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. **Please join the community and contribute!**
The project can be found here: https://github.com/tonirosendahl/Thunderbird
PX4 Autopilot: https://docs.px4.io/

This is a description how we transform a stock Yuneec Typhoon H into a Thunderbird.

Table of content

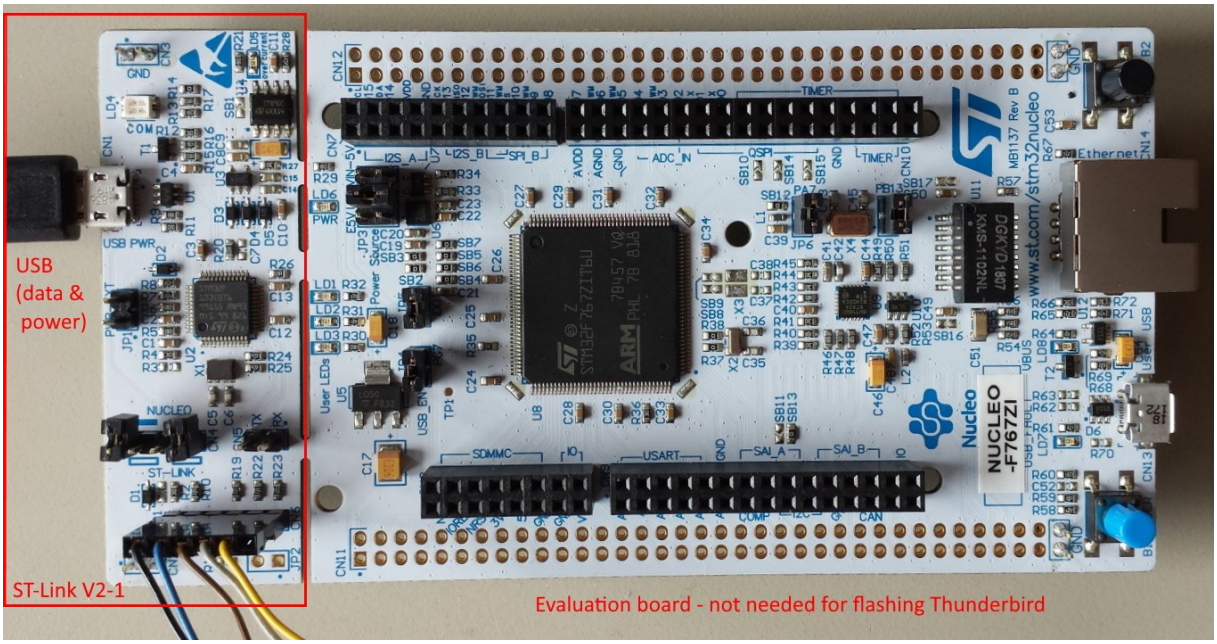
How to flash the MCU board of the Typhoon H

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

Preparation

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



Please note that there are other (and maybe cheaper) evaluations boards on the market that have ST-Link V2-1 programmer onboard.

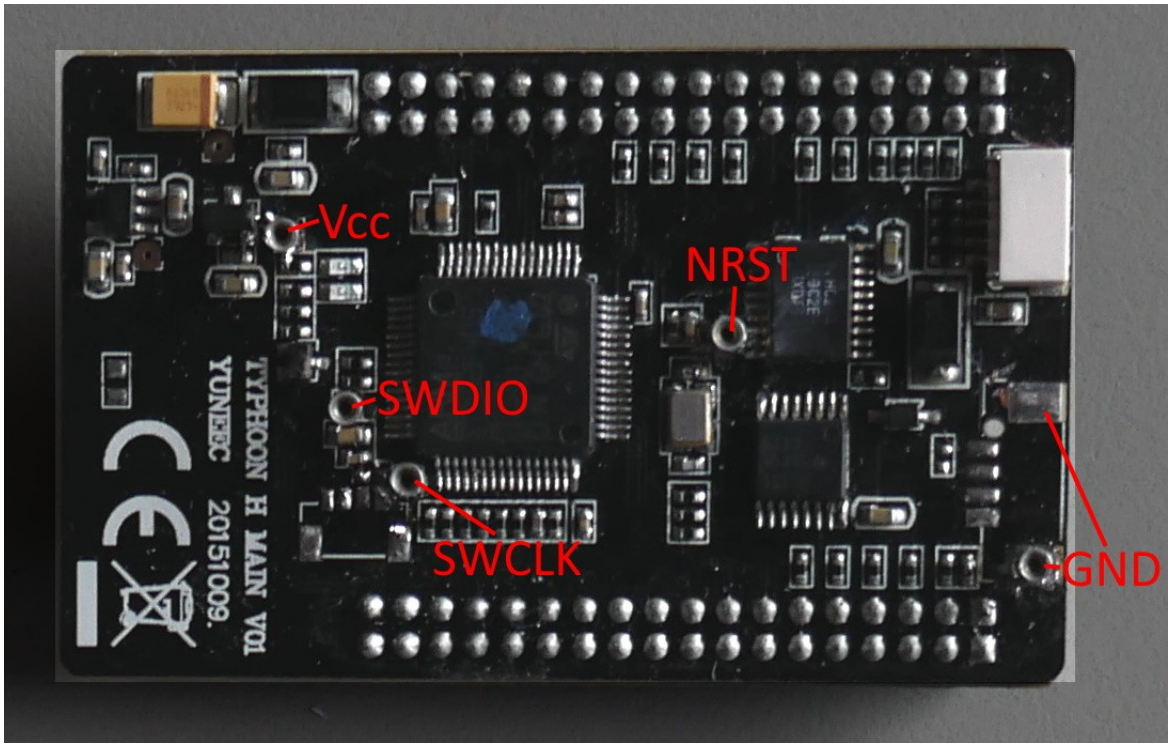
Also we need the related flash tool, the ST-Link utility. Download and install it.
https://www.st.com/en/development-tools/stsw-link004.html

Have two good micro USB data cables ready and the micro USB pigtail cable from inside the Typhoon H to connect MCU-board to computer via USB for power supply during flashing and for testing afterwards.

We need a good soldering station and soldering skills to prepare the contact pads which have to be connected to the programmer unit.

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.

Quick test:
Before we start at all, check if the MCU-Board is OKI and working. Connect it via USB to the PC and check if the LEDs on the MCU board are blinking. In Device Manager > Ports (COM & LPT) a device “STMicroelectronics Virtual COM Port (COM...)” should appear.



