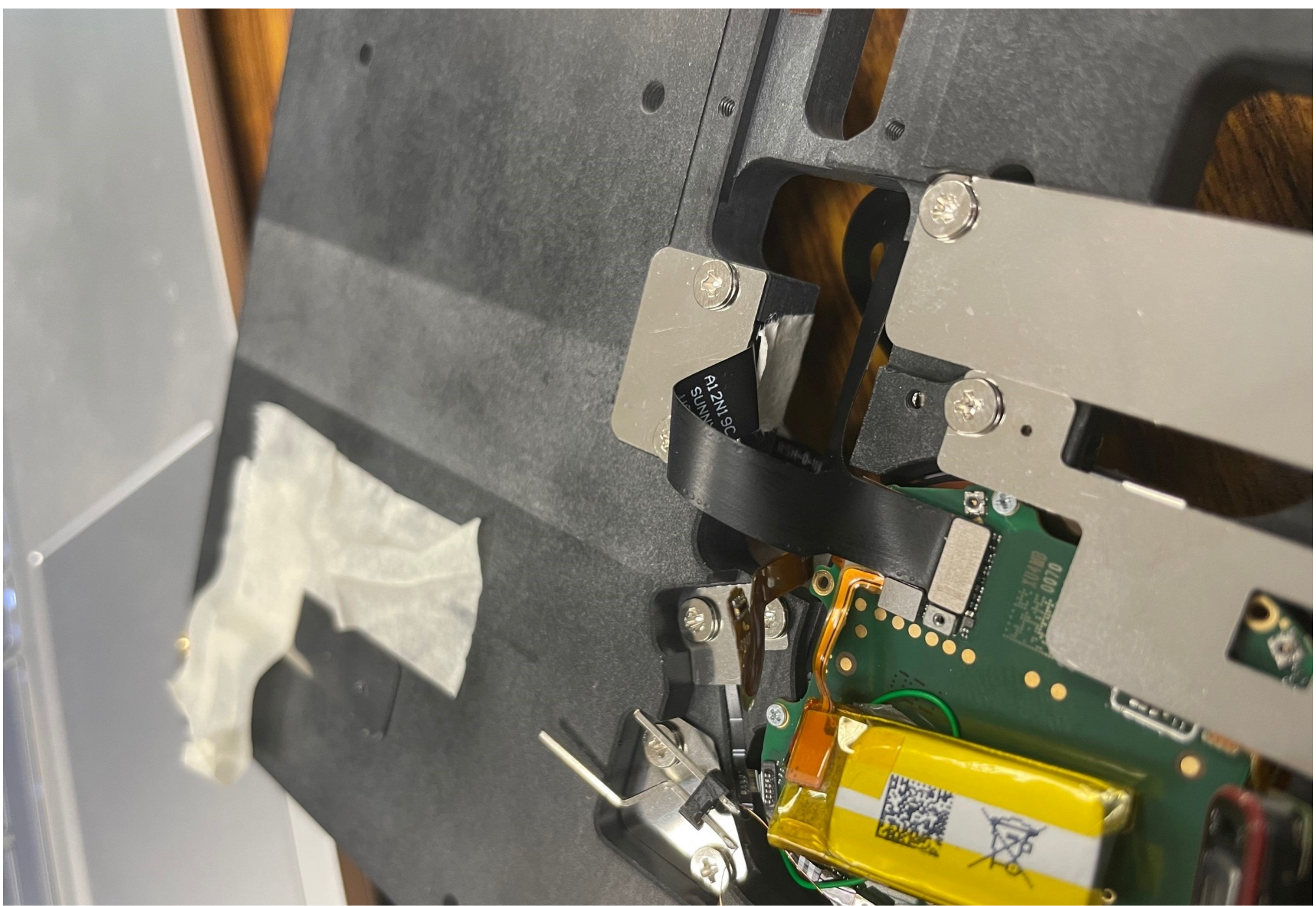
Note: The USB-C that came in the box has a red button. If you don't see anything under `adb devices,` try to 1) flip the cable side 2) press the button.

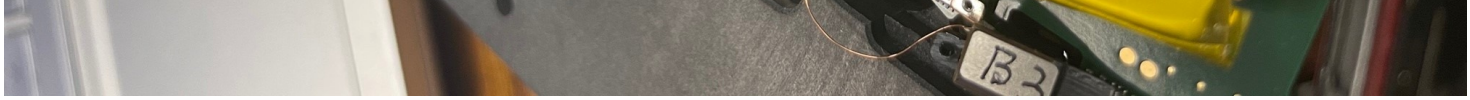
Note: The quanta build on some NEDs may have developer options turned off and those maybe needed for ```adb devices``` to work.

Battery:

Quanta team advised us that we need to connect the battery directly to the board - do not use the battery flex.

1. Flip your POC NED device over, look for the housing that the battery flex is under.
2. Remove the protective clip (needs a very small Phillips head screwdriver).
3. Unplug the battery flex cable from the board and connect the battery.



