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

Thunderbird....1

How to flash the MCU board of the Typhoon H....2

Preparation....2

Step 1: Wiring for the programmer (the most dangerous step)....4

Step 2: Replace the original bootloader by the new one (no way back!)....6

Step 3: Flash the main application....10

Step 4: Set drone parameters....12

Step 5: Set hardware and calibrate sensors....13

Troubleshooting....15

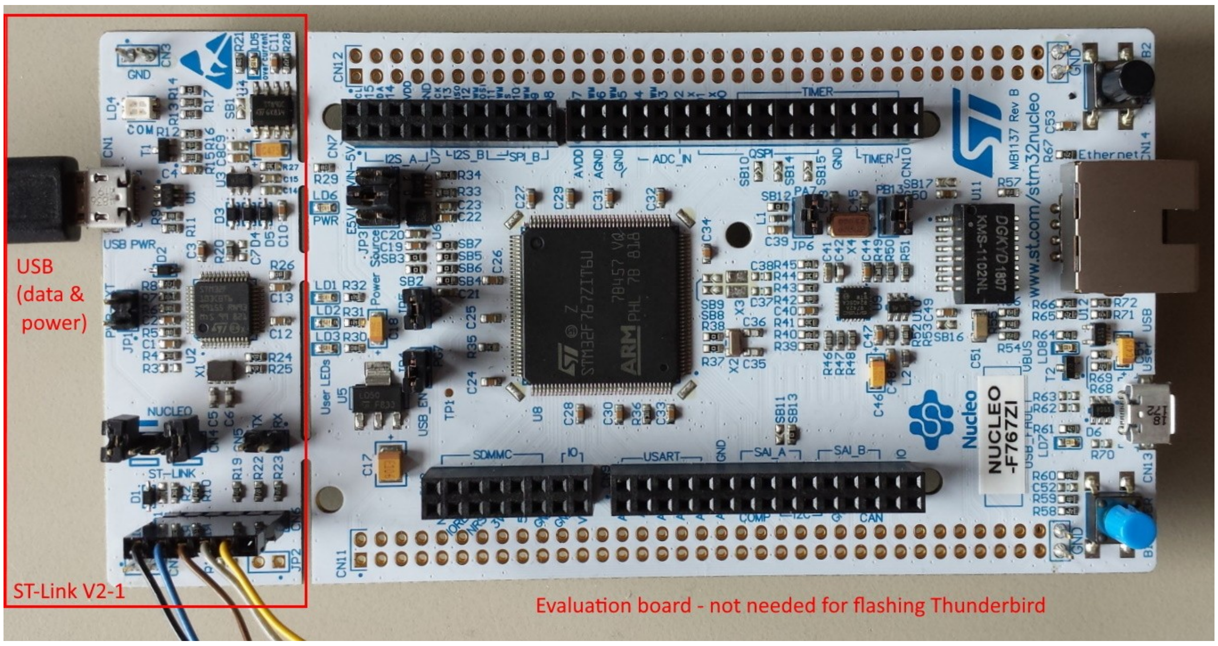
How to flash the MCU board of the Typhoon H

⚠**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.

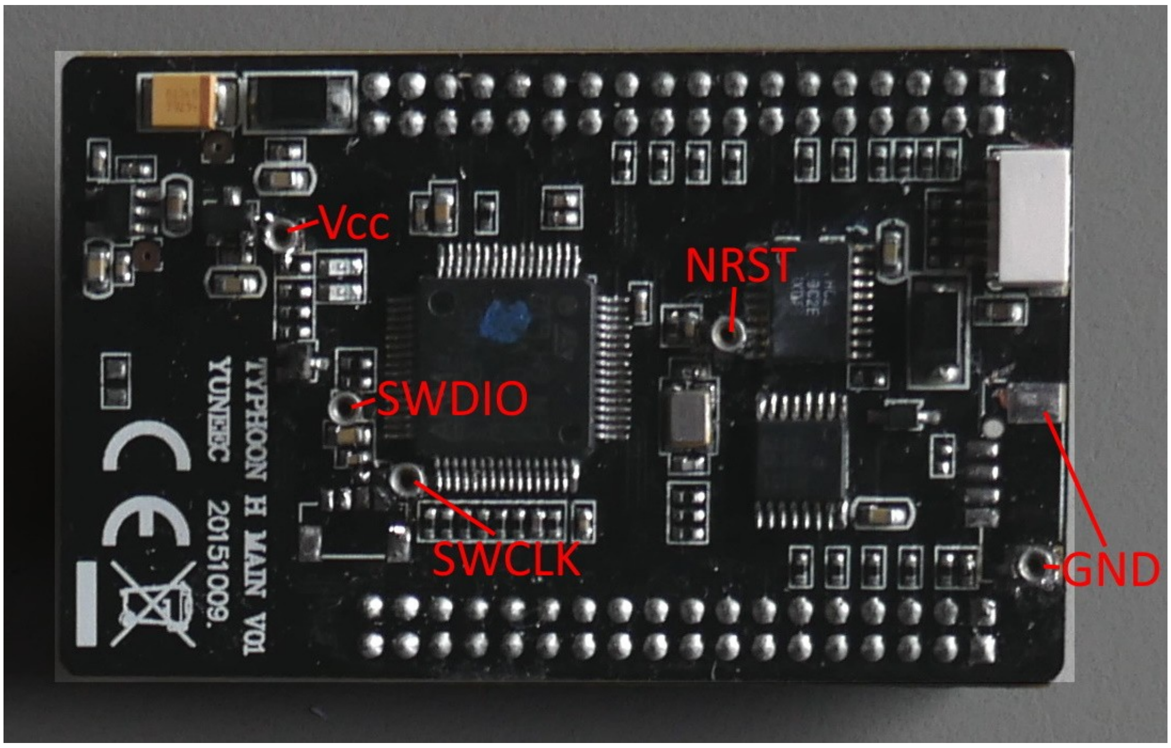
Also we need the related flash tool, the ST-Link utility. Download and install it.
https://www.st.com/en/development-tools/stsw-link004.html

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.



