Thunderbird

… is a customized firmware for Yuneec Typhoon H (aka H480) based on PX4 autopilot, developed by **Toni Rosendahl**. Introduction see:
https://yuneecpilots.com/threads/typhoon-h-480-px4-v1-10-stability-issues.18205/page-3

The firmware is Open Source.
The project can be found here: https://github.com/tonirosendahl/Thunderbird
PX4 Autopilot: https://docs.px4.io/

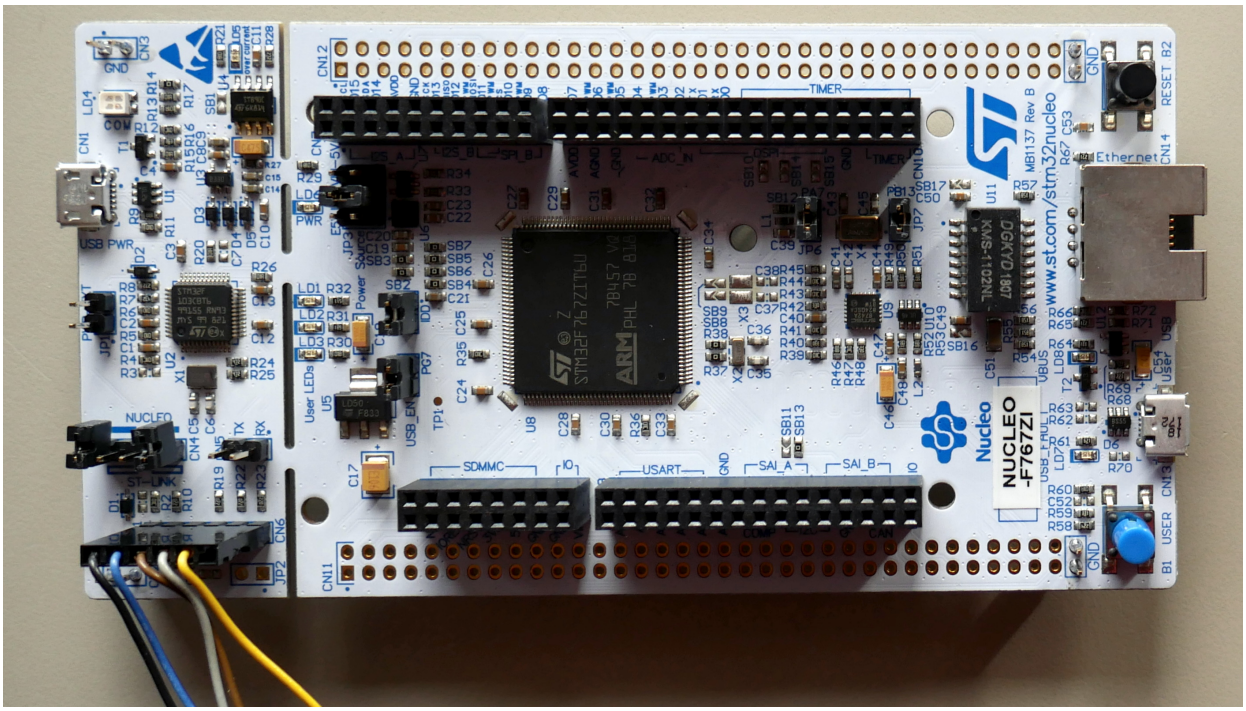
How to flash the MCU board of the Typhoon H

⚠**Note: Once the bootloader was updated there is no way back. The Typhoon H is going to be the Thunderbird from now on.**

What we need:

Download and unzip binaries:
https://github.com/tonirosendahl/Thunderbird/blob/Typhoon_H_480/Thunderbird_19122019_FT.zip

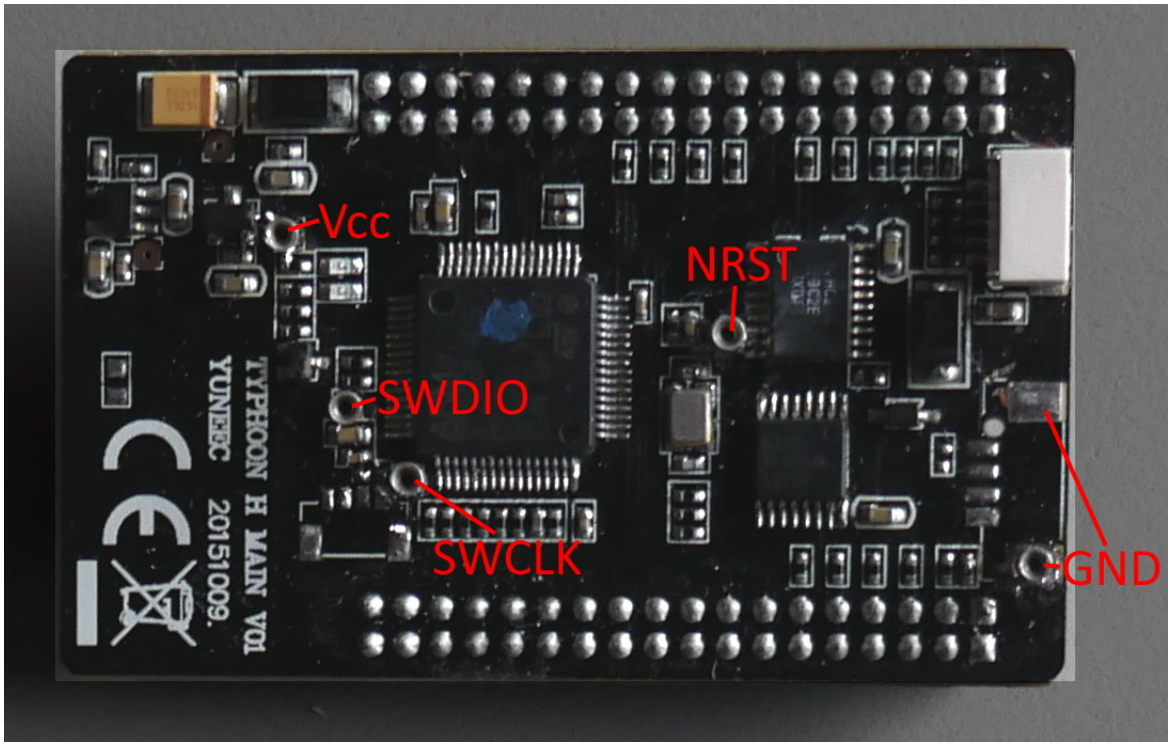
Get a programmer for STMxx. We need a ST-Link V2-1. Something like that:
https://www.st.com/en/evaluation-tools/nucleo-f767zi.html



Version 1.0 from 2020-01-12

Also we need the related flash tool, the ST-Link utility. Download and install it.
https://www.st.com/en/development-tools/stsw-link004.html

And of course the MCU board from Typhoon H. It's recommended to have it as spare part just to keep the original board to fall back to legacy Typhoon H.