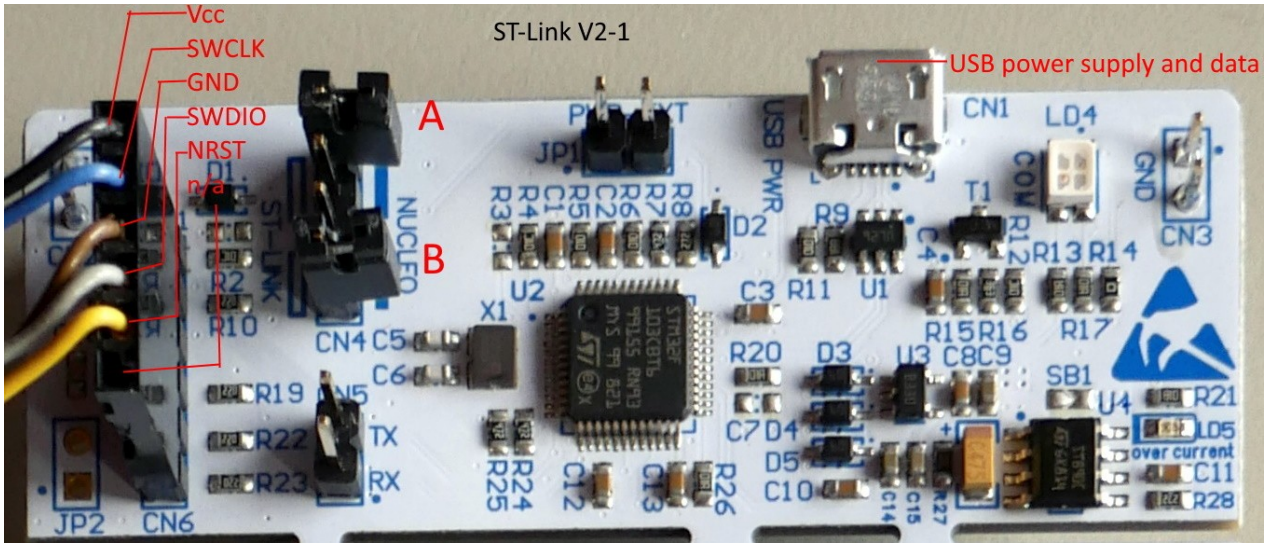
To set up parameters and update the drone we need PX4 configuration tool "QGroundControl". Download and install it. It needs Administrator rights at least for Windows10.
http://qgroundcontrol.com/downloads/

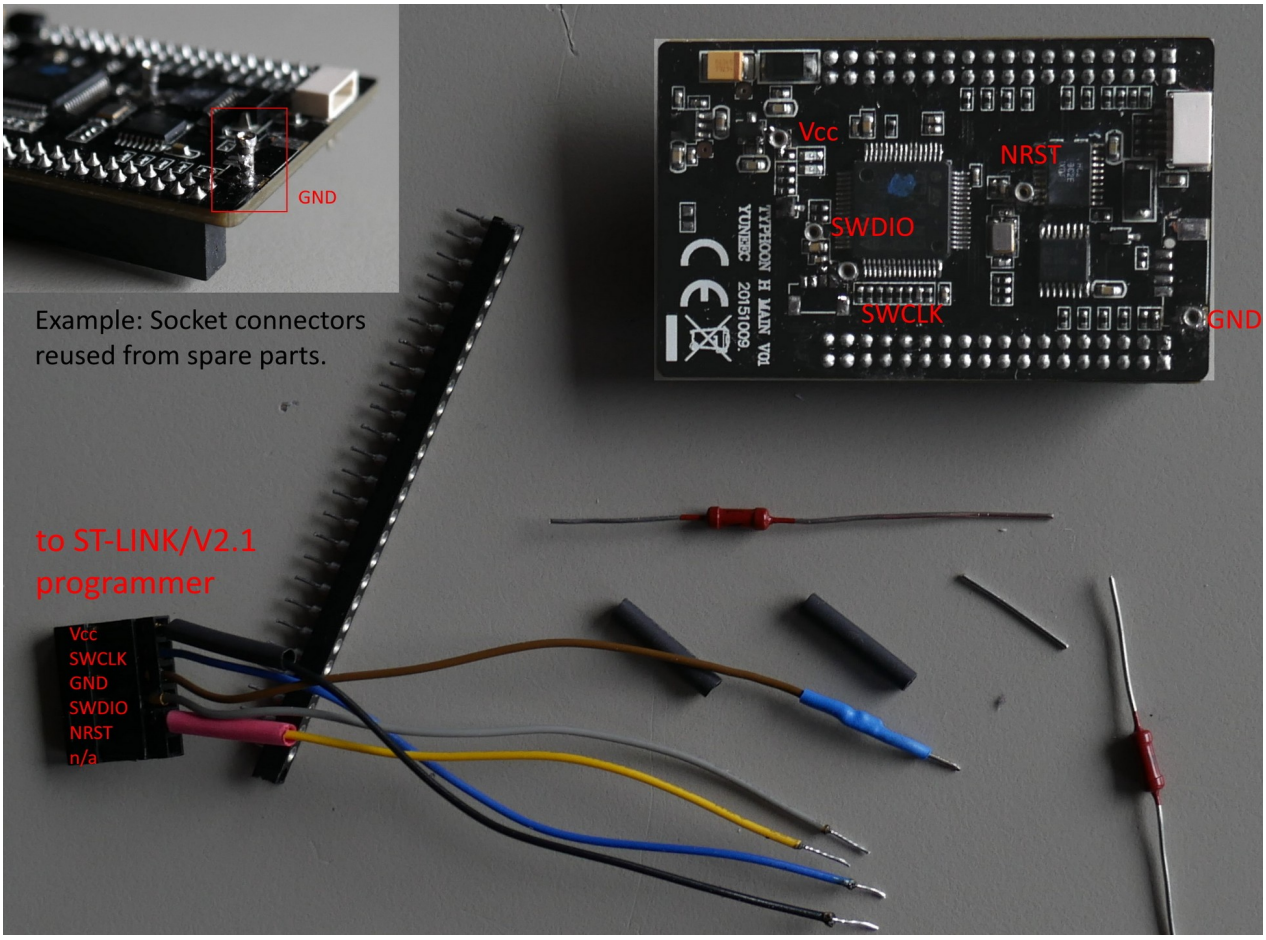
⚠**Notes:**

- ➔ **To avoid static damage follow the ESD rules. Ground everything you can.**
- ➔ **Before we go ahead, download and read the documentation for the programmer unit and the ST-Link utility.**