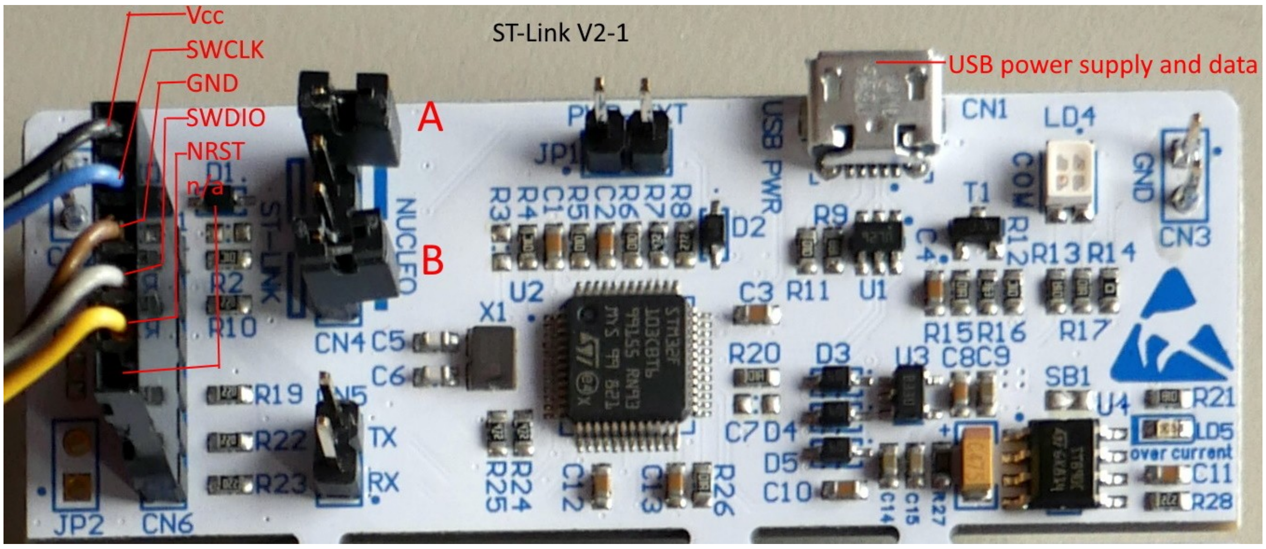
To set up parameters and update the drone we need PX4 configuration tool "QGroundControl". Download and install it. It needs Administrator rights at least for Windows10.
http://qgroundcontrol.com/downloads/