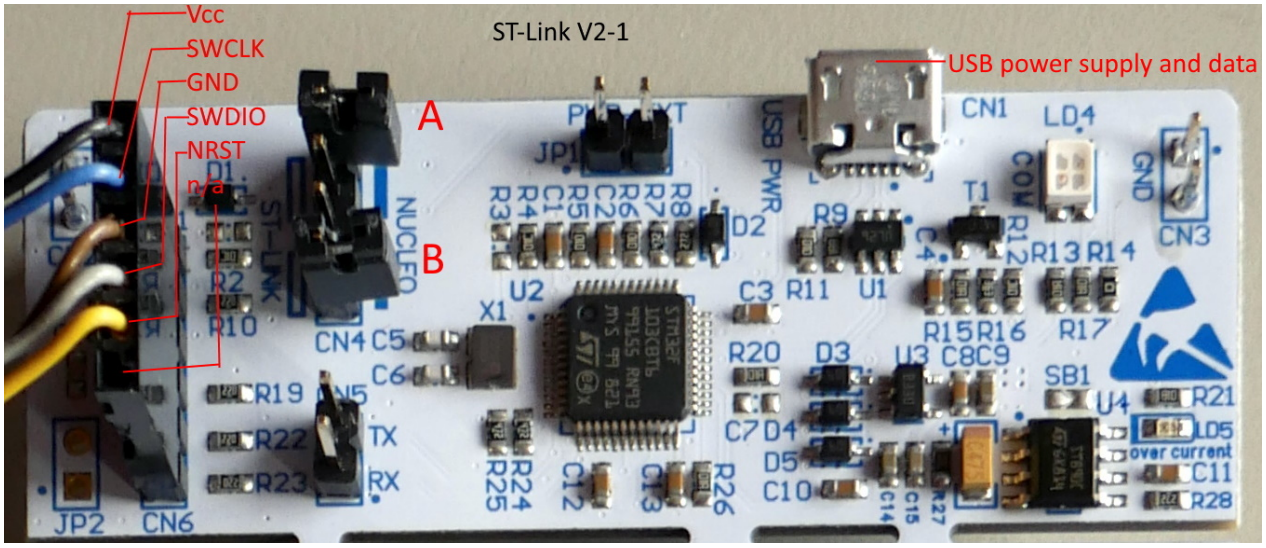


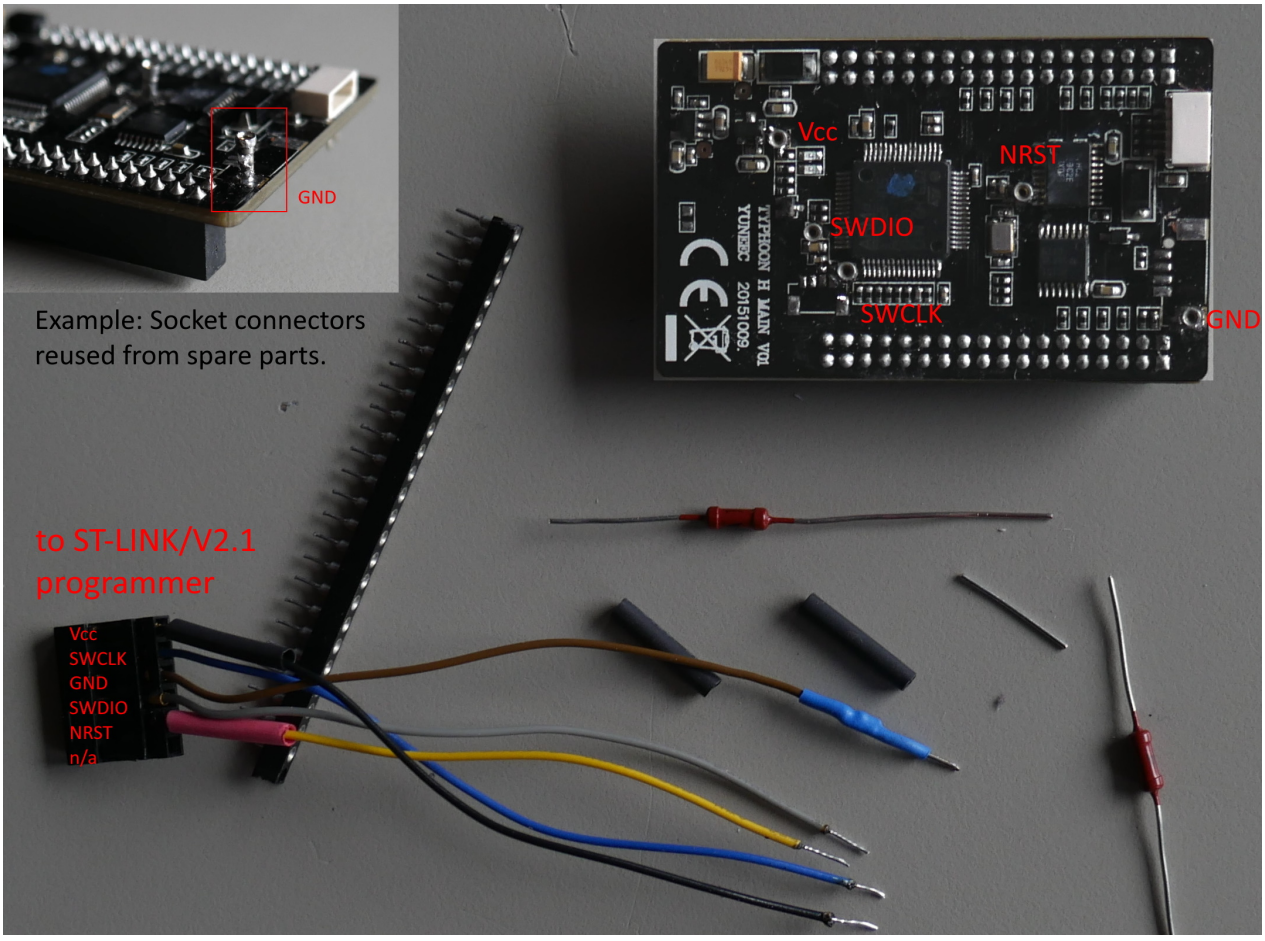
To set up parameters and update the drone we need PX4 configuration tool "QGroundControl". Download and install it. It needs Administrator rights at least for Windows10.
http://qgroundcontrol.com/downloads/

⚠**Note:** **Before we go ahead, download and read the documentation for the stuff above.**

Step 1: Wiring for the programmer (the most dangerous step)

We need Ground (GND), +3.3V (Vcc), clock (SWCLK), data (SWDIO) and reset (NRST). For all those connections are small measuring points available on the MCU board. It's up to you how to connect those five wires to the SWD port at the ST-Link (soldering wires or tiny socket connectors).