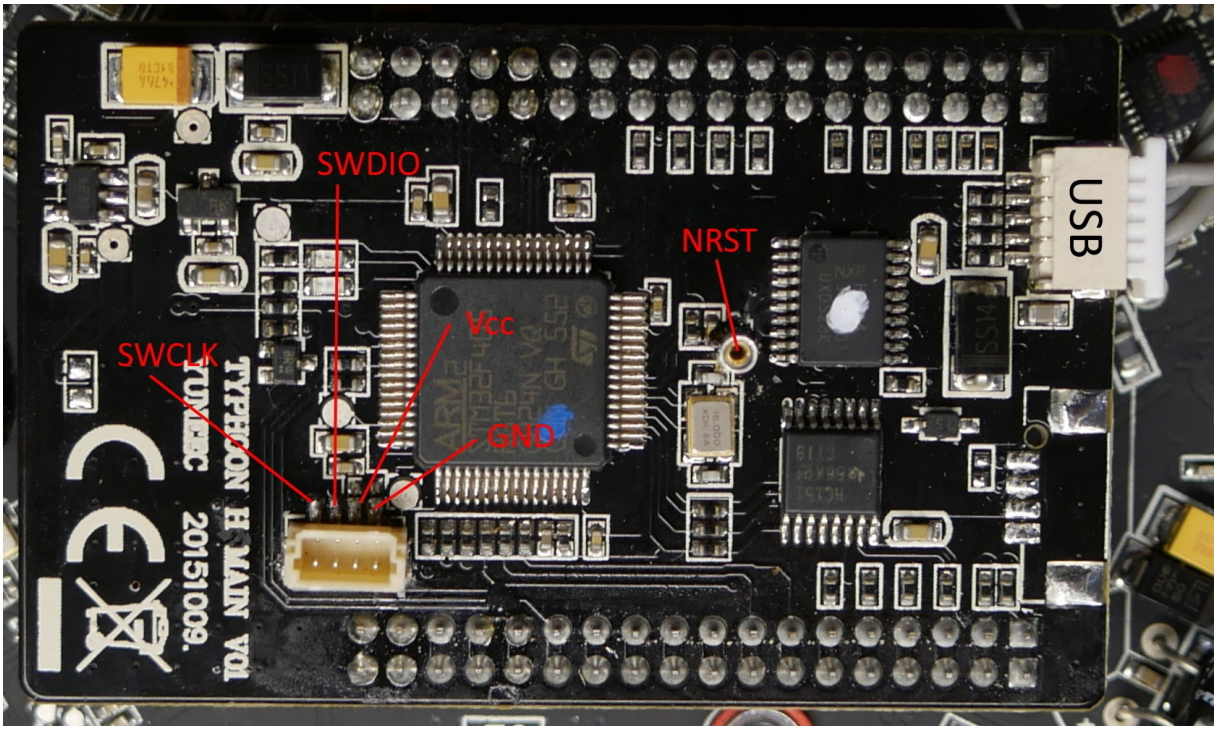
Step 1: Wiring for the programmer (the most dangerous step)

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).





We can also use the pads for a tiny connector if you have one available that fits the pads:



The NRST pin still has to be wired because it is not available at the connector pads.

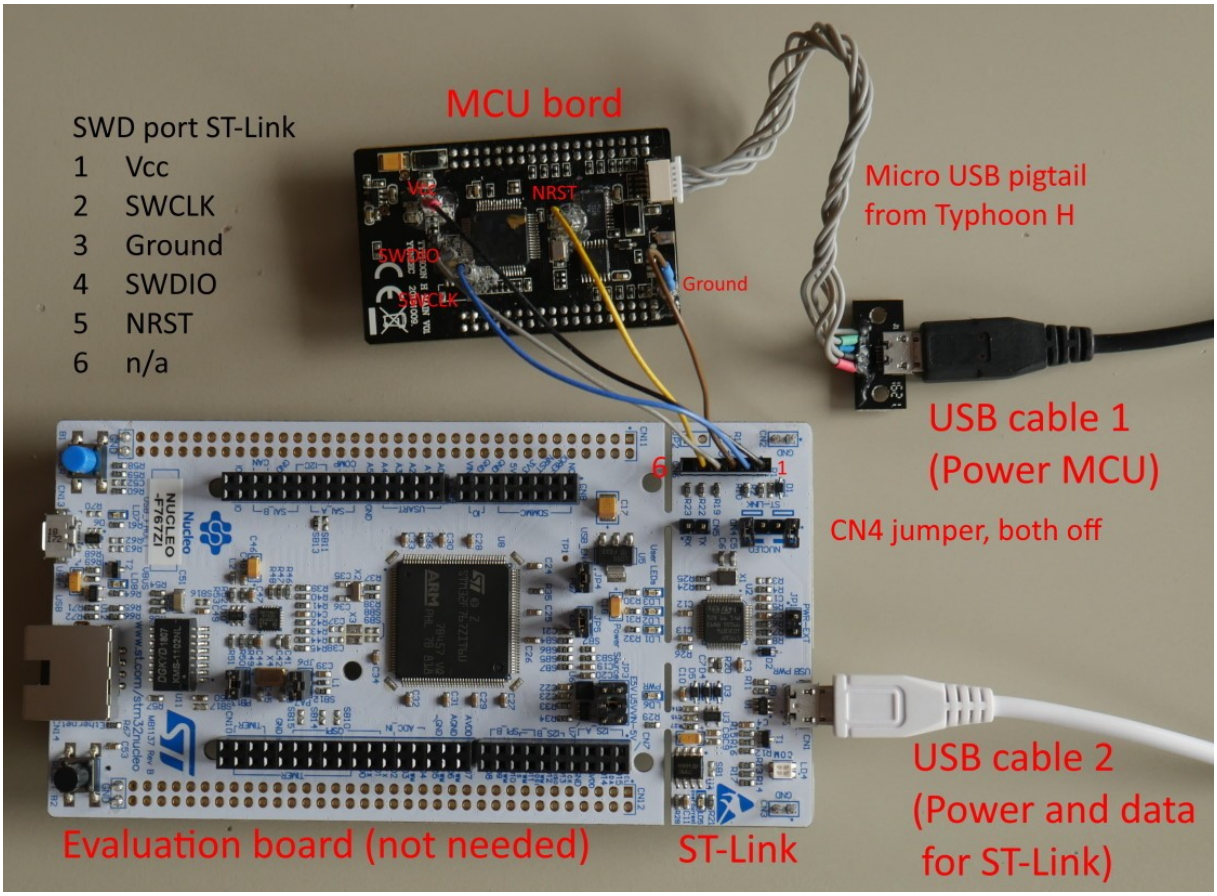
It's a good idea to secure the connectors with a tiny tip of super glue. Be careful when plug-in and more careful when plug-out the wiring.

Quick test:
After wiring was done, check if the MCU-Board is still booting. Connect it via USB to the PC and check if the LEDs on the MCU board are blinking. In Device Manager > Ports (COM & LPT) a device “STMicroelectronics Virtual COM Port (COM...)” should appear.

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. Connect the ST-Link with the second USB-cable to the PC.