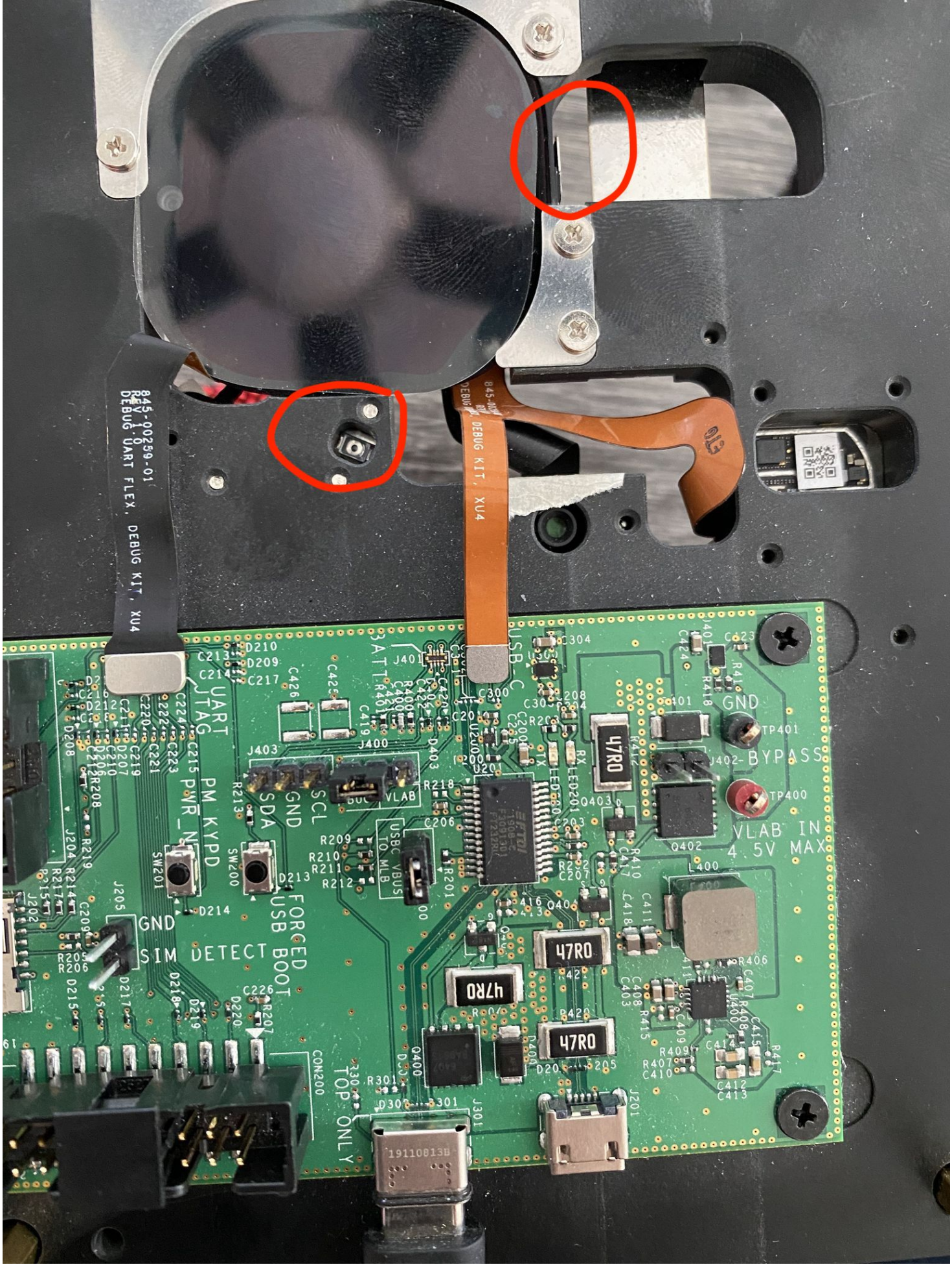


How to Flash the Unit:

The Maui CLI provides a simple command to look up, download, unzip and flash a CI build to your device. You may need to first put your device into fastboot mode (`adb reboot bootloader`) before maui will recognize this device.

FORCING THE DEVICE INTO FASTBOOT MODE

If you flashed a bad image and the device refuses to boot, you can try to force it into fastboot mode, by powering down the board, then holding down the **camera button** while you press the **power button**.



FB IMAGE

Same as steps highlighted in the EDB wiki.

If you'd like to do this process manually:

1. Download a good (== "passes all tests") fastboot image **after Oct.7** from here. Unzip it and cd into the resulting directory.
2. a. If you built a fastboot image on your devserver it will be in ~/local/mos/out/target/product/mos/mos-update-img-eng.$USER.zip. If you need to flash custom system images, disable verity and reboot your device before flashing custom images: `adb root  adb disable-verity  adb reboot`
3. Connect your NED to your computer by USB-C.
4. Verify that adb can see it with `adb devices` . If there is more than one, disconnect the others temporarily so the flash script knows which one to flash.
5. `cd` to the directory where you unzipped the image
6. Flash the board (rebuilding partitions if it is your first time):
7. **First time flashing?** `./flash_all.py -p -w -f`
    - `-p` will rebuild the partitions
    - `-w` will wipe userdata
    - `-f` will override the logic that seeks to avoid accidentally flashing non-milan android devices, so please be careful to ensure you are flashing the right thing.
8. You should see logs like P135685538.
9. **All other times**: `./flash_all.py` .
    - You may want to use -w to wipe userdata, but this is optional.
    - You should not need to force with -f or to repartition with -p.
10. The board should reboot and be visible via `adb devices` once more in a couple of minutes.
11. Profit!!!

You can check which build is flashed on your device with the command: `adb shell getprop | grep ro.vendor.build.fingerprint`

QUANTA IMAGE

Default image will not support camera use (camera app will cause UI to crash).

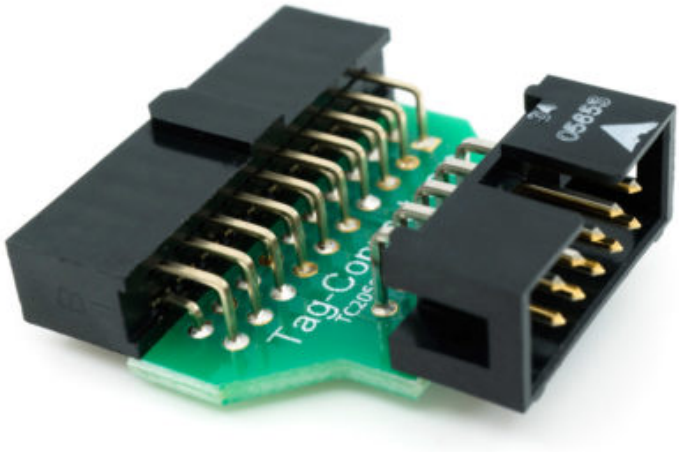
Please update to Quanta SW V20 here

Flashing Instructions can be found here

MCU JTAG:

The MCU can be reprogrammed (SRAM only) and booted with the latest in ovrsource. J204 is a 2x5 pin, 0.1 pitch JTAG connector.

Order this cable for your J-Link JTAG adapter. One at https://www.tag-connect.com/product/tc2050-arm2010-arm-20-pin-to-tc2050-adapter has been tested and is confirmed to work. You will need a ten wire female-to-female cable, readily available from multiple sources. When looking for the cable, go for a short one.



Until you have a cable, jumper cables can be used with the following connections:

```
Name    J204        J-Link
1V8     Pin 1       Pin 1
SWDIO   Pin 2       Pin 7
GND     Pin 3       Pin 8
SWDCLK  Pin 4       Pin 9
GND     Pin 5       Pin 10
```

MCU Debugging without JTAG

If you don't have a JTAG adapter, are debugging a form-factor POC, or somehow have a faulty JTAG flex cable, you can still debug the POC MCU using logging, as follows, with ADB connected:

Flash the image (without picocom running!):

```
$ make VERBOSE=1 BOARD=poc flash_from_soc
```

Connect picocom to get the MCU console:

```
$ adb root
$ adb shell
mos:/ # picocom -b 115200 -f n /dev/ttyHS0
```