Step 2: Replace the original bootloader by the new one (no way back!)

Both CN4 jumper A and B have to be off (not connected) to enable SWD port to flash external processors.

Connect MCU board via USB to PC to keep power during flashing.

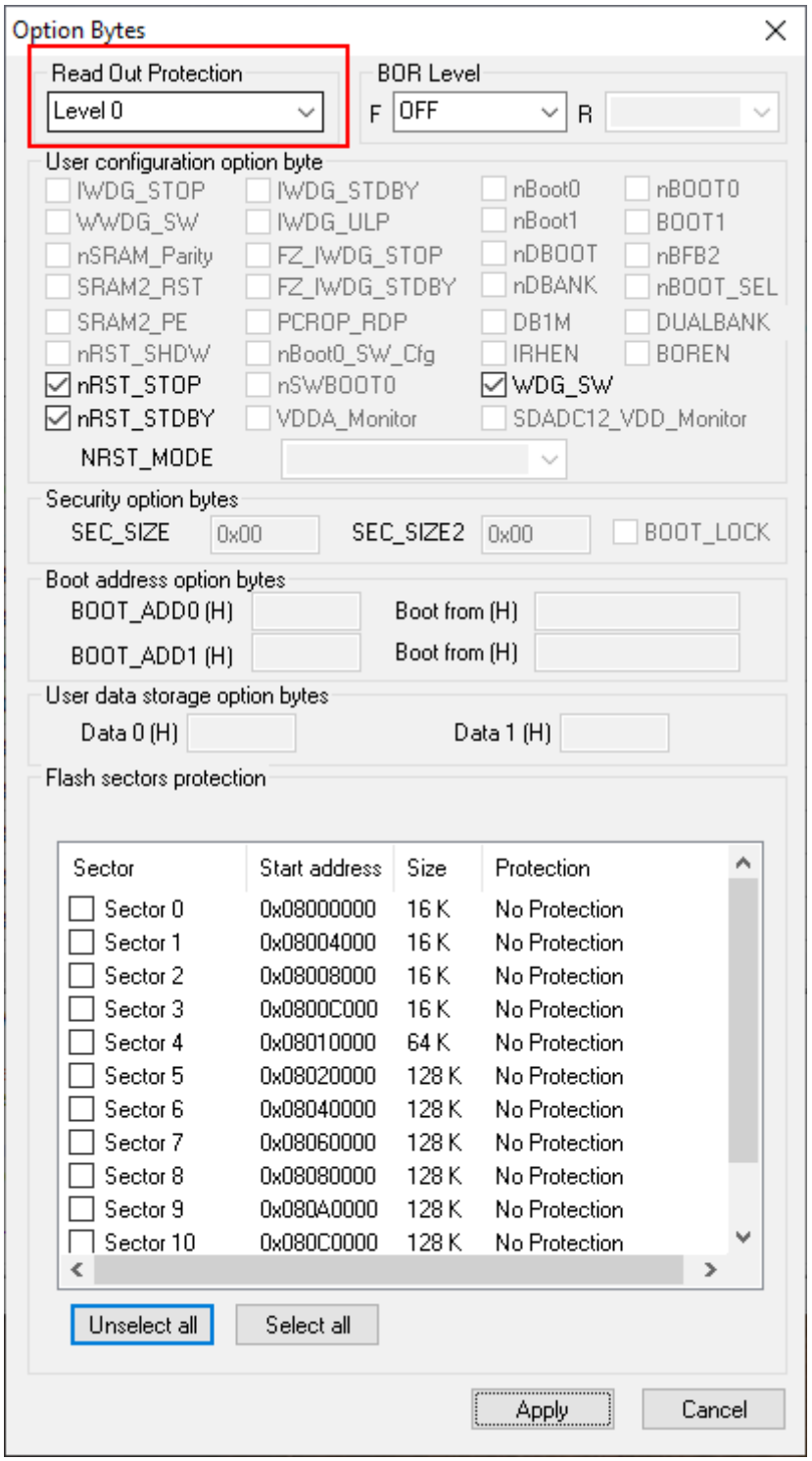
Connect SWD port with measuring points at MCU board.

Start the STM32 ST-Link utility.