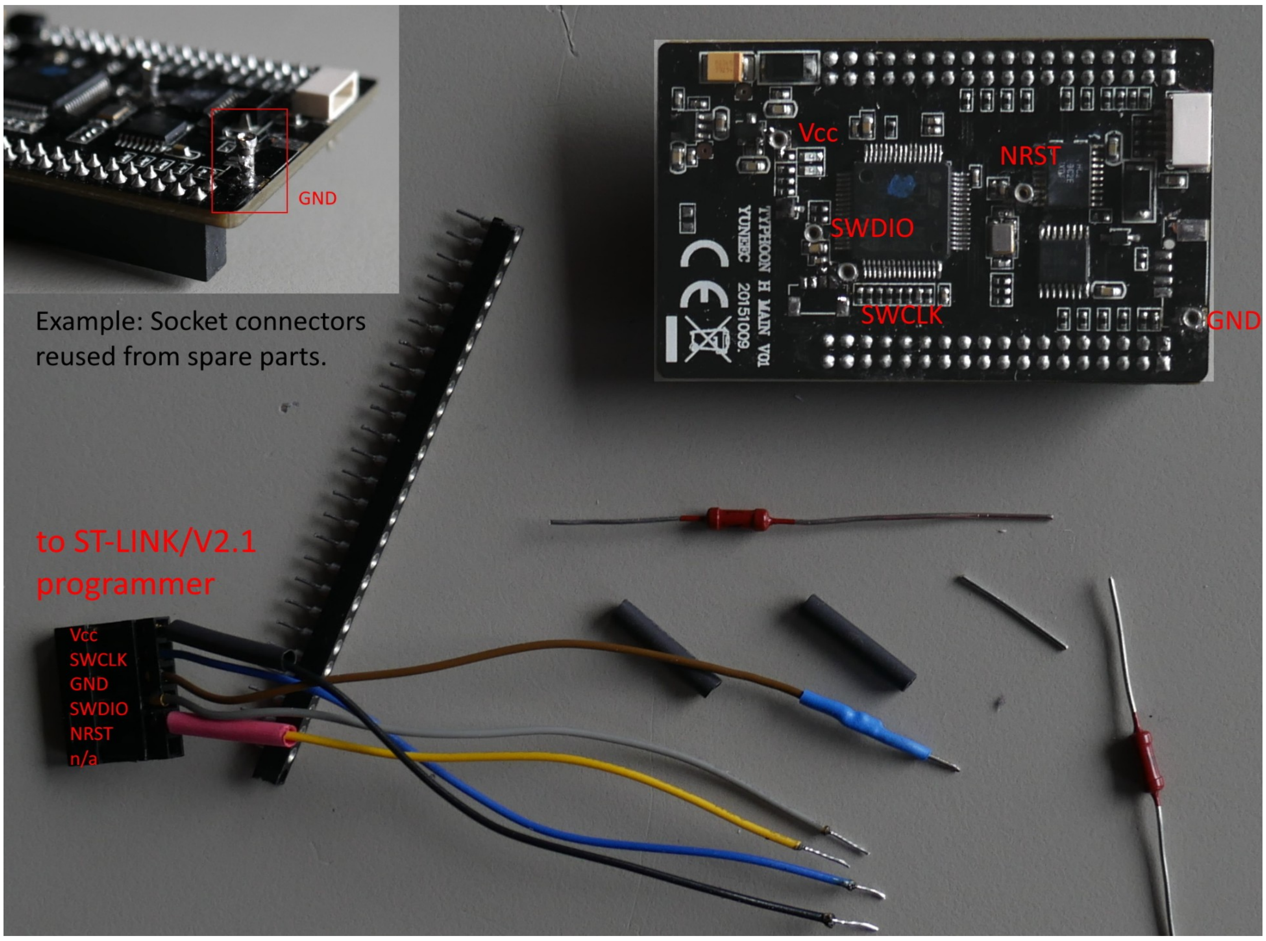
⚠**Notes:**

- ➔ **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.**

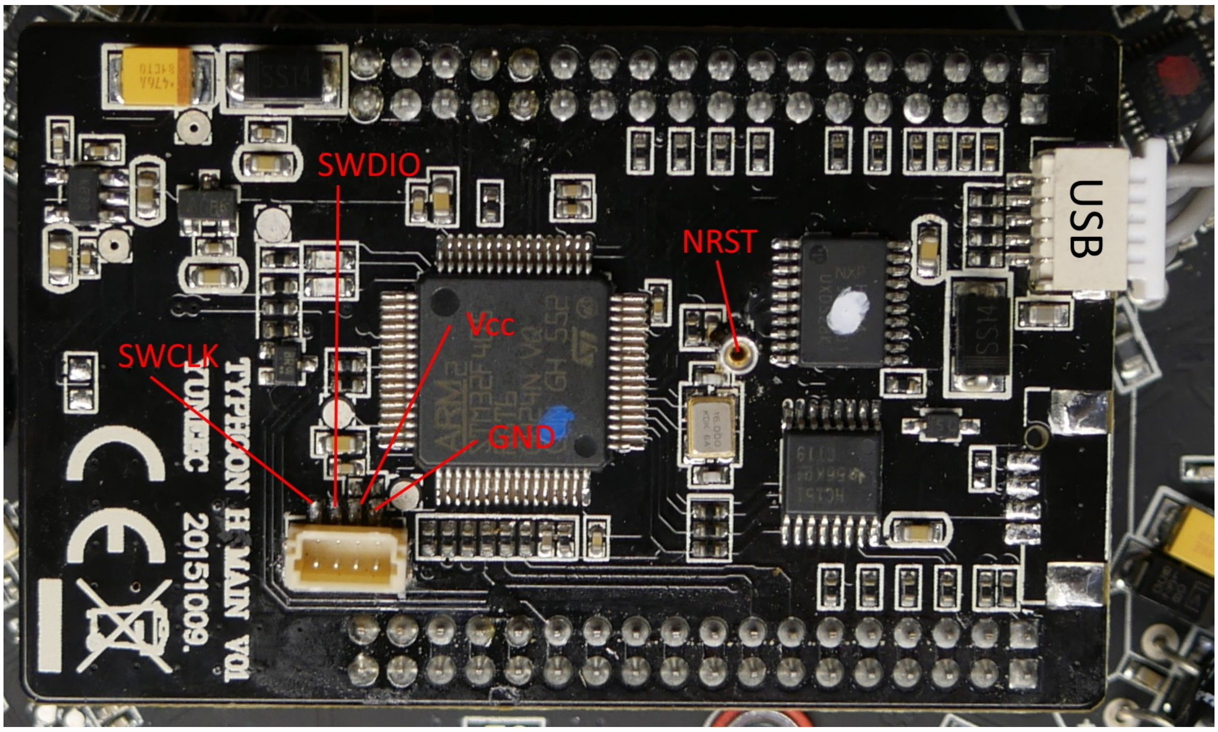
Step 1: Wiring for the programmer (the most dangerous step)

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).




GND
Example: Socket connectors reused from spare parts.
Vcc
NRST
SWDIO
SWCLK
GND
to ST-LINK/V2.1 programmer
Vcc
SWCLK
GND
SWDIO
NRST
n/a

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.

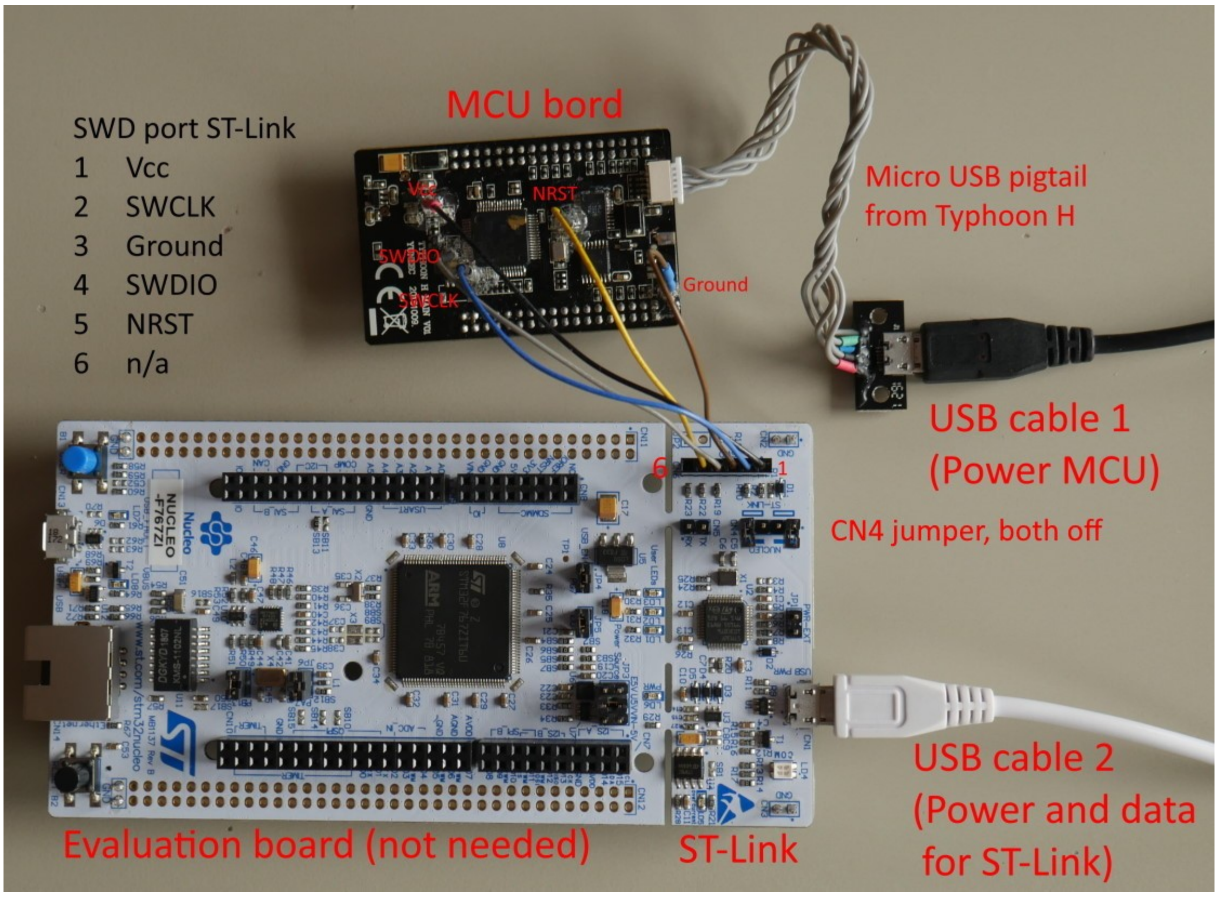
It's a good idea to secure the connectors with a tiny tip of super glue. Be careful when plug-in and more careful when plug-out the wiring.