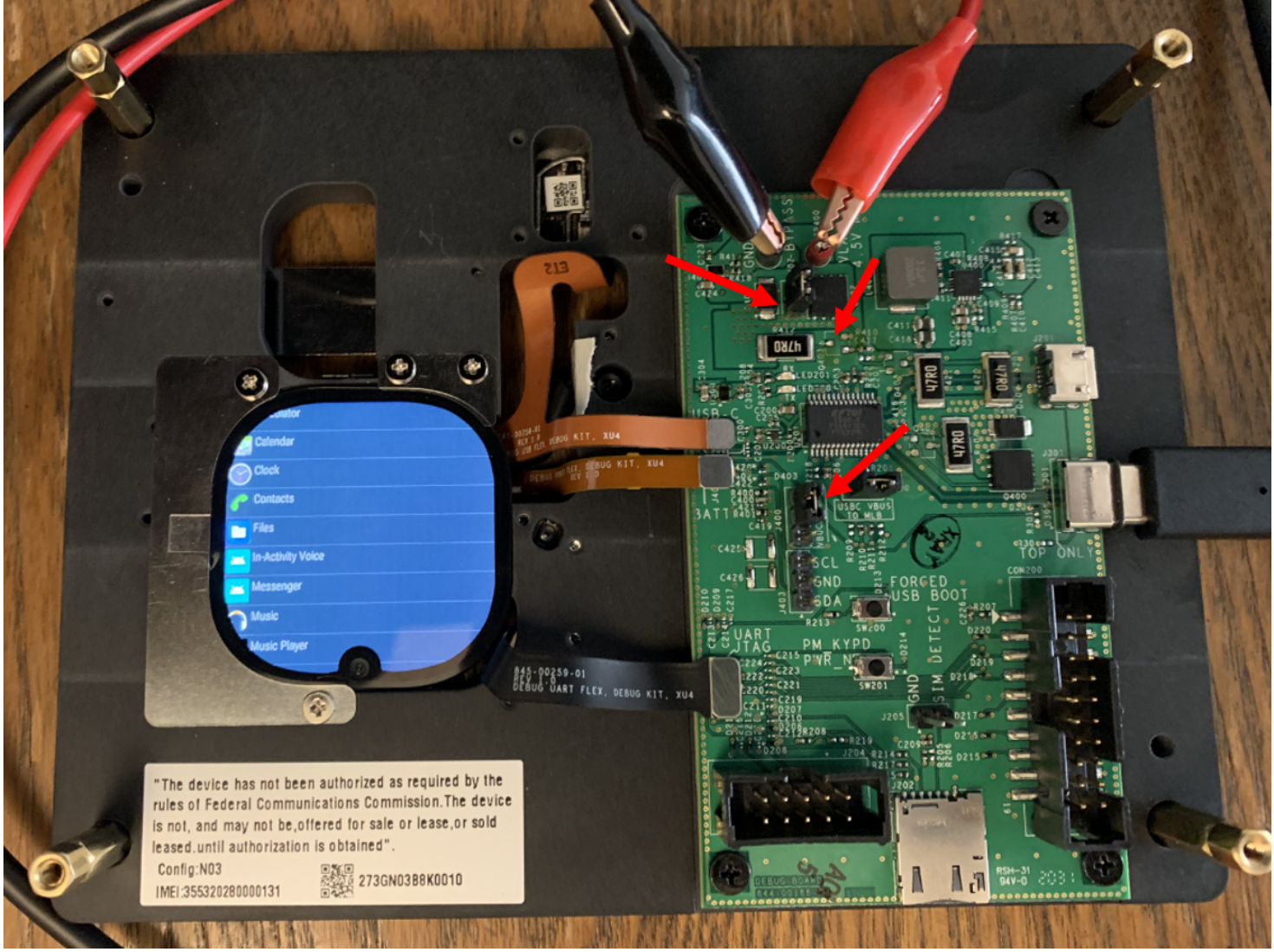
Reboot the MCU to capture the logs from boot:

```
$ adb shell
mos:/ # echo "flashing" > "/sys/devices/platform/soc/soc:oculus,rt600_ctrl/boot_state"
mos:/ # echo "normal" > "/sys/devices/platform/soc/soc:oculus,rt600_ctrl/boot_state"
```

Power Measurement:

For accurate power measurements, please use **Lab Entry Supply**, by selecting **2-3 on J400** and shorting **J402**. Additionally please **remove Q403**, otherwise it will discharge power while USBC Vbus is present. (Similar protection mechanism as mentioned above).

- Please also **disable charging**, using the same command as mentioned above.



Wireless Subsystems

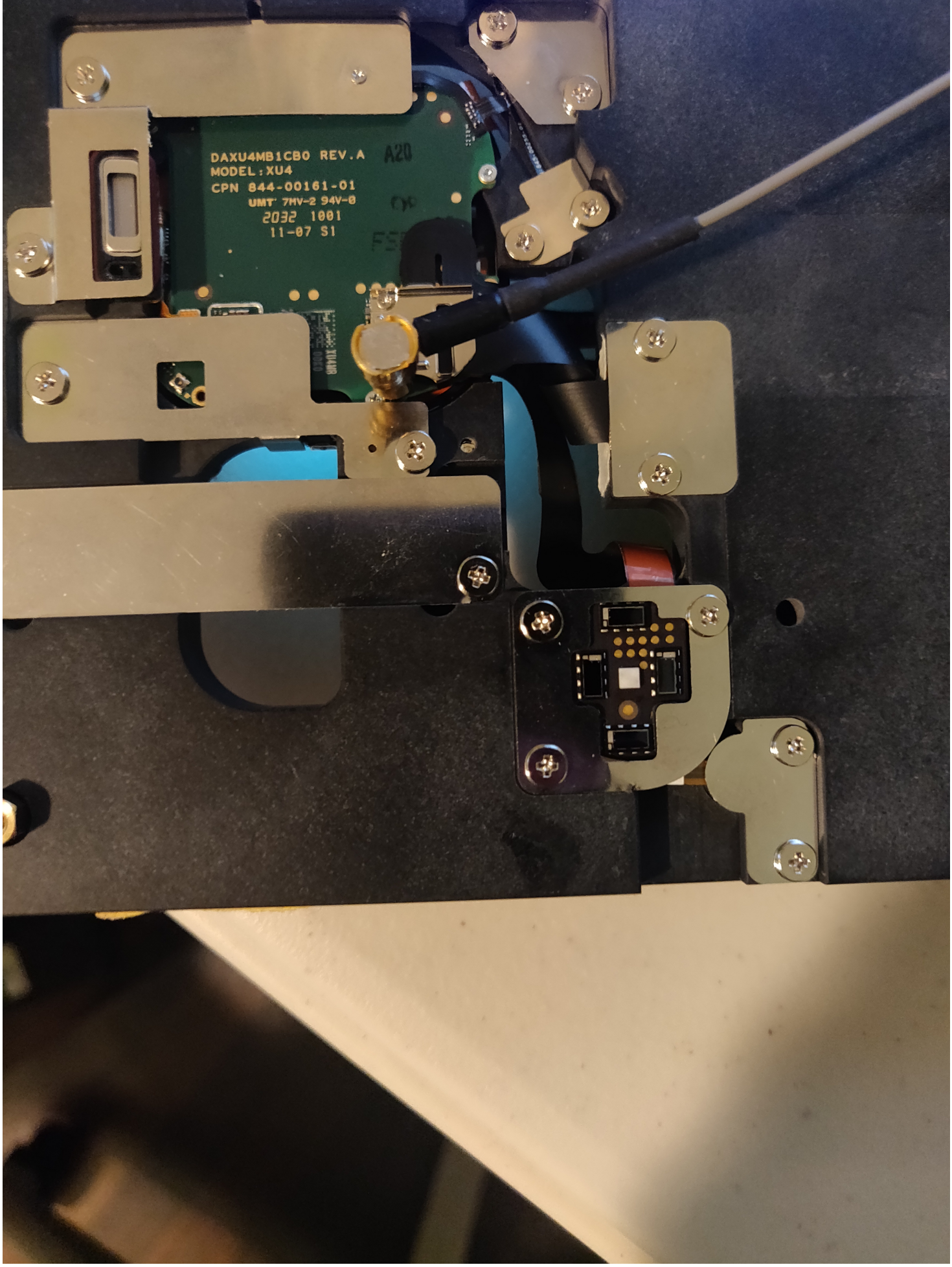
> *The POC-NED is known to have weak WiFi/BT connectivity due to internal-antenna design issues. Please consider using an external antenna or bring your wireless connection closer to the device to aid in development (physically move closer to router or host a mobile hotspot). https://fburl.com/br6wel22*