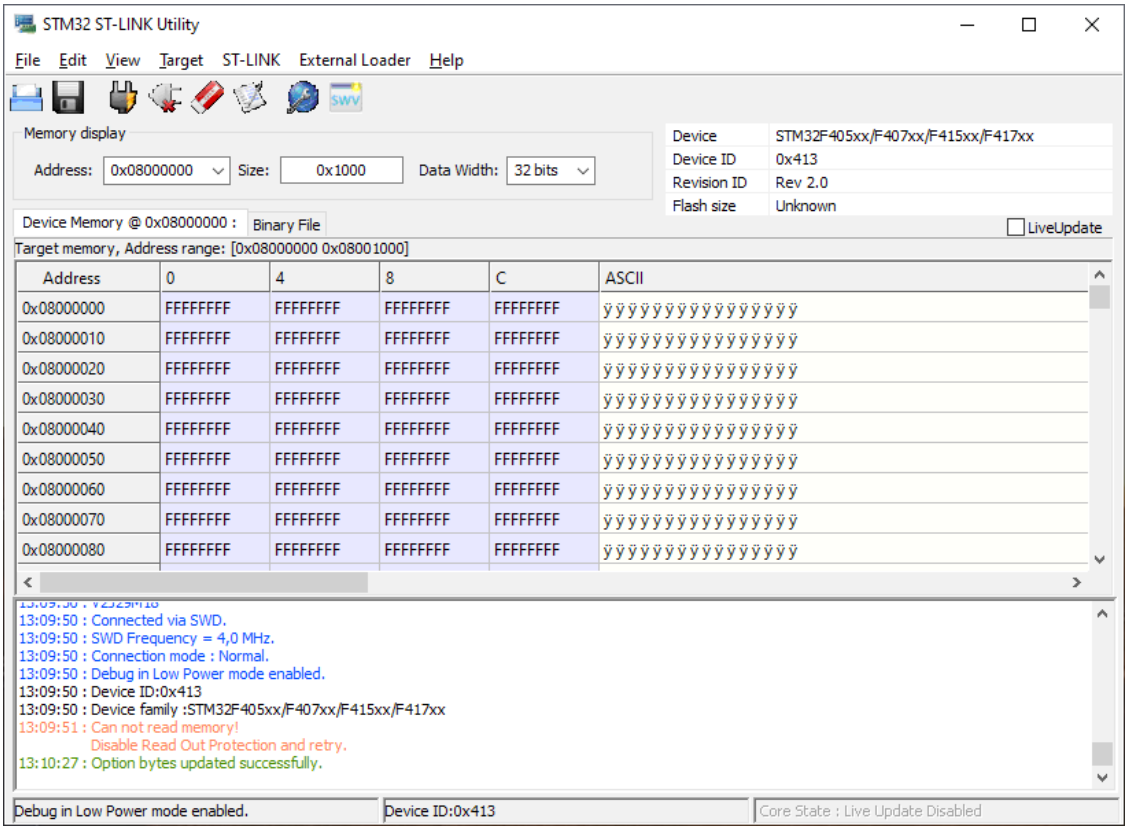
Go to Target > Connect to verify that the MCU is connected. If not, check the wiring.

Set Read Out Protection to Level 0:
Target > Option Bytes…
Select "Level 0" > Apply.

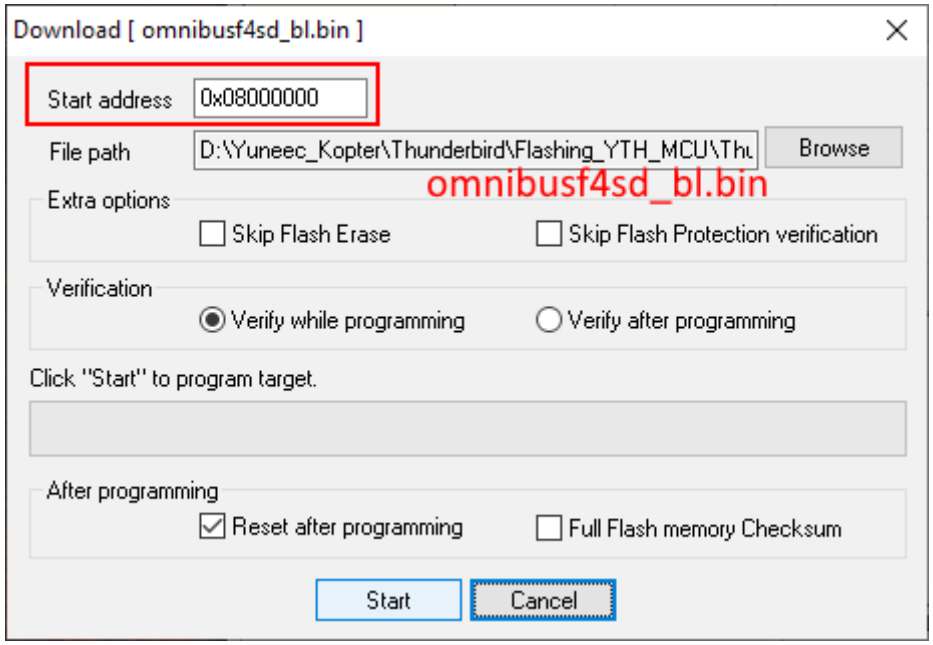
Now the old bootloader will be removed.



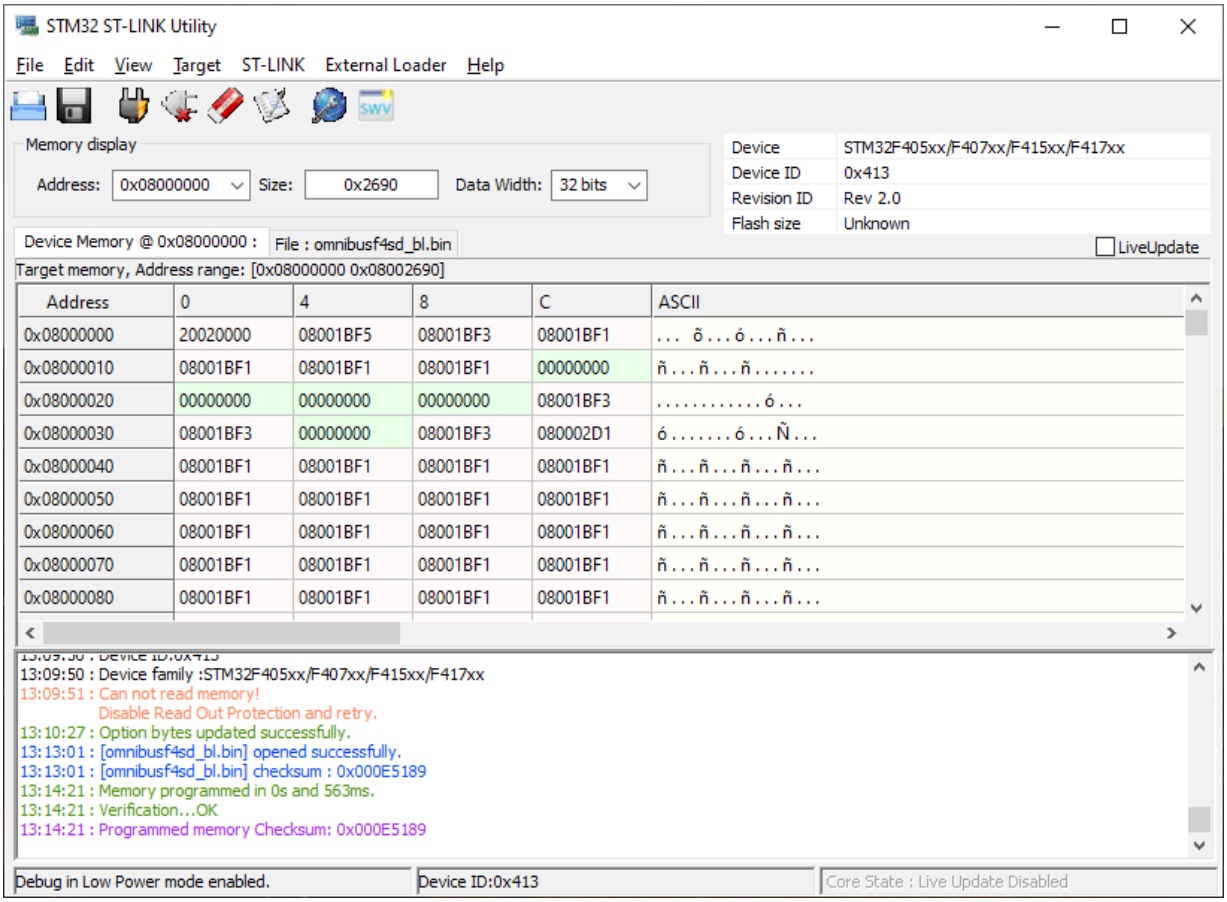
If success it looks like this:



Now let's flash the new bootloader to address 0x08000000. Go to Target > Program & Verify. Select "**omnibusf4sd_bl.bin**" from downloaded binaries > Start.



If success it looks like this:



The MCU has now a PX4 bootloader, placed in the beginning of the Flash, 0x8000000. This is where the ROM bootloader starts up, whatever there is. There is no way to brick the STM32, the ROM bootloader will always be there, and the bootloader you just flashed, is a "second stage" bootloader, which is started by the ROM bootloader. The MCU or the ROM bootloader does not actually care what is placed into 0x8000000, whatever there is, gets started.

The actual PX4 firmware is compiled and linked so, that it starts from 0x8008000. That gives some space to save parameters, between the bootloader and the PX4 main application. The linker script, if interested, is at Thunderbird/boards/yuneec/typhoon_h/nuttx-config/scripts/script.ld but you should not need to modify it. If you want to ditch the PX4 bootloader and save parameters on a SD-card, then this is the place to modify the PX4 to start from 0x8000000. But for now, do not modify anything there.

Step 3: Flash the main application

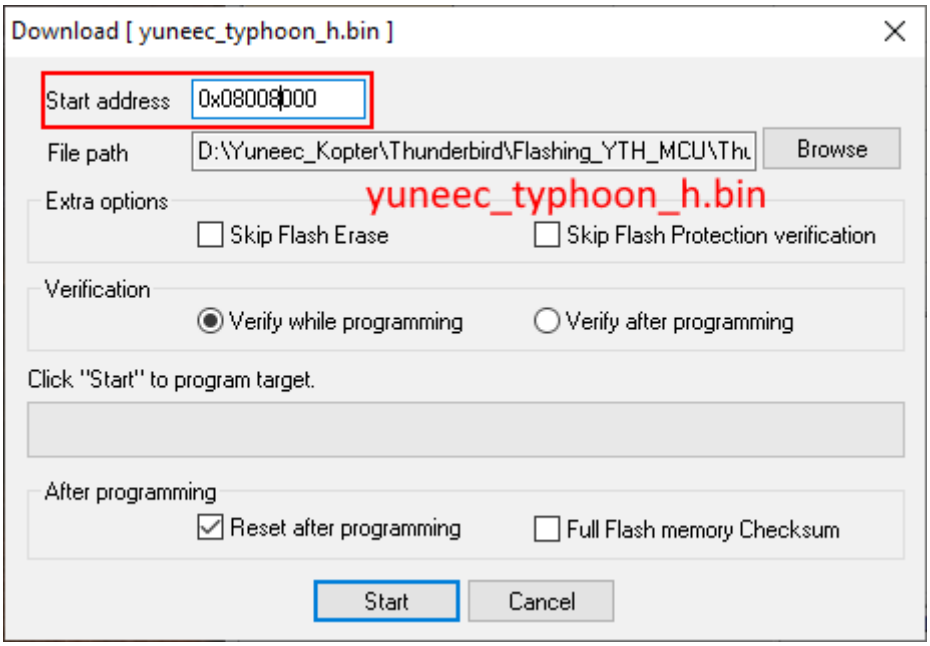
Now, there are two ways to flash the main application.

A) The bootloader way: The PX4 bootloader you just flashed, is capable of flashing the main application via Typhoon's USB and then start it, from address 0x8008000. No need to use the ST-Link any more; you can close the drone's lid. The bootloader has few seconds delay for flashing. The flashing is done using a slightly modified px_uploader.py script. Clone my repository to get these. This script requires the .bin file to be converted to .fw file (also in the zip) after compiling. See the "flash_typhoon_bootloader.py" script, write your COM port there, and while the script is running, connect the Typhoon to your computer. I do my development in Ubuntu 16.04 LTS, I don't know how this works in Windows, I'll write soon how to setup a nice development environment, too. ;-) However, if this gives you issues, there is also...