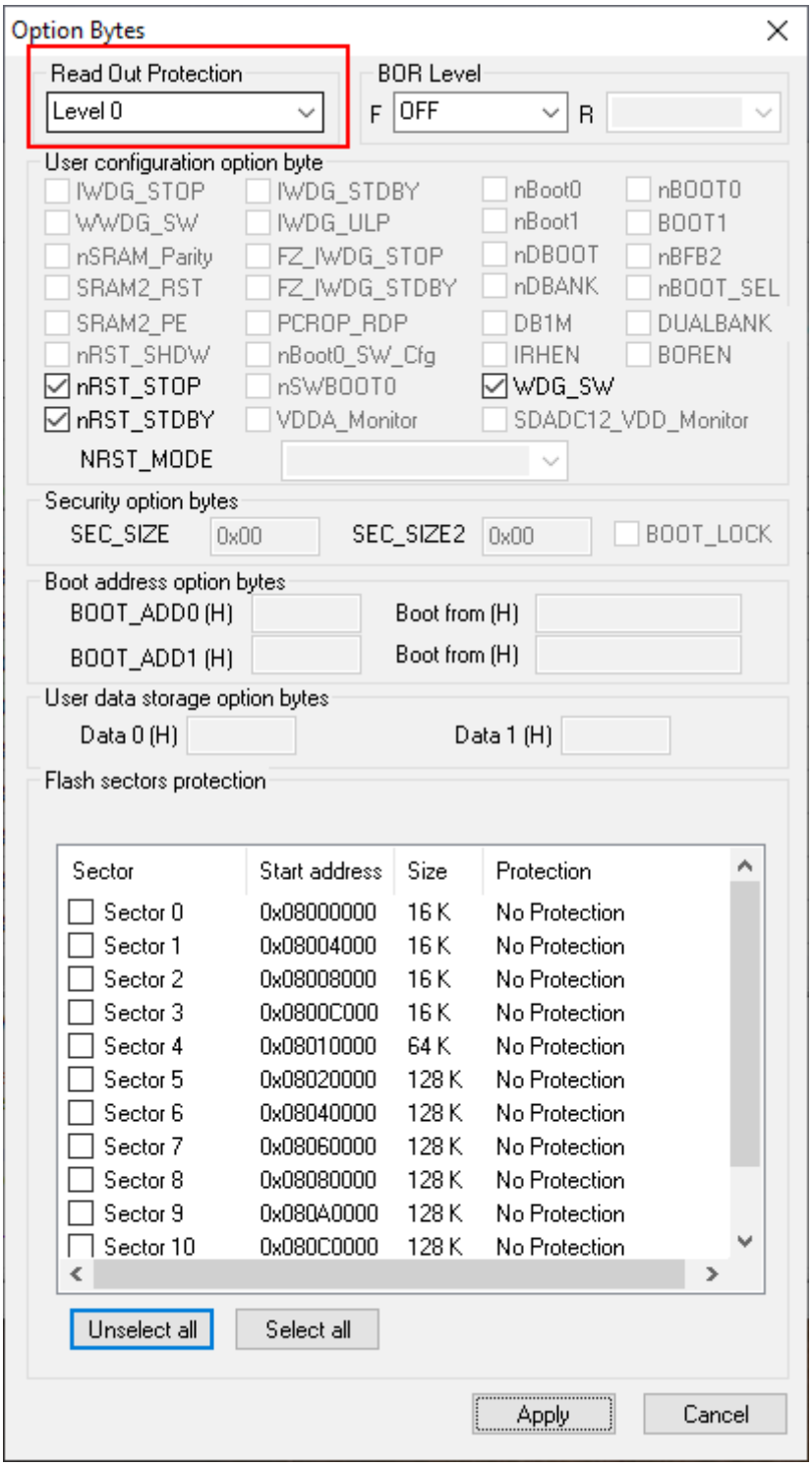
If all is connected as shown above, start the STM32 ST-Link utility.

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

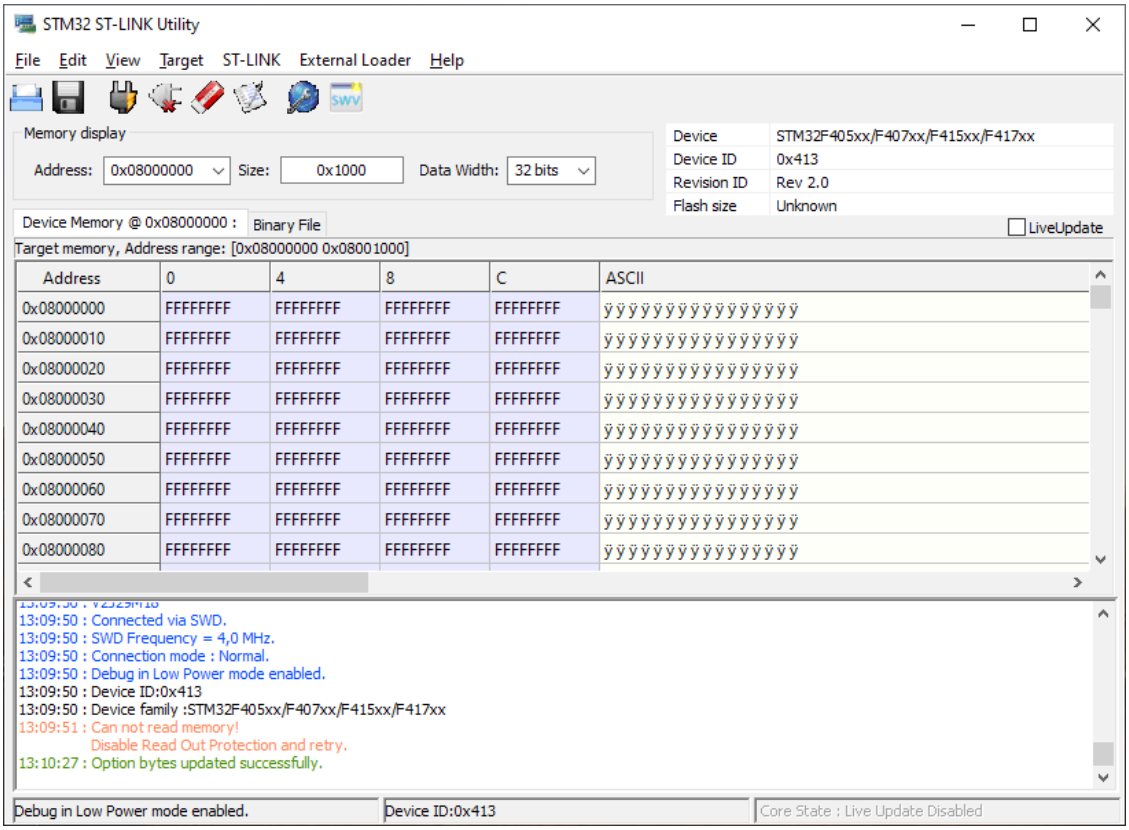
Set Read Out Protection to Level 0:

Target > Option Bytes…
Select "Level 0" > Apply.