Setup

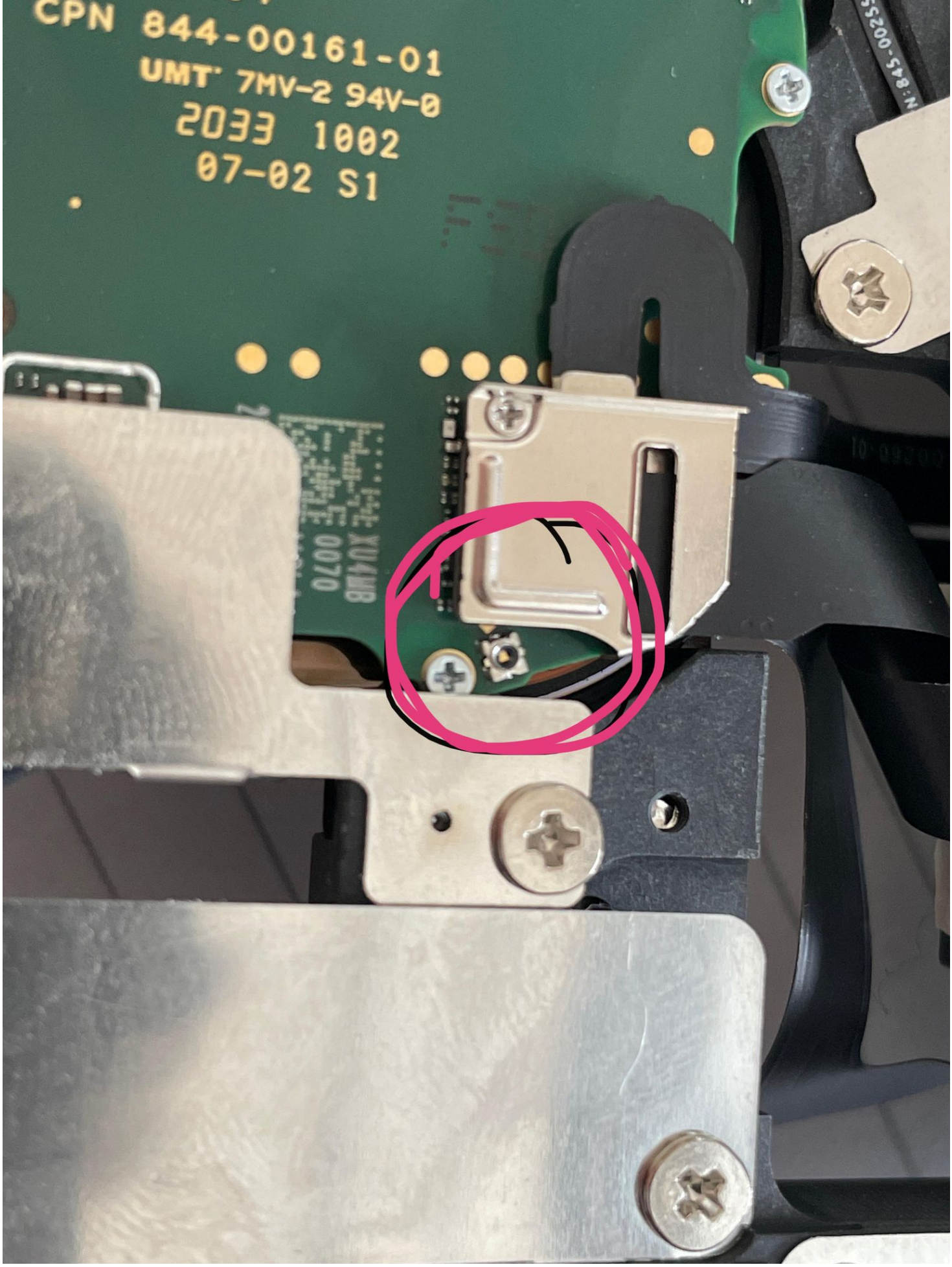
See more info on **eSim profile switching**, **alternate profile testing**, **nano Sim rework**, and **GPS testing instructions** *here*

Note: This is currently tested on the Quanta SW image only

Note: To get GPS to work you will have to attach the antenna to the bottom of the board right behind the display.



The spot to attach antenna is hard to spot, so here it is:



you will hear a click when the antenna will go in

Sensor Subsystem

Sensor HAL

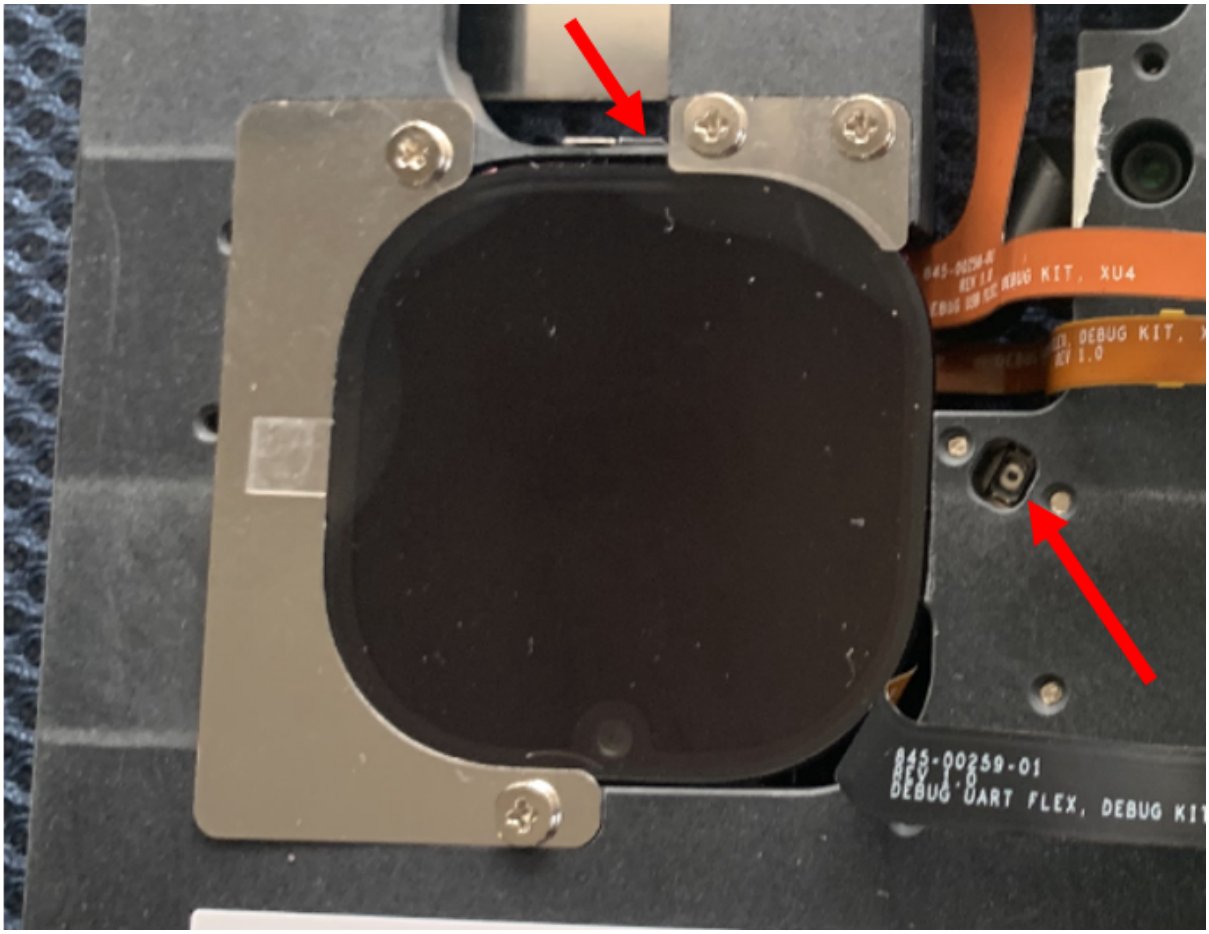
See more info on setting up the sensor HAL here

Note: This is currently tested on the Quanta SW image only

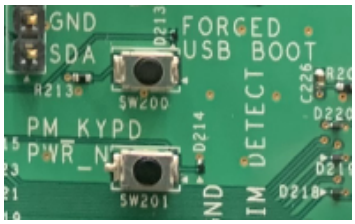
Peripherals

Buttons

There are two buttons in the form factor, these can be accessed on the NED above (capture) and to the right (power) of the display.



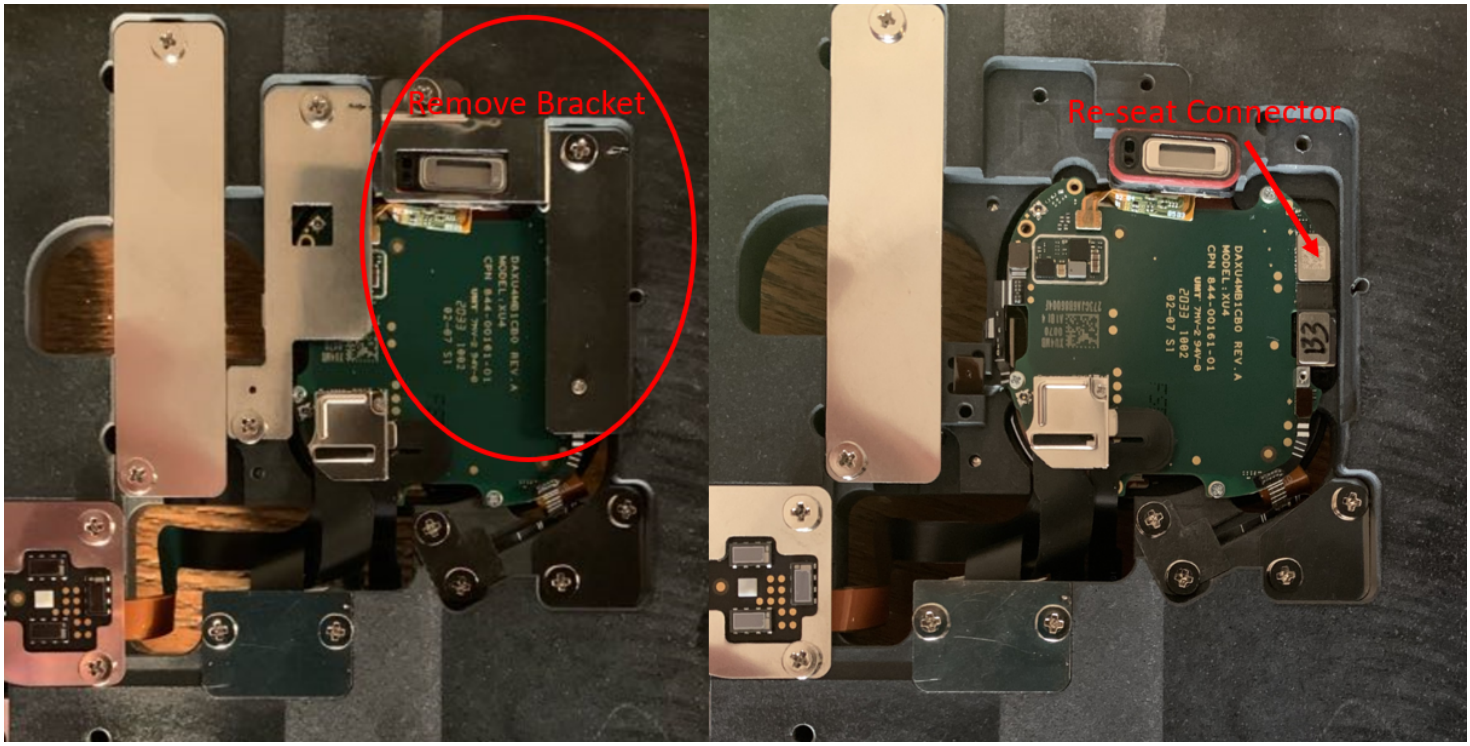
There are two debug buttons in NED, they can be found on the debug board. They are power and USB_BOOT



Cameras

If you cannot find FCAM in the camera app once an image supporting camera drivers is loaded (see above section on flashing), it is likely that the connector has popped off during shipment.