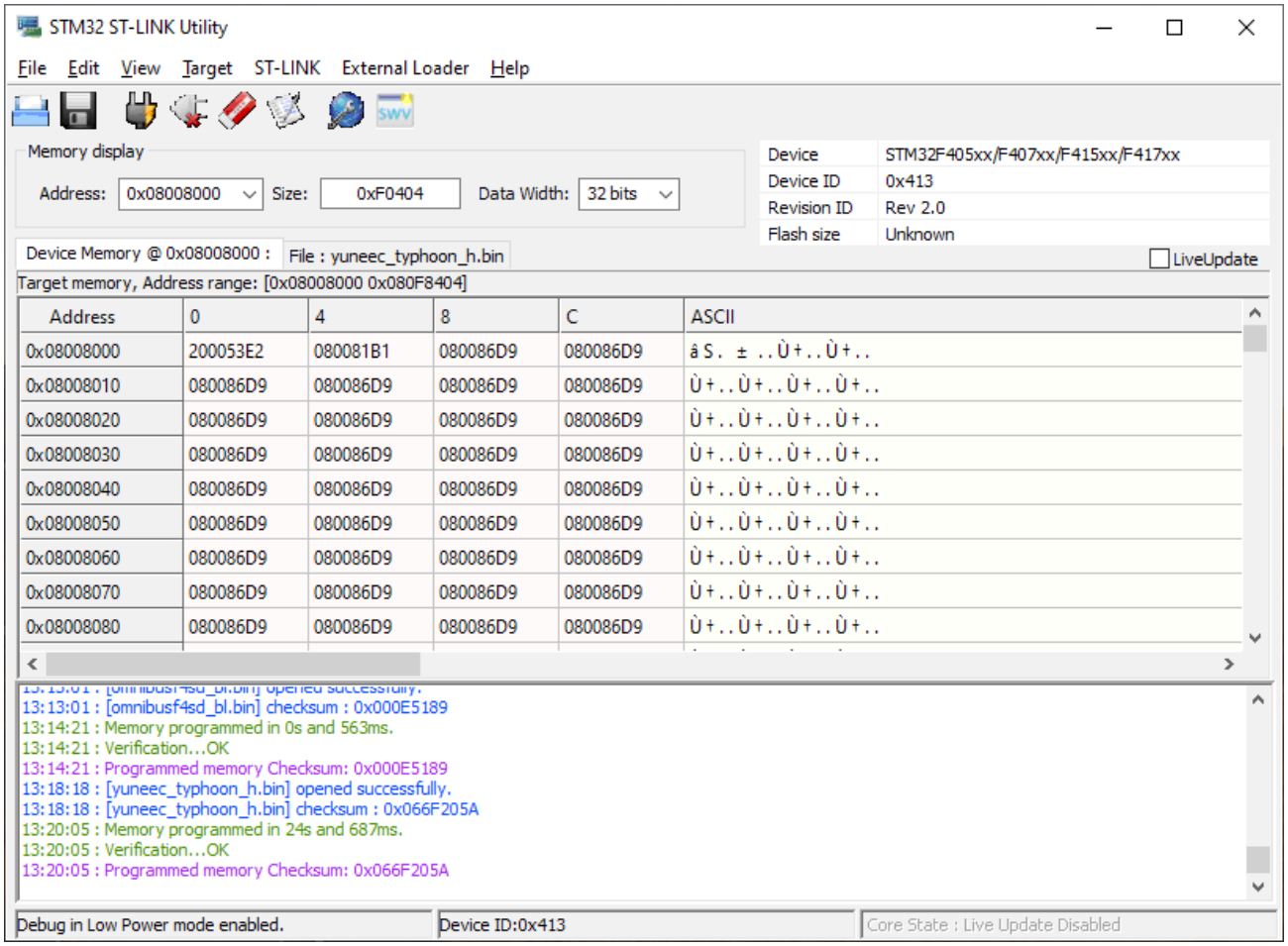
B) The ST-Utility-way (aka "my python scripts suck" -way). Flash the "yuneec_typhoon_h.bin" to the MCU, the same way you flashed the bootloader, but this time, starting from 0x8008000.

Option A) is for later firmware updates. Because we are still connected to the programmer interface, we use option B)

Go to Target > Program & Verify. Select "**yuneec_typhoon_h.bin**" from downloaded binaries.
Set Start address to 0x8008000 and start flashing.



If success it looks like this:



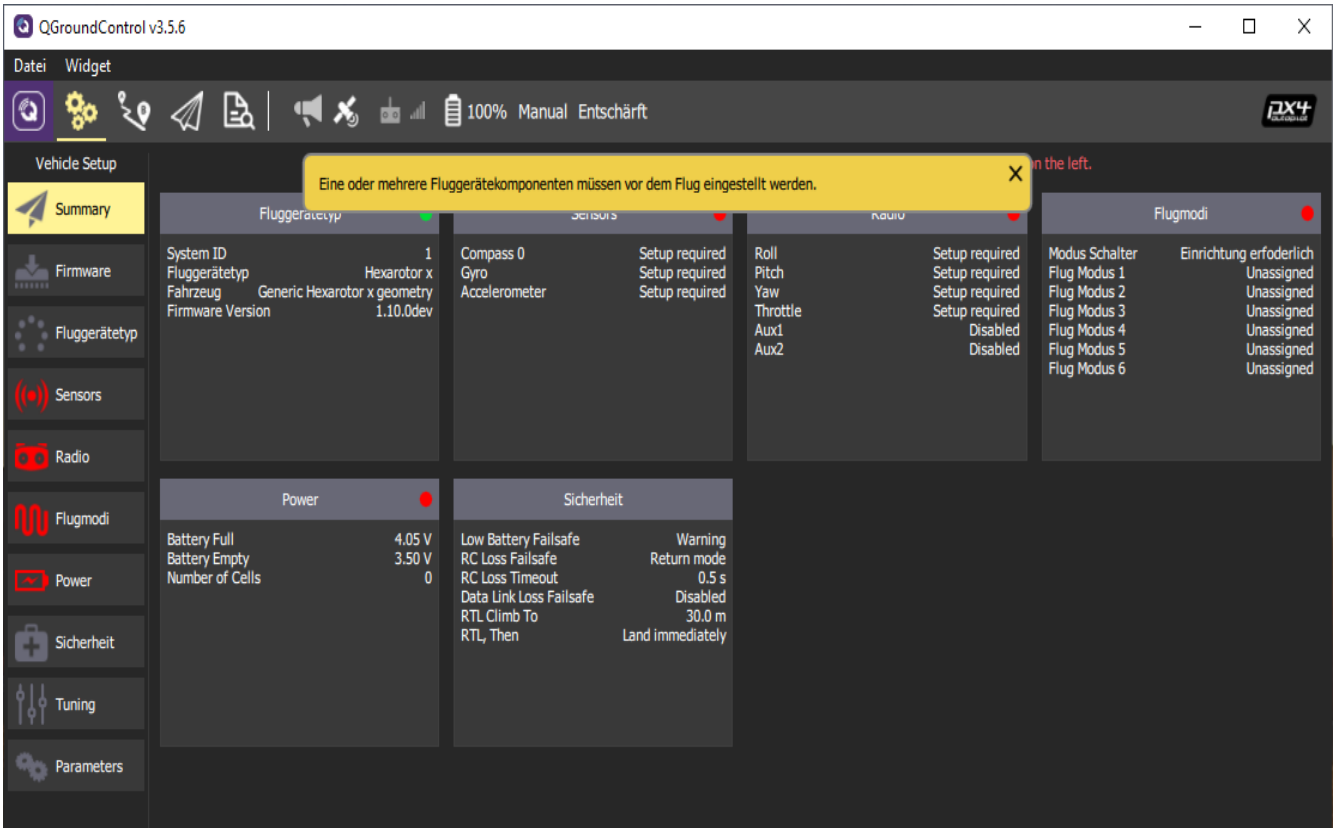
Done!

Disconnect all USB connections and power down. Remove programmer wiring. ST-Link is not needed anymore except a new bootloader is needed. This may happen during this early phase of project development.