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.



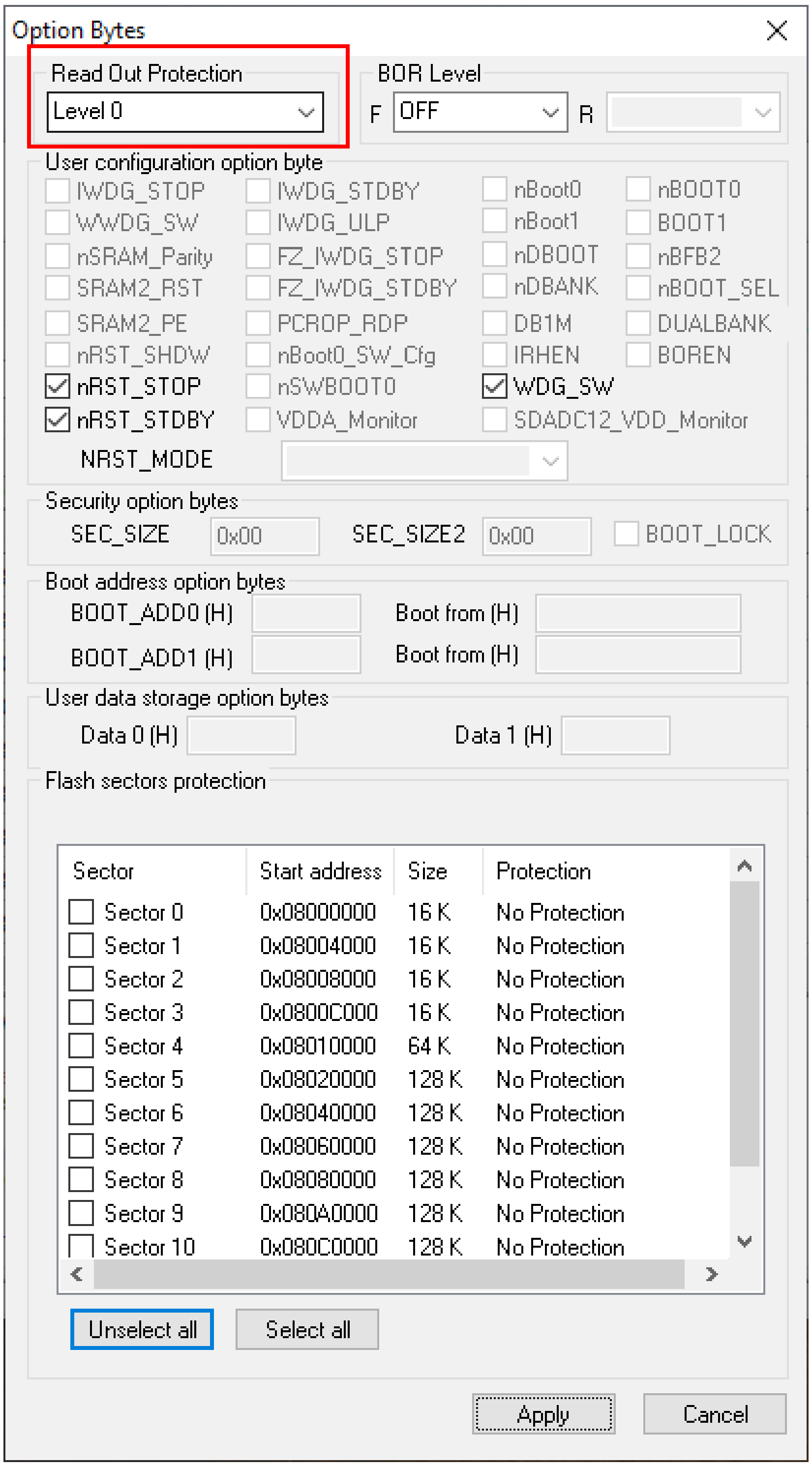
If all is connected as shown above, start the STM32 ST-Link utility.

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.



Option Bytes

Read Out Protection: Level 0

BOR Level: F OFF R

User configuration option byte

☐ IWDG_STOP	☐ IWDG_STDBY	☐ nBoot0	☐ nBOOT0
☐ WWDG_SW	☐ IWDG_ULP	☐ nBoot1	☐ BOOT1
☐ nSRAM_Parity	☐ FZ_IWDG_STOP	☐ nDBOOT	☐ nBFB2
☐ SRAM2_RST	☐ FZ_IWDG_STDBY	☐ nDBANK	☐ nBOOT_SEL
☐ SRAM2_PE	☐ PCROP_RDP	☐ DB1M	☐ DUALBANK
☐ nRST_SHDW	☐ nBoot0_SW_Cfg	☐ IRHEN	☐ BOREN
✓ nRST_STOP	☐ nSWBOOT0	✓ WDG_SW	
✓ nRST_STDBY	☐ VDDA_Monitor	☐ SDADC12_VDD_Monitor	