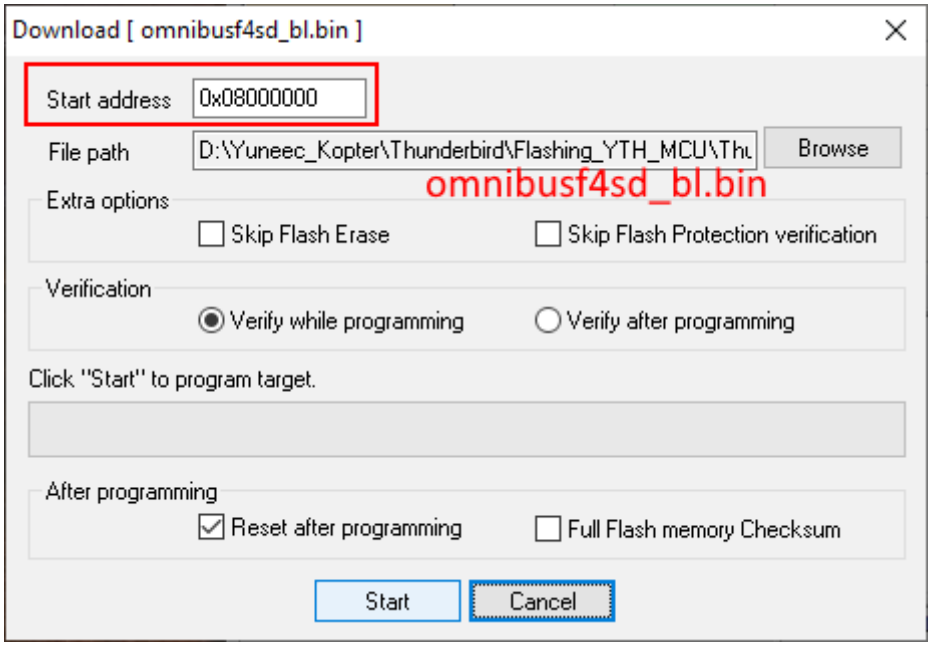
Now the old bootloader will be removed.



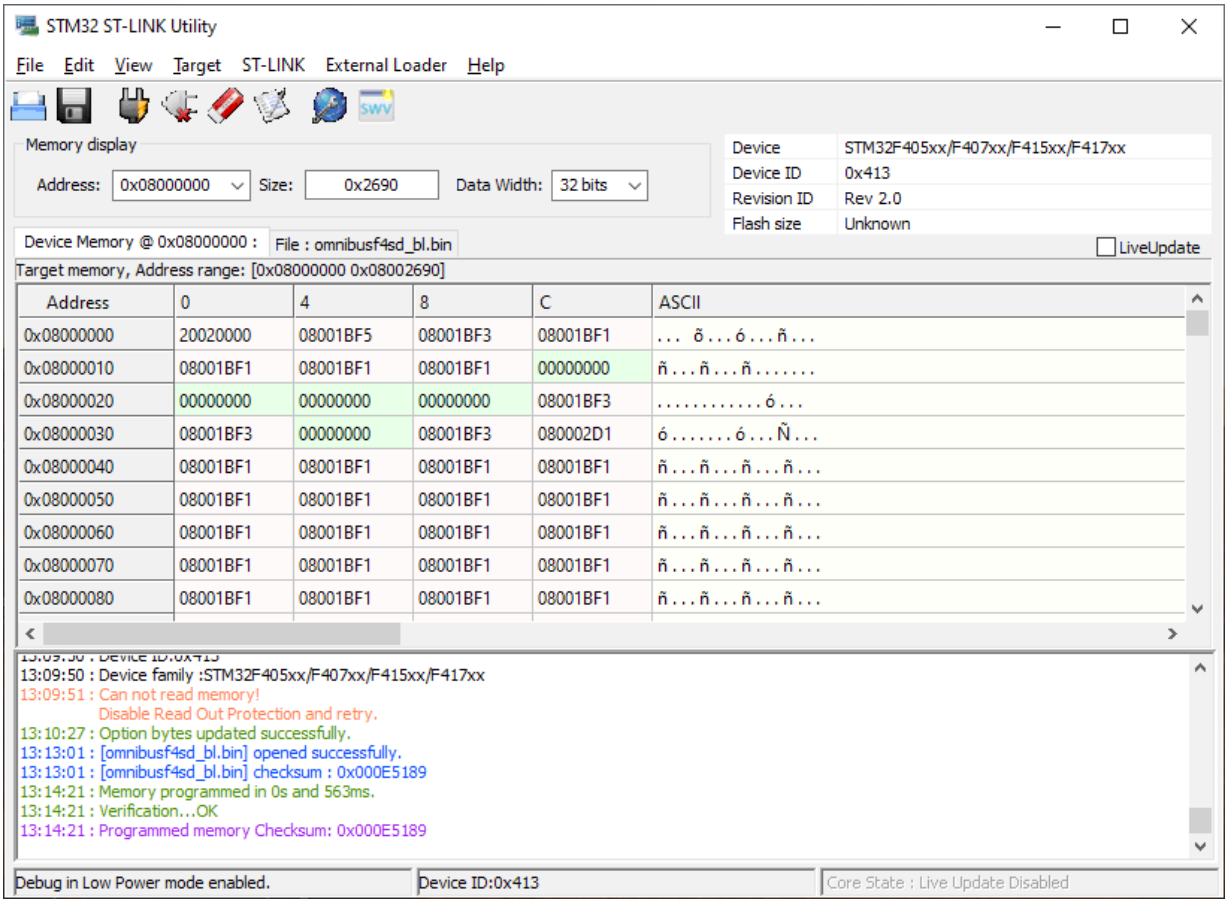
If success it looks like this:



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:



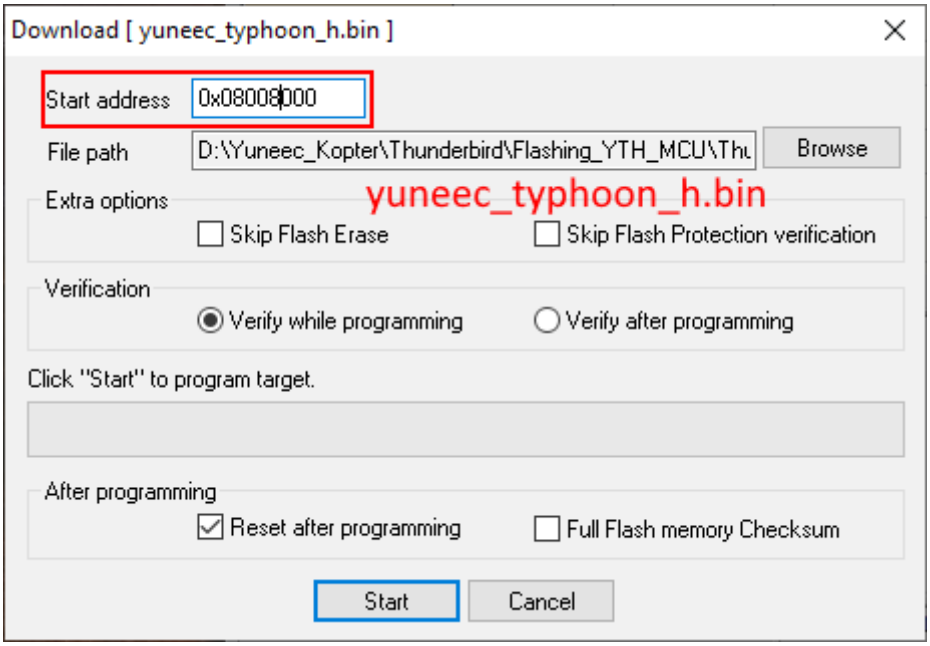
The MCU has now a PX4 bootloader, placed in the beginning of the Flash, 0x8000000. This is where the ROM bootloader starts up, whatever there is. There is no way to brick the STM32, the ROM bootloader will always be there, and the bootloader you just flashed, is a "second stage" bootloader, which is 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.

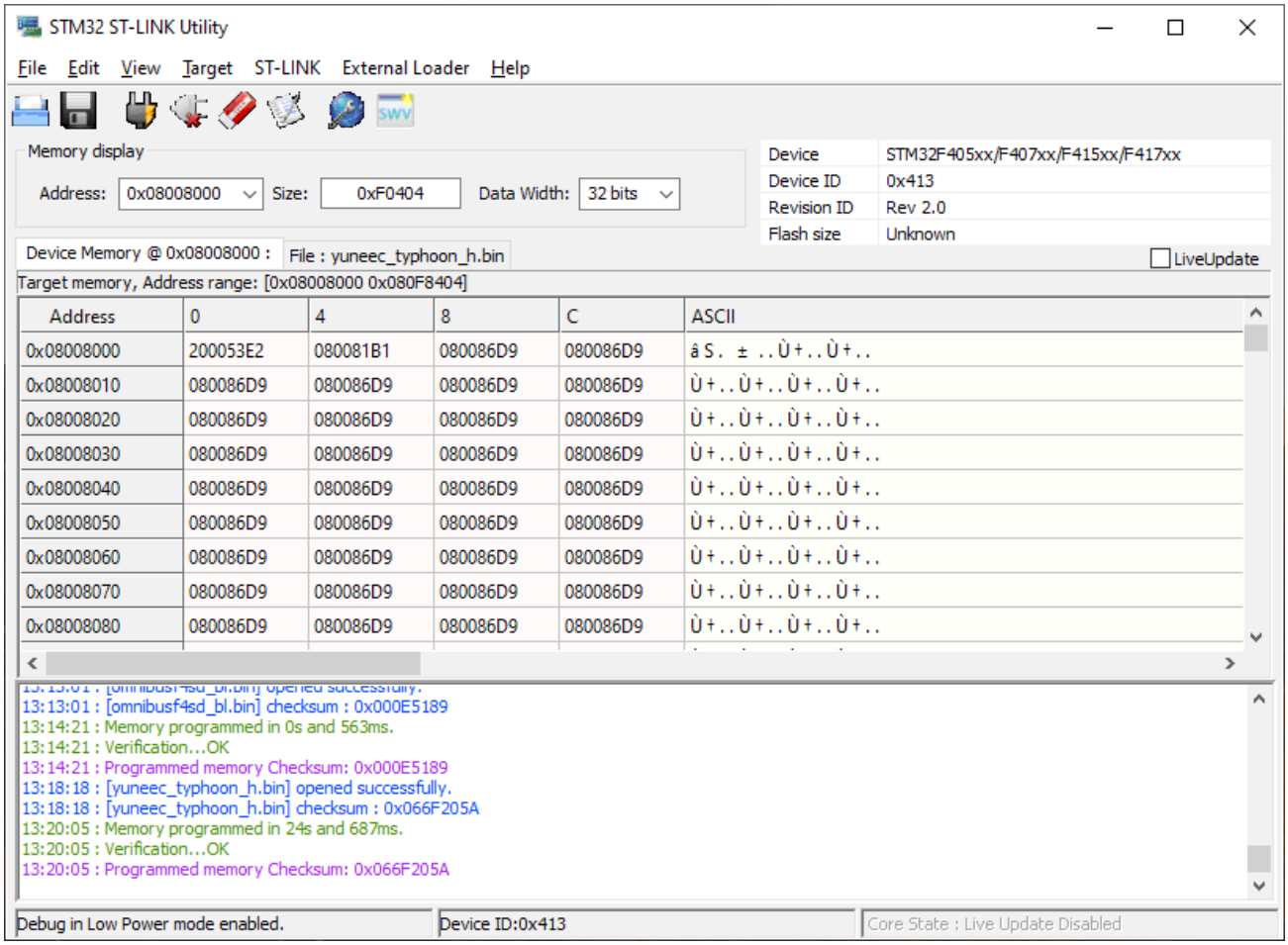
Step 3: Flash the main application

Because we are still connected to the programmer interface, we upload the latest firmware to the MCU-board. Later firmware updates will be done via USB. We don't need the hardware programmer interface anymore.

Go to Target > Program & Verify. Select "**yuneec_typhoon_h.bin**" (or whatever filename the latest firmware has) from downloaded binaries.
<u>After</u> selection of the file, 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.

Quick test MCU board:
Reconnect it via USB cable to PC. The LEDs on MCU board are no more operable but in Windows Device Manager > Ports (COM & LPT) a device “Legacy FMU (COM...)” should appear now.

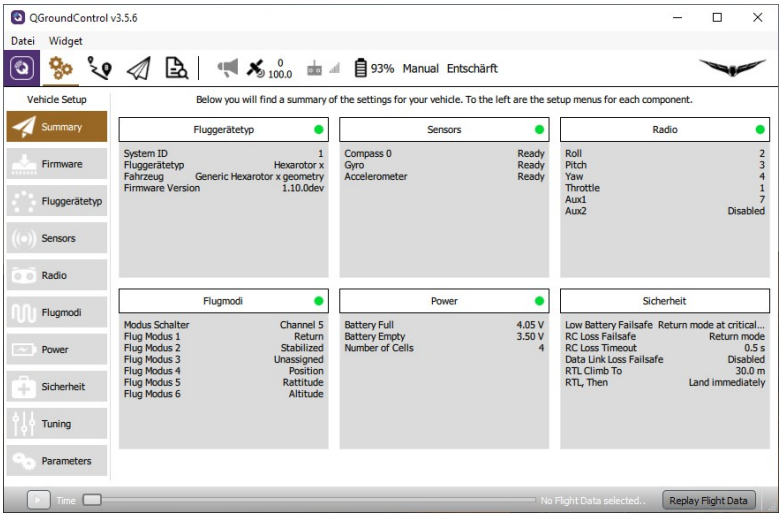
Reassemble the MCU board to the main board (ESC board). Check if it is correctly inserted.