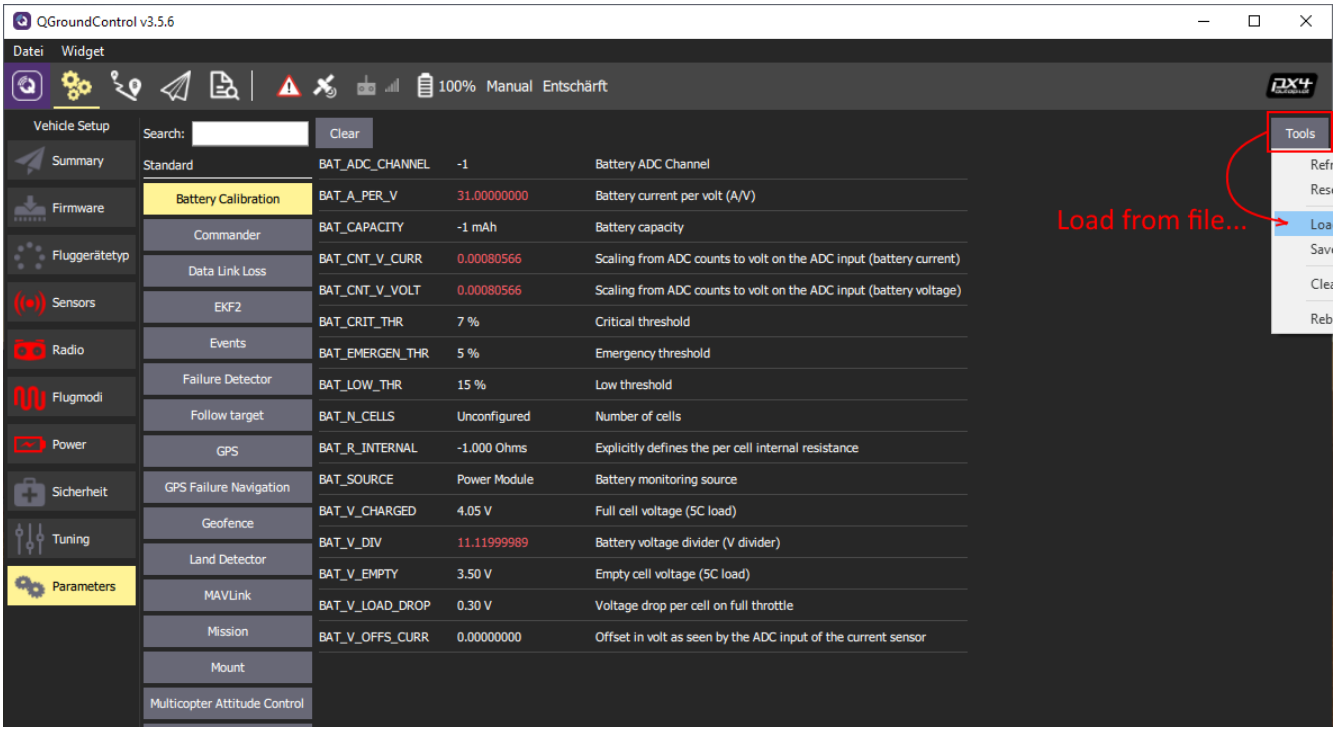
Reboot the drone. The drone's power button has a delay of ~8 seconds. During those, the button must be kept pressed. When all lights illuminate, release the power button.
The drone is turned off by removing the battery.

Step 4: Set drone parameters

Install the Qgroundcontrol (aka QGC). Start QGroundControl as Administrator. Power up the drone. Connect the USB.



Load parameter file "**typhoon_h_parameters.params**" using QGC's parameter page. Go to parameters > Tools > Load from file…



Step 5: Set hardware and calibrate sensors

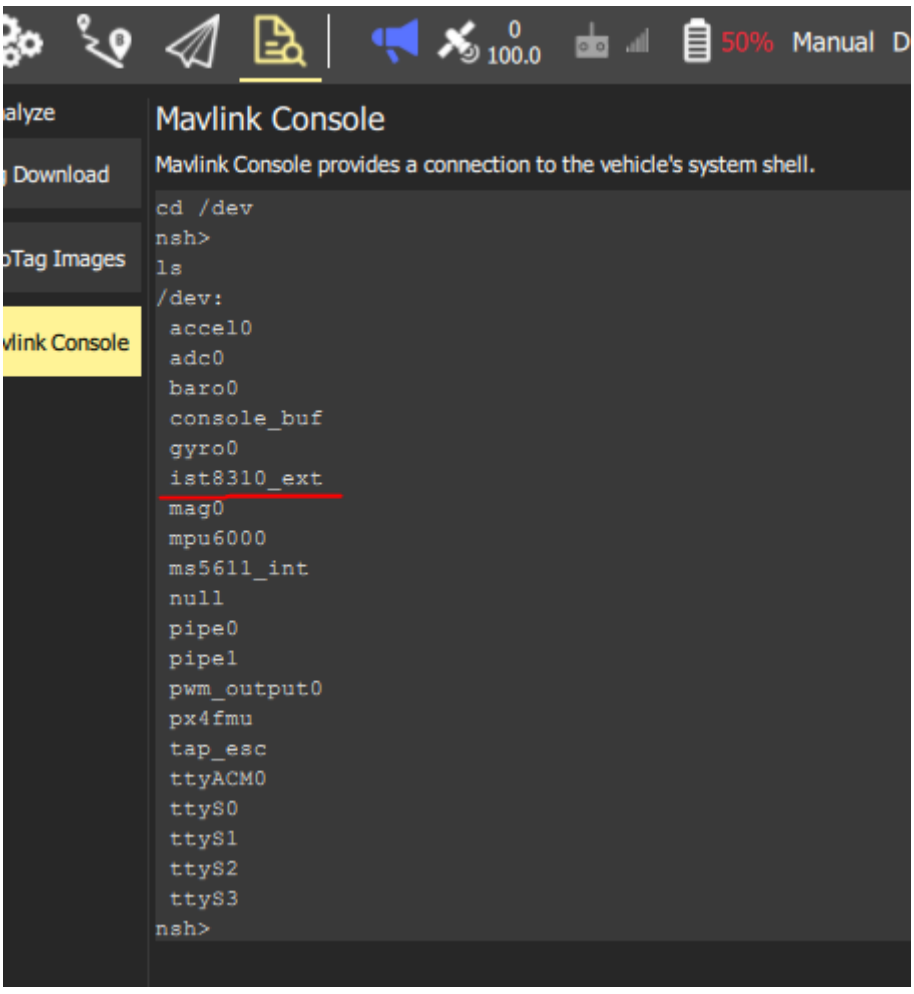
For Typhoon H exists two different compass chips. Older GPS boards have HMC5883, newer boards have IST8310 applied as compass chip. To find out which one you have, connect the drone to QGC and go to Mavlink console:

Type:

```
cd /dev
ls
```

and you will get a list of drivers for the hardware.