NRST_MODE

Security option bytes

SEC_SIZE 0x00 SEC_SIZE2 0x00 ☐ BOOT_LOCK

Boot address option bytes

BOOT_ADD0 (H) Boot from (H)

BOOT_ADD1 (H) Boot from (H)

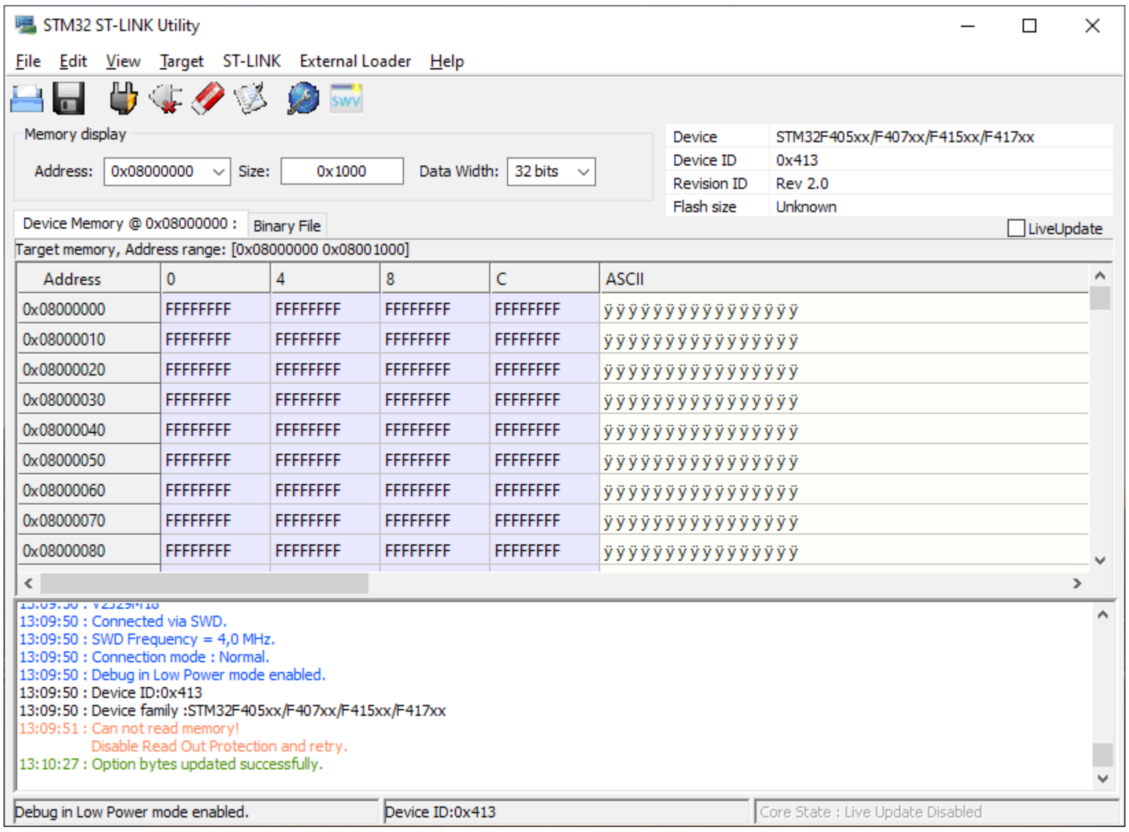
User data storage option bytes

Data 0 (H) Data 1 (H)

Flash sectors protection

Sector	Start address	Size	Protection
☐ Sector 0	0x08000000	16 K	No Protection
☐ Sector 1	0x08004000	16 K	No Protection
☐ Sector 2	0x08008000	16 K	No Protection
☐ Sector 3	0x0800C000	16 K	No Protection
☐ Sector 4	0x08010000	64 K	No Protection
☐ Sector 5	0x08020000	128 K	No Protection
☐ Sector 6	0x08040000	128 K	No Protection
☐ Sector 7	0x08060000	128 K	No Protection
☐ Sector 8	0x08080000	128 K	No Protection
☐ Sector 9	0x080A0000	128 K	No Protection
☐ Sector 10	0x080C0000	128 K	No Protection

If success it looks like this:


STM32 ST-LINK Utility
File Edit View Target ST-LINK External Loader Help
Memory display
Device Memory @ 0x08000000 :
Binary File
LiveUpdate

Address: 0x08000000 Size: 0x1000 Data Width: 32 bits

Device	STM32F405xx/F407xx/F415xx/F417xx
Device ID	0x413
Revision ID	Rev 2.0
Flash size	Unknown

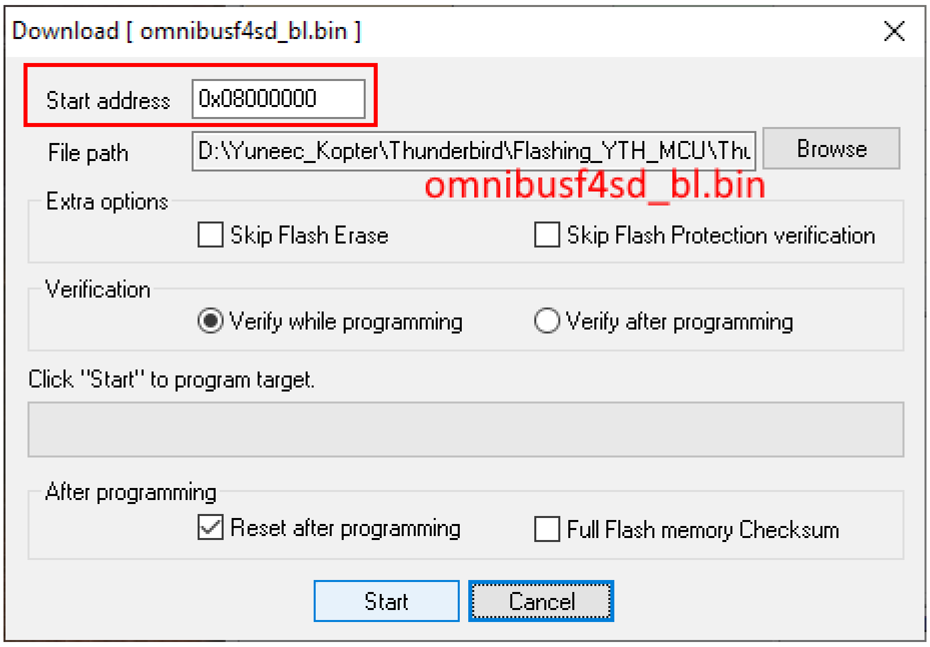
Target memory, Address range: [0x08000000 0x08001000]