FCAM CONN:

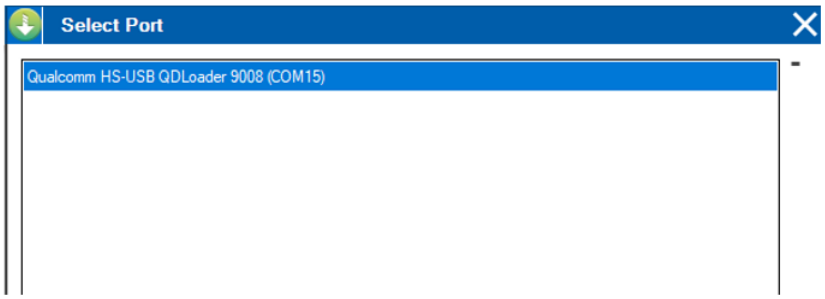


Cables

In the POC ned package, there are 2 USB cables. The USB cable has a red button. The purpose for the red button is that short USB D+ and GND. when USB D+ and GND are shorted, it can force device enter emergency download when power on.

There are 2 USB cable, one is typeC - typeC, the other one is typeC – typeA. When plugging the typeC-typeC USB cable, the device does not power on, you need to switch the direction. So we recommend using typeA-typeC USB cable.

Boot up normally and then read USB
Unit will show up as Diagnostics 90B8 in QFIL

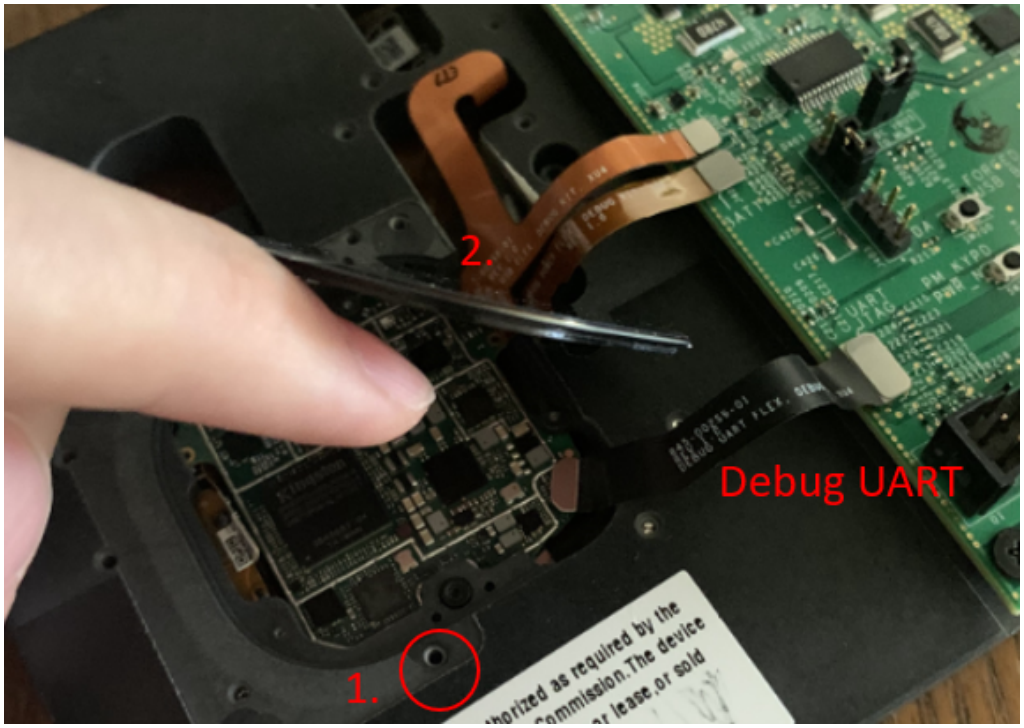


1.Connect fake battery (uUSB)
2.Connect “shorted/pressed” USB-C line
3.Obsever uUSB pull ~130mA
4.Unplug USB-C, “unshort/unpress” USB-C line
5.Connect USB-C, see EDL mode
Unit will show up as QDLoader 9008 in QFIL

Debug UART Flex Repair:

In case the debug uart flex pops off, it is possible to re-assemble the flex back onto the board.

1. Remove display brackets.
2. Gently lift the display, note it is hinged on one corner.



Modifications:

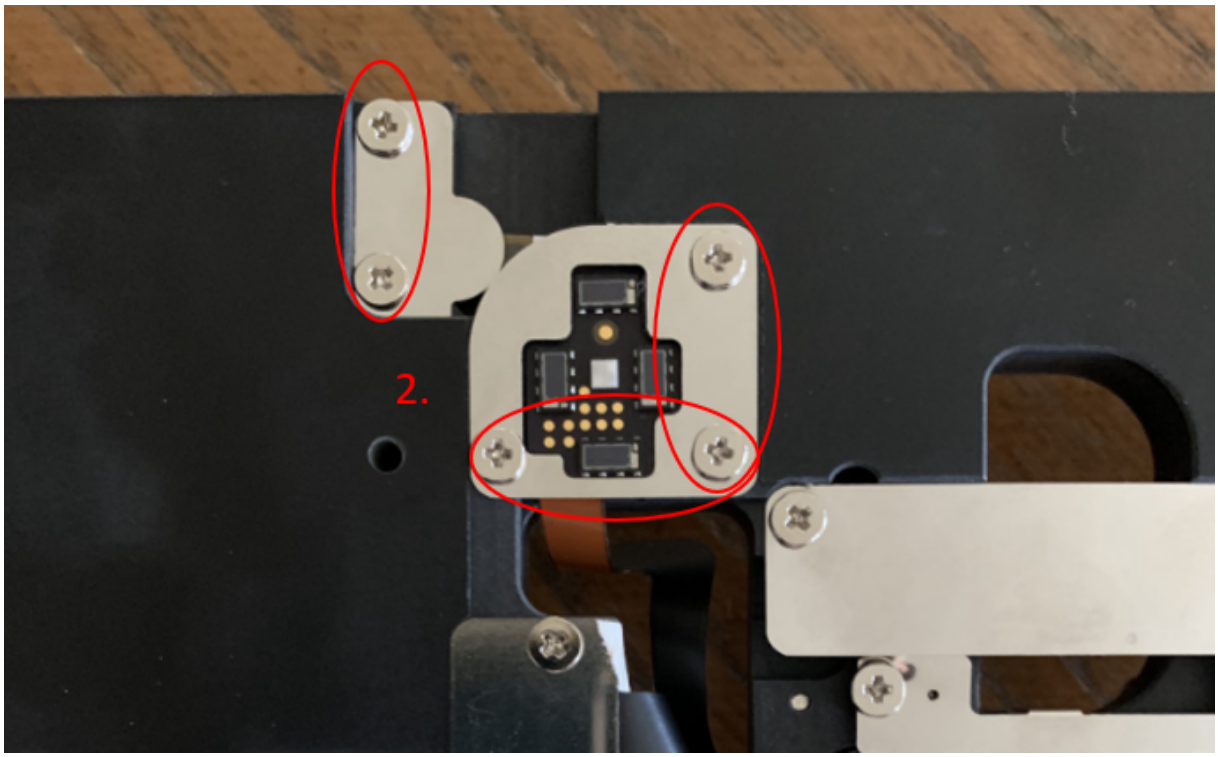
Sensor Board Replacement

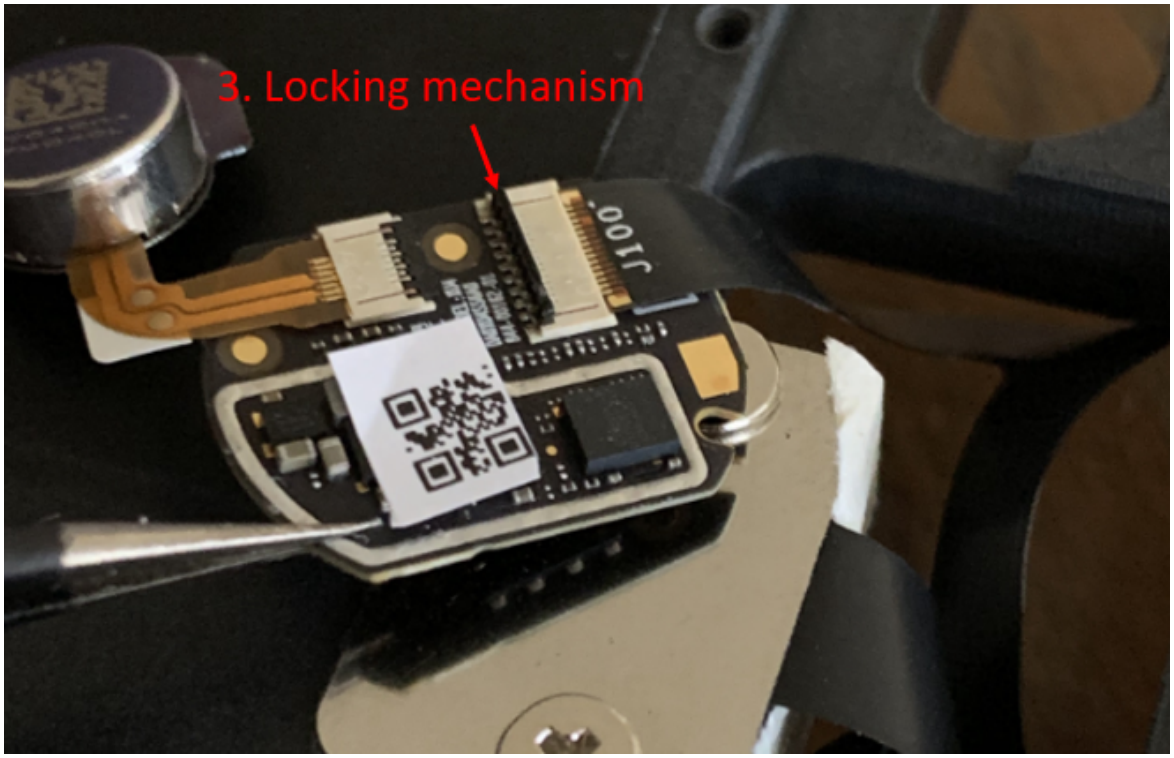
The initial batch of NEDs all came with sensor boards that had LSM6DSO32 IMUs, these IMUs do not have the ML core and are not POR. All NED users should be receiving a new replacement sensor board with the POR intent LSM6DSOX IMU.

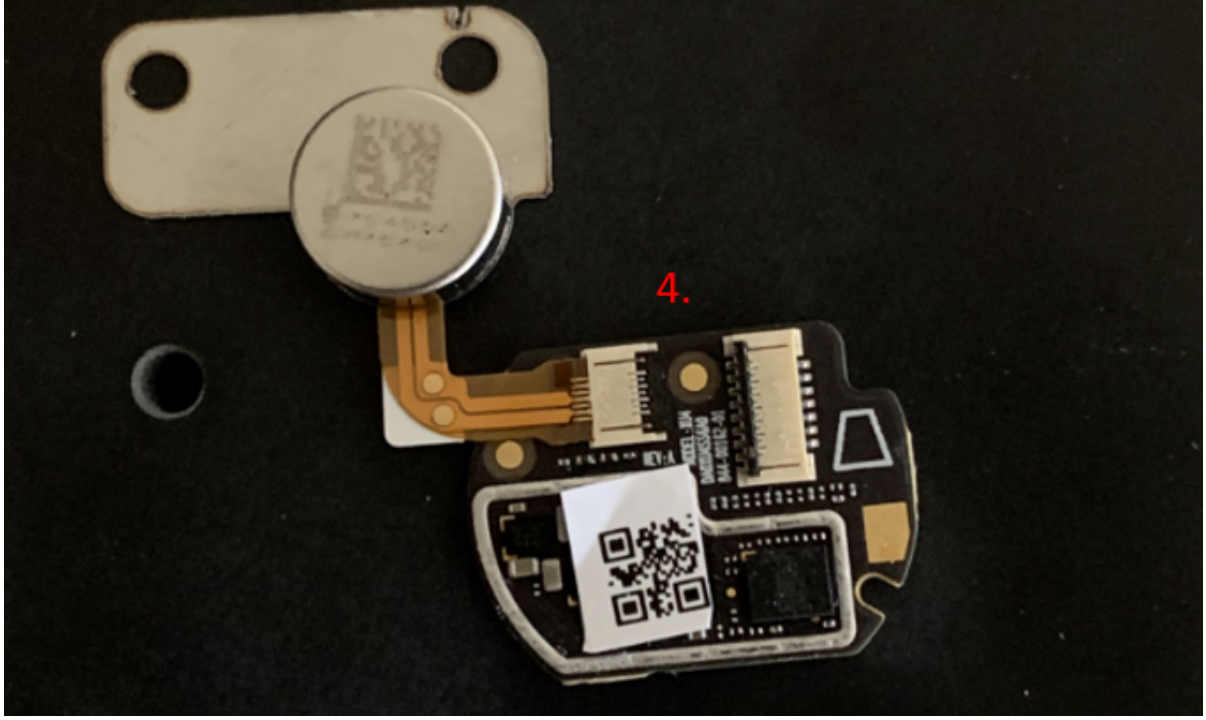
To replace the sensor board:

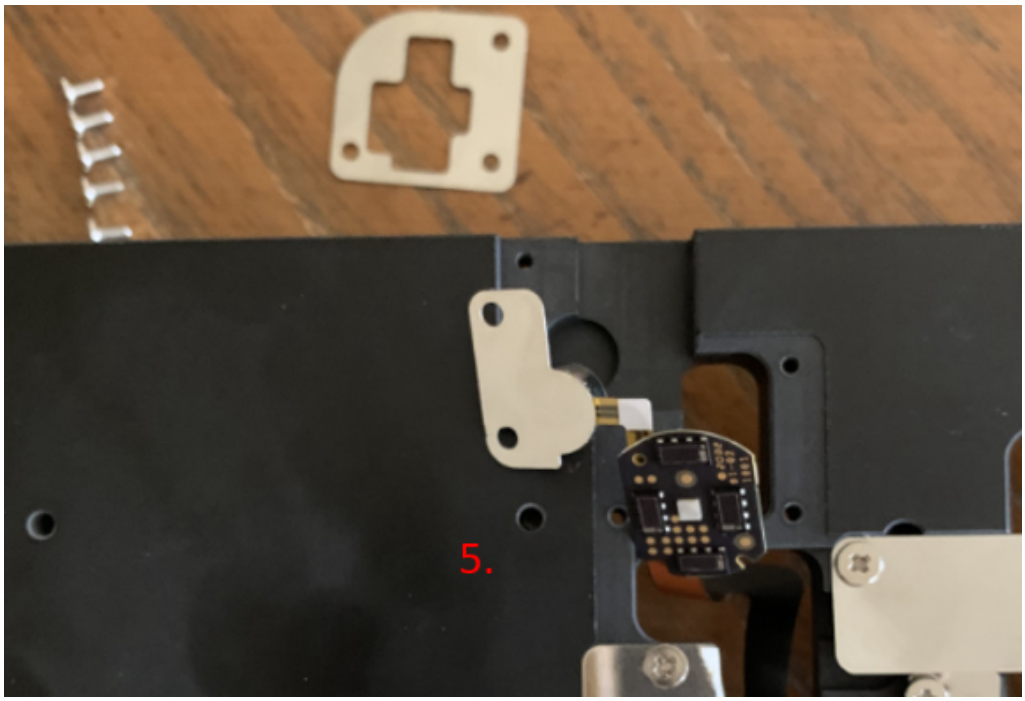
1. Flip to the backside of the NED.
2. Remove Sensor board and LRA motor brackets.
3. Remove ZIF connectors on the Sensor board, note: there are locking mechanisms that need to be flipped first.
4. Assemble the new Sensor board with the same ZIFs

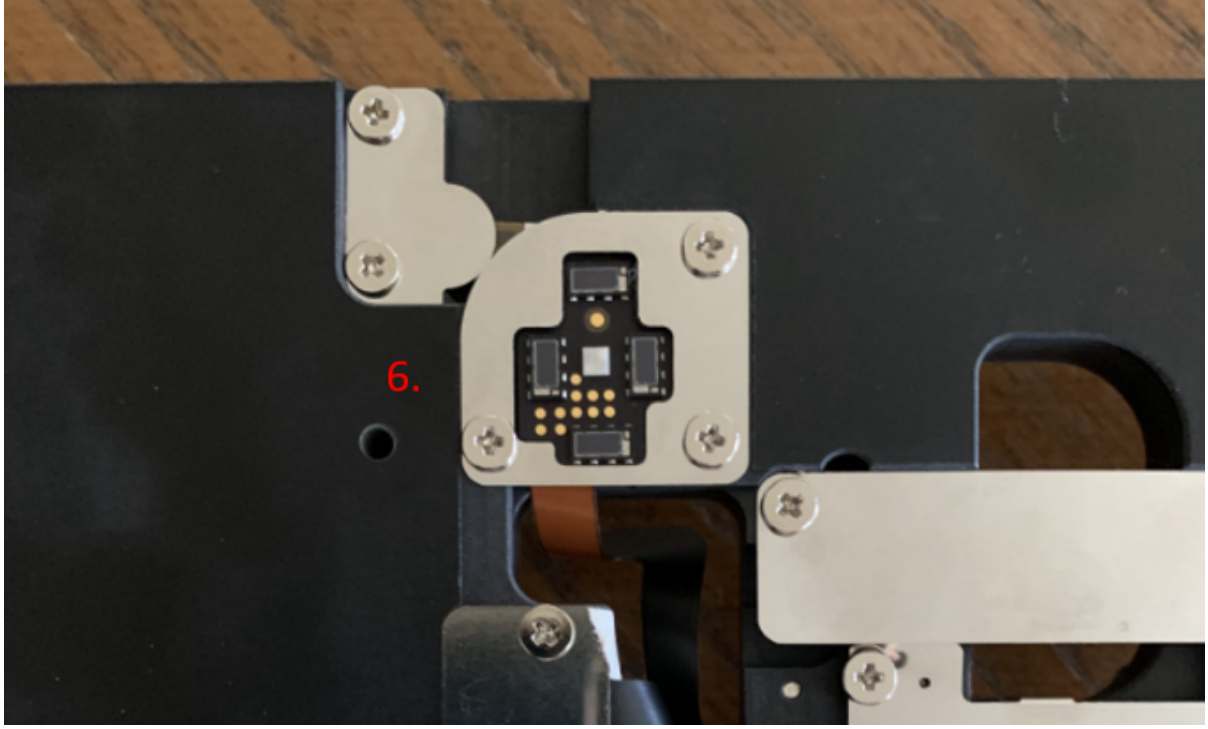
5. Add sensor board back to the holder
6. Add brackets back.






4.


5.



Additional Info:

NEDs are electrically the same as form-factor in terms of components and displays, but laid out on an acrylic case instead of being assembled into watch. These will come with POR-size displays. *Think of these as a 'board' version of the POC.*

POC is a limited quantity build, as we're not yet able to make large quantity if HW in our factory (in fact, this is the very first HW to be built in our contract manufacturer's factory). For SW, we have 35 FF and 100 NEDs to share and use amongst us.

There are some components that are not POR (e.g. battery, camera auto focus actuator, speaker quality etc.)

Since we have a limited quantity of this HW, it would be really helpful **if folks who're receiving POC NEDs and don't necessarily need EDBs can return their EDBs** so those can be backfilled to new hires or folks who don't have HW to use now. POC NED shipments should come with return labels - place the EDB back in the same box to return it back to our warehouse.

« POC Allocations (SW)

Live Network Access »

)}