Nice to see what is there. For compass it could be "hmc5883_ext" or "ist8310_ext".

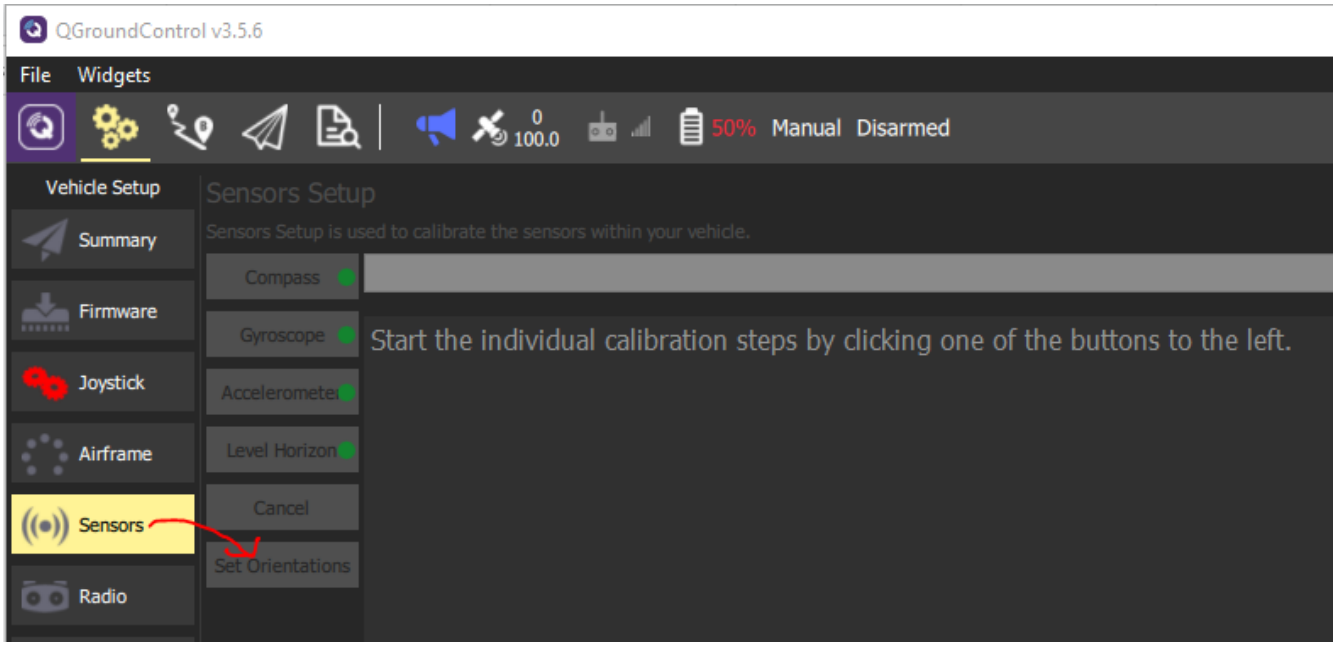


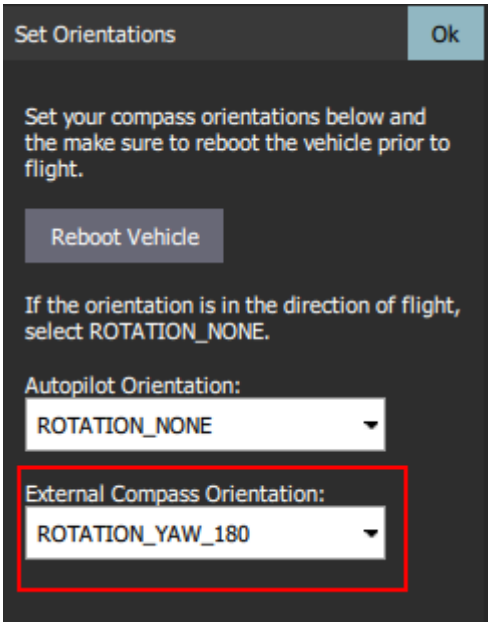
Important: If you have the "hmc5883_ext", you have to change External Compass Orientation to "ROTATION_YAW_270".

"ist8310_ext" has to have "ROTATION_YAW_180".

Never change Autopilot orientation. It must be set to "ROTATION_NONE".

Go to Settings > Sensors > Set Orientation





Then set External Compass Orientation to the correct value depending on you compass hardware.

Do sensor calibrations (don't use default settings) like you would do for a fresh PX4 drone (see Basic Configuration · PX4 v1.9.0 User Guide). **Do not try to calibrate ESC's.** It will not work and is not required.
https://docs.px4.io/master/en/config/

Verify settings, check calibrations and sensors. In my settings, the top position of the flight mode switch is Stabilize (Typhoon without GPS), middle position is, well, Position (Typhoon with the GPS) and bottom one is RTH. The acro/rattitude mode are not there, be careful with those. The red button does NOT arm or disarm the drone. How to start it - you have to find out this by yourself by going through the PX4 documents for this one (As it is highly recommended to read how to operate the PX4, I want you to take a look at them).
https://docs.px4.io/master/en/flying/

Arming and disarming (this is important to know!) requires the speed selector to be in FAST mode. You do not have to do anything in ST16 side. Stock settings there are OK.
If you want to re-bind, run "**`typhoon_bind start`**" command on the PX4's command line. Bind and reboot. Go flying and have fun!

Troubleshooting

- Check if the drone is booting up correctly after soldering the connections at the measuring points on the MCB board. If not, oh-oh! Please check with a magnifying glass for solder bridges.

- If the drone does not appear as an USB device when the USB is connected AND the drone is powered on: Well, all is lost now. You are having a bootloader flashing issue OR the main autopilot firmware got accidentally flashed to 0x8000000. Re-flash everything from the beginning and it will be fine.

- If the drone appears as an USB device but it does not start after a long press of the power button: The main firmware flashing went wrong. Re-flash, starting from 0x8008000 (Step 3).