Address	0	4	8	C	ASCII
0x08000000	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000010	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000020	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000030	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000040	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000050	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000060	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000070	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000080	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ

```
13:09:50 : Connected via SWD.
13:09:50 : SWD Frequency = 4,0 MHz.
13:09:50 : Connection mode : Normal.
13:09:50 : Debug in Low Power mode enabled.
13:09:50 : Device ID:0x413
13:09:50 : Device family :STM32F405xx/F407xx/F415xx/F417xx
13:09:51 : Can not read memory!
           Disable Read Out Protection and retry.
13:10:27 : Option bytes updated successfully.
```

Debug in Low Power mode enabled. | Device ID:0x413 | Core State : Live Update Disabled

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



Start address: 0x08000000

File path: D:\Yuneec_Kopter\Thunderbird\Flashing_YTH_MCU\Thu

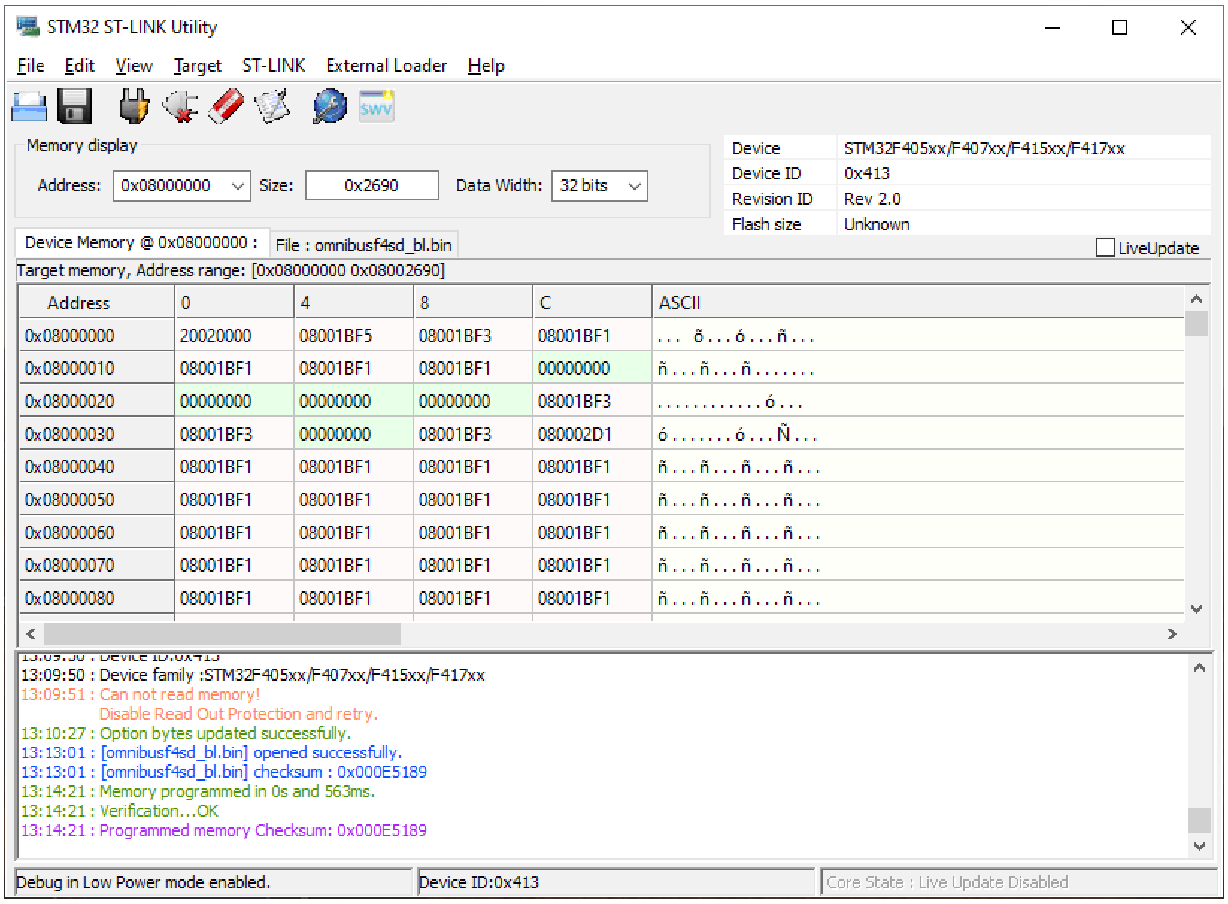
Extra options: ☐ Skip Flash Erase ☐ Skip Flash Protection verification

Verification: [✓ Verify while programming] Verify after programming

Click "Start" to program target.

After programming: ☑ Reset after programming ☐ Full Flash memory Checksum

If success it looks like this:


STM32 ST-LINK Utility

File Edit View Target ST-LINK External Loader Help

Memory display

Address: 0x08000000 Size: 0x2690 Data Width: 32 bits

Device	STM32F405xx/F407xx/F415xx/F417xx
Device ID	0x413
Revision ID	Rev 2.0
Flash size	Unknown

Device Memory @ 0x08000000 : File : omnibusf4sd_bl.bin

LiveUpdate

Target memory, Address range: [0x08000000 0x08002690]

Address	0	4	8	C	ASCII
0x08000000	20020000	08001BF5	08001BF3	08001BF1	... õ...ó...ñ...
0x08000010	08001BF1	08001BF1	08001BF1	00000000	ñ...ñ...ñ.......
0x08000020	00000000	00000000	00000000	08001BF3	............ó...
0x08000030	08001BF3	00000000	08001BF3	080002D1	ó.......ó...Ñ...
0x08000040	08001BF1	08001BF1	08001BF1	08001BF1	ñ...ñ...ñ...ñ...
0x08000050	08001BF1	08001BF1	08001BF1	08001BF1	ñ...ñ...ñ...ñ...
0x08000060	08001BF1	08001BF1	08001BF1	08001BF1	ñ...ñ...ñ...ñ...
0x08000070	08001BF1	08001BF1	08001BF1	08001BF1	ñ...ñ...ñ...ñ...
0x08000080	08001BF1	08001BF1	08001BF1	08001BF1	ñ...ñ...ñ...ñ...

```
13:09:50 : Device ID:0x413
13:09:50 : Device family :STM32F405xx/F407xx/F415xx/F417xx
13:09:51 : Can not read memory!
           Disable Read Out Protection and retry.
13:10:27 : Option bytes updated successfully.
13:13:01 : [omnibusf4sd_bl.bin] opened successfully.
13:13:01 : [omnibusf4sd_bl.bin] checksum : 0x000E5189
13:14:21 : Memory programmed in 0s and 563ms.
13:14:21 : Verification...OK
13:14:21 : Programmed memory Checksum: 0x000E5189
```

Debug in Low Power mode enabled. | Device ID:0x413 | Core State : Live Update Disabled

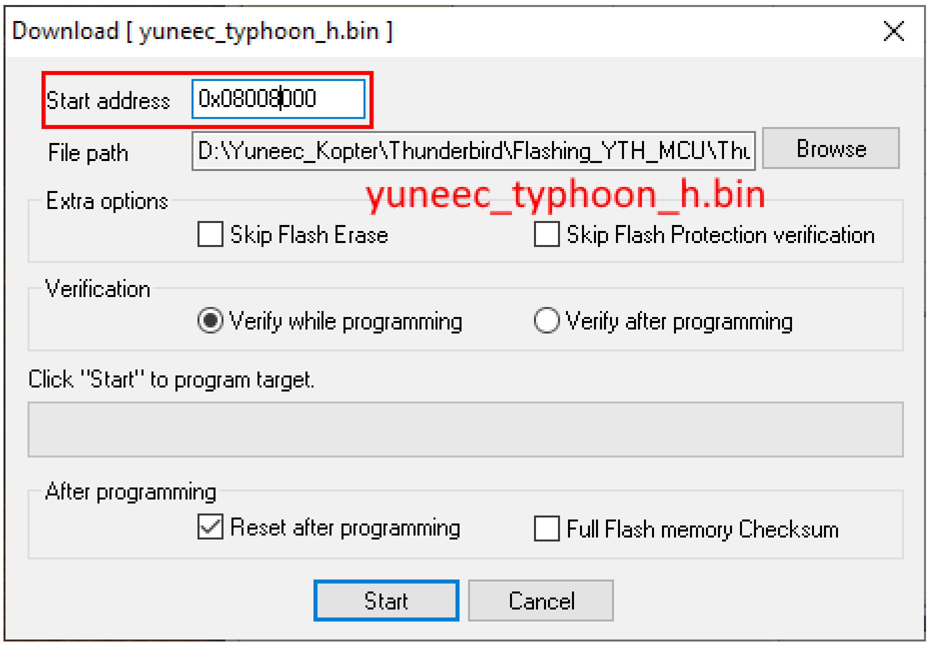
The MCU has now a PX4 bootloader, placed in the beginning of the Flash, 0x8000000. This is where the ROM bootloader starts up, whatever there is. There is no way to brick the STM32, the ROM bootloader will always be there, and the bootloader you just flashed, is a "second stage" bootloader, which is 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.
After selection of the file, set Start address to 0x8008000 and start flashing.



Download [yuneec_typhoon_h.bin]

Start address: 0x08008000

File path: D:\Yuneec_Kopter\Thunderbird\Flashing_YTH_MCU\Thu

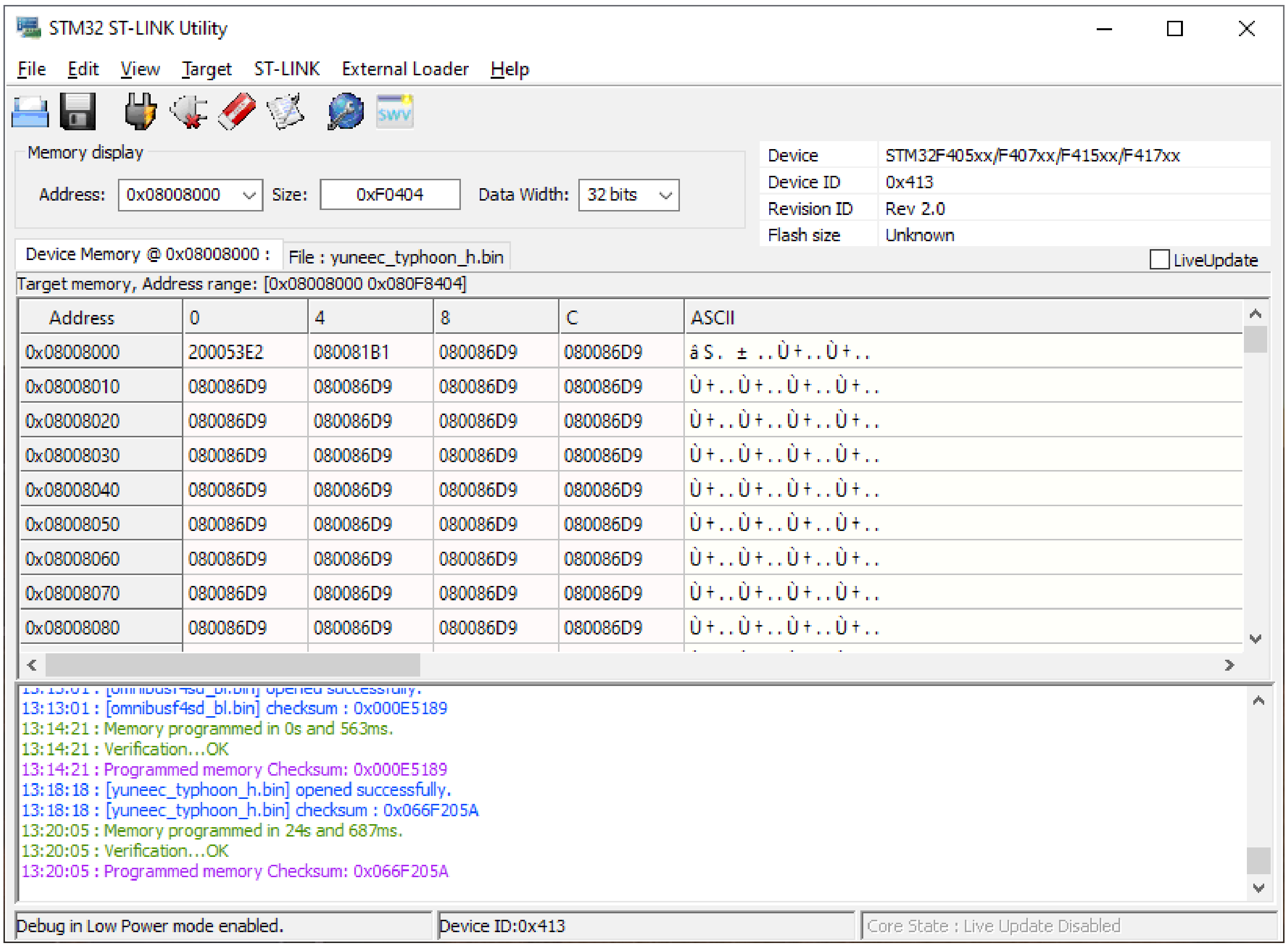
Extra options: ☐ Skip Flash Erase ☐ Skip Flash Protection verification

Verification: ● Verify while programming ○ Verify after programming

Click "Start" to program target.

After programming: ✓ Reset after programming ☐ Full Flash memory Checksum

If success it looks like this:



STM32 ST-LINK Utility

Memory display

Address: 0x08008000 Size: 0xF0404 Data Width: 32 bits

Device	STM32F405xx/F407xx/F415xx/F417xx
Device ID	0x413
Revision ID	Rev 2.0
Flash size	Unknown

Device Memory @ 0x08008000 : File : yuneec_typhoon_h.bin

Target memory, Address range: [0x08008000 0x080F8404]

Address	0	4	8	C	ASCII
0x08008000	200053E2	080081B1	080086D9	080086D9	â S . ± . . Ù + . . Ù + . .
0x08008010	080086D9	080086D9	080086D9	080086D9	Ù + . . Ù + . . Ù + . . Ù + . .
0x08008020	080086D9	080086D9	080086D9	080086D9	Ù + . . Ù + . . Ù + . . Ù + . .
0x08008030	080086D9	080086D9	080086D9	080086D9	Ù + . . Ù + . . Ù + . . Ù + . .
0x08008040	080086D9	080086D9	080086D9	080086D9	Ù + . . Ù + . . Ù + . . Ù + . .
0x08008050	080086D9	080086D9	080086D9	080086D9	Ù + . . Ù + . . Ù + . . Ù + . .
0x08008060	080086D9	080086D9	080086D9	080086D9	Ù + . . Ù + . . Ù + . . Ù + . .
0x08008070	080086D9	080086D9	080086D9	080086D9	Ù + . . Ù + . . Ù + . . Ù + . .
0x08008080	080086D9	080086D9	080086D9	080086D9	Ù + . . Ù + . . Ù + . . Ù + . .

```
13:13:01 : [omnibusf4sd_bl.bin] opened successfully.
13:13:01 : [omnibusf4sd_bl.bin] checksum : 0x000E5189
13:14:21 : Memory programmed in 0s and 563ms.
13:14:21 : Verification...OK
13:14:21 : Programmed memory Checksum: 0x000E5189
13:18:18 : [yuneec_typhoon_h.bin] opened successfully.
13:18:18 : [yuneec_typhoon_h.bin] checksum : 0x066F205A
13:20:05 : Memory programmed in 24s and 687ms.
13:20:05 : Verification...OK
13:20:05 : Programmed memory Checksum: 0x066F205A
```

Debug in Low Power mode enabled. Device ID:0x413 Core State : Live Update Disabled

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.

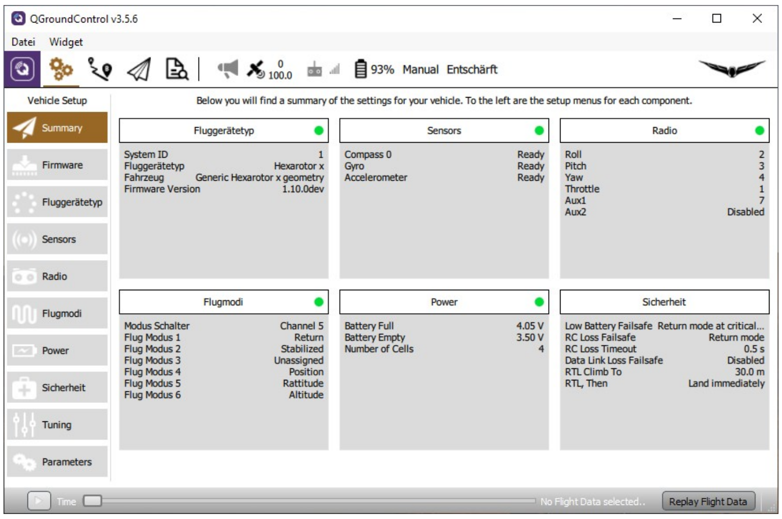
Reassemble the MCU board to the main board (ESC board). Check if it is correctly inserted.

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.



Below you will find a summary of the settings for your vehicle. To the left are the setup menus for each component.

Fluggerätetyp	
System ID	1
Fluggerätetyp	Hexarotor x
Fahrzeug	Generic Hexarotor x geometry
Firmware Version	1.10.0dev

Sensors	
Compass 0	Ready
Gyro	Ready
Accelerometer	Ready

Radio	
Roll	2
Pitch	3
Yaw	4
Throttle	1
Aux1	7
Aux2	Disabled