Reboot the drone. The drone's power button has a delay of ~8 seconds. During this time interval 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

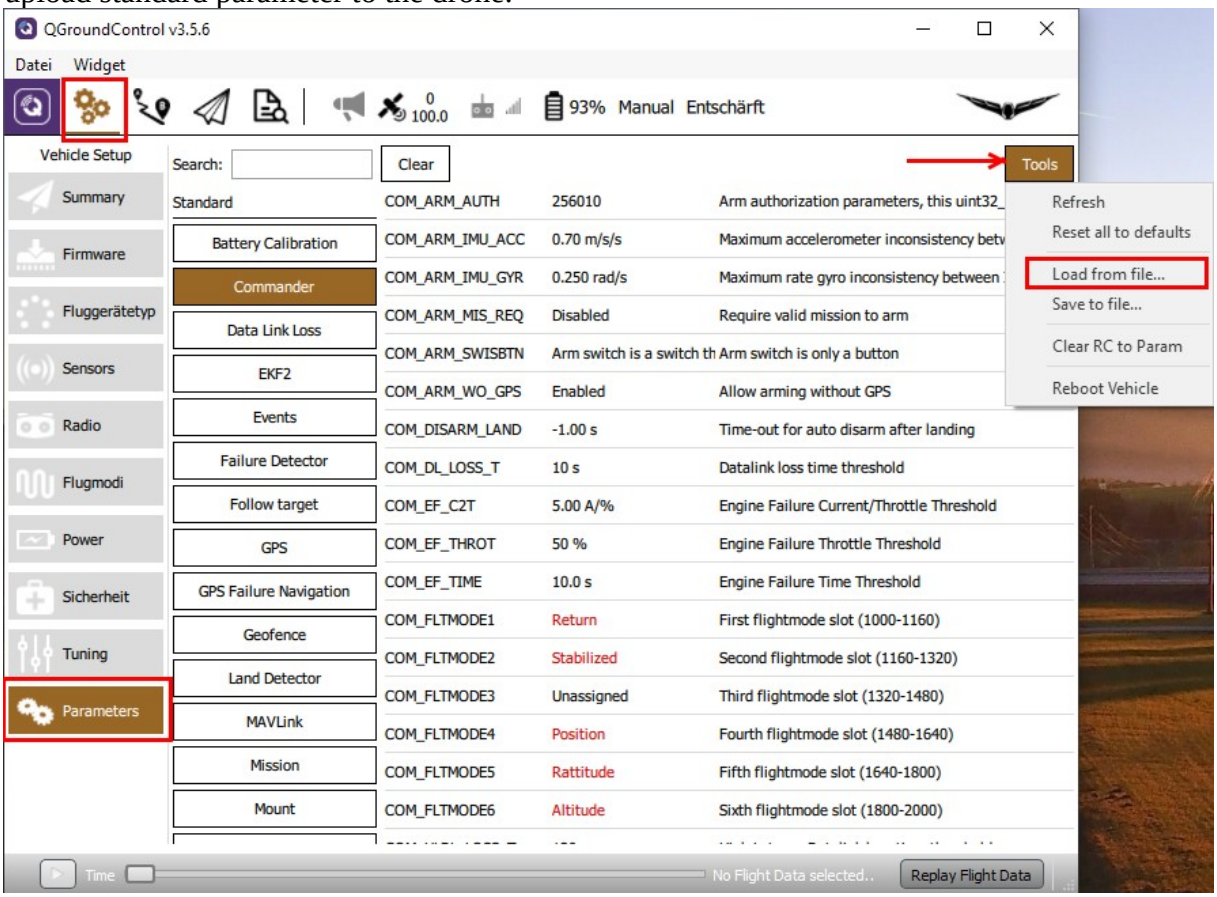
Install QGroundControl (aka QGC) if not yet done. Start QGroundControl as Administrator.

Power up the drone. Connect the USB cable and wait until drone was connected to QGC. This may take some time.
If connected, the telemetry will appear and a hint that the drone was not yet configured.



Load parameter file "**typhoon_h_default_parameters.params**" using QGC's parameter page.

Go to parameters > Tools > Load from file… > Select file "typhoon_h_parameters.params" to upload standard parameter to the drone.



Step 5: Set hardware and calibrate sensors

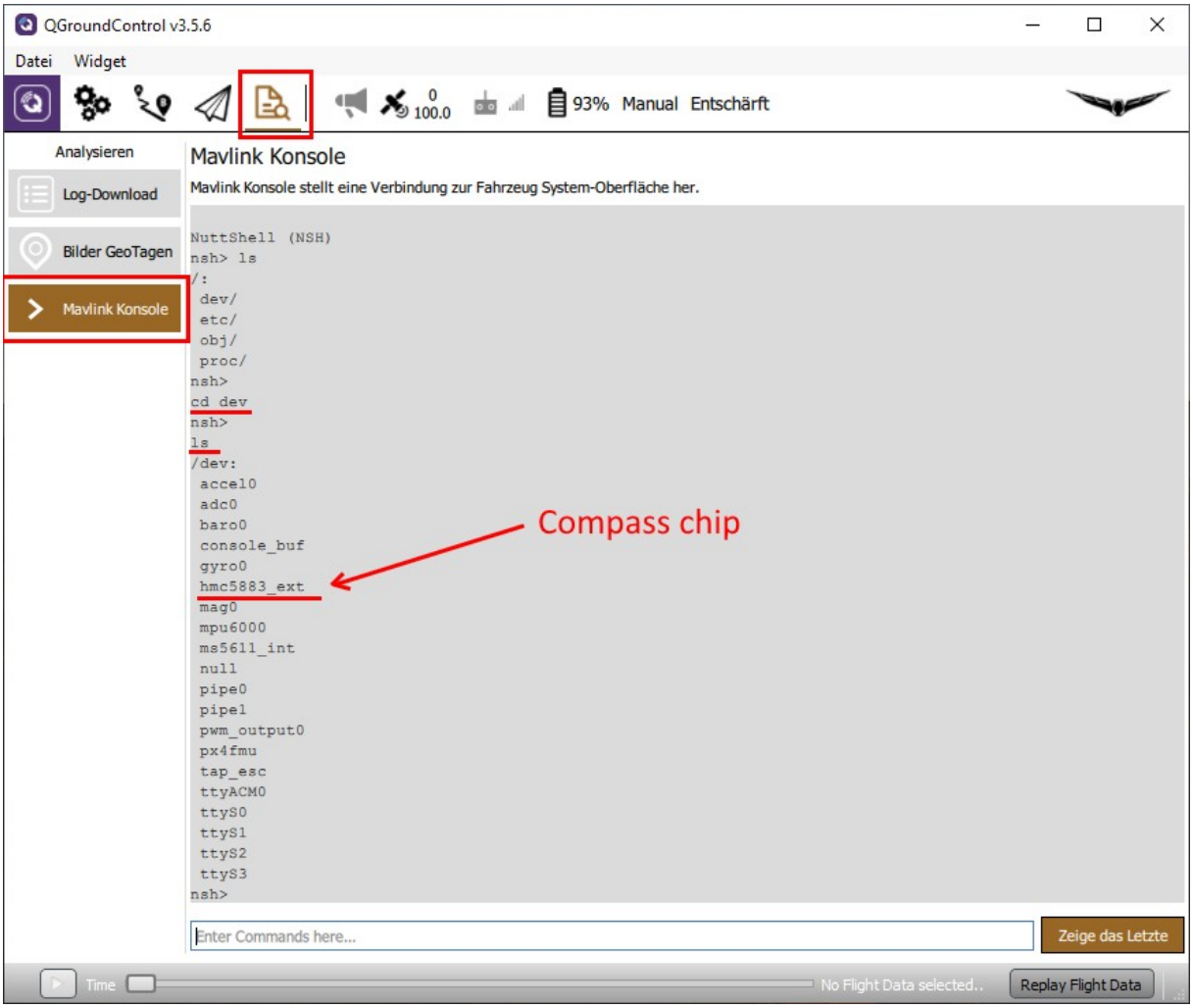
For Typhoon H exists two different compass chips. Older GPS boards have HMC5883, newer boards have IST8310 applied as compass chip. To find out which one you have, connect the drone to QGC and go to Mavlink console:

Type:

cd /dev
ls

and you will get a list of drivers for the hardware. Nice, what we can see here.

There could be "hmc5883_ext" or "ist8310_ext" as compass chip.

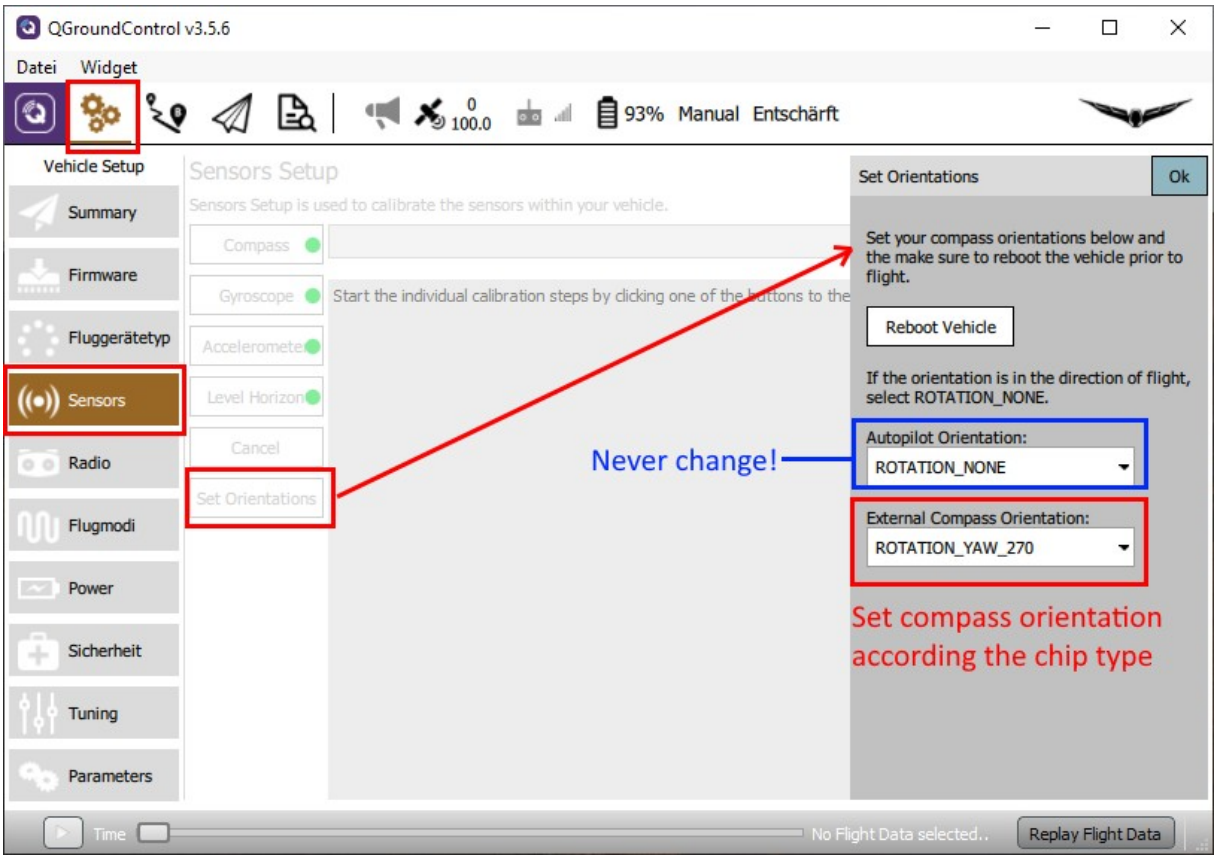


Important: If you have the "hmc5883_ext", you have to change External Compass Orientation to "ROTATION_YAW_270".

"ist8310_ext" it must be "ROTATION_YAW_180".

Never change Autopilot orientation. **It must be kept at "ROTATION_NONE"**.

Go to Settings > Sensors > Set Orientation



Then set External Compass Orientation to the correct value depending on you compass hardware. Save with "OK" and reboot the Thunderbird.

Do sensor calibrations like you would do for a fresh PX4 drone (see Basic Configuration · PX4 v1.9.0 User Guide). Don't use initial default settings. Flight battery must be well charged to allow proper calibration.
https://docs.px4.io/master/en/config/

Do not try to calibrate ESC's. It will not work and is not required.

Verify settings, check calibrations and sensors.

Troubleshooting

- Check if the drone is booting up correctly after soldering the connections at the measuring points on the MCU-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. 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). Be aware to change the start address to 0x8008000 **after** selection of the firmware file.

- Check if the MCU-board was correctly inserted into the plugs on the ESC-board. It could be accidentally shifted by one pin or something like that. Also pins tent to bent sometimes.

- In case of hardware changes or repairs do again Step 5: HW settings and calibration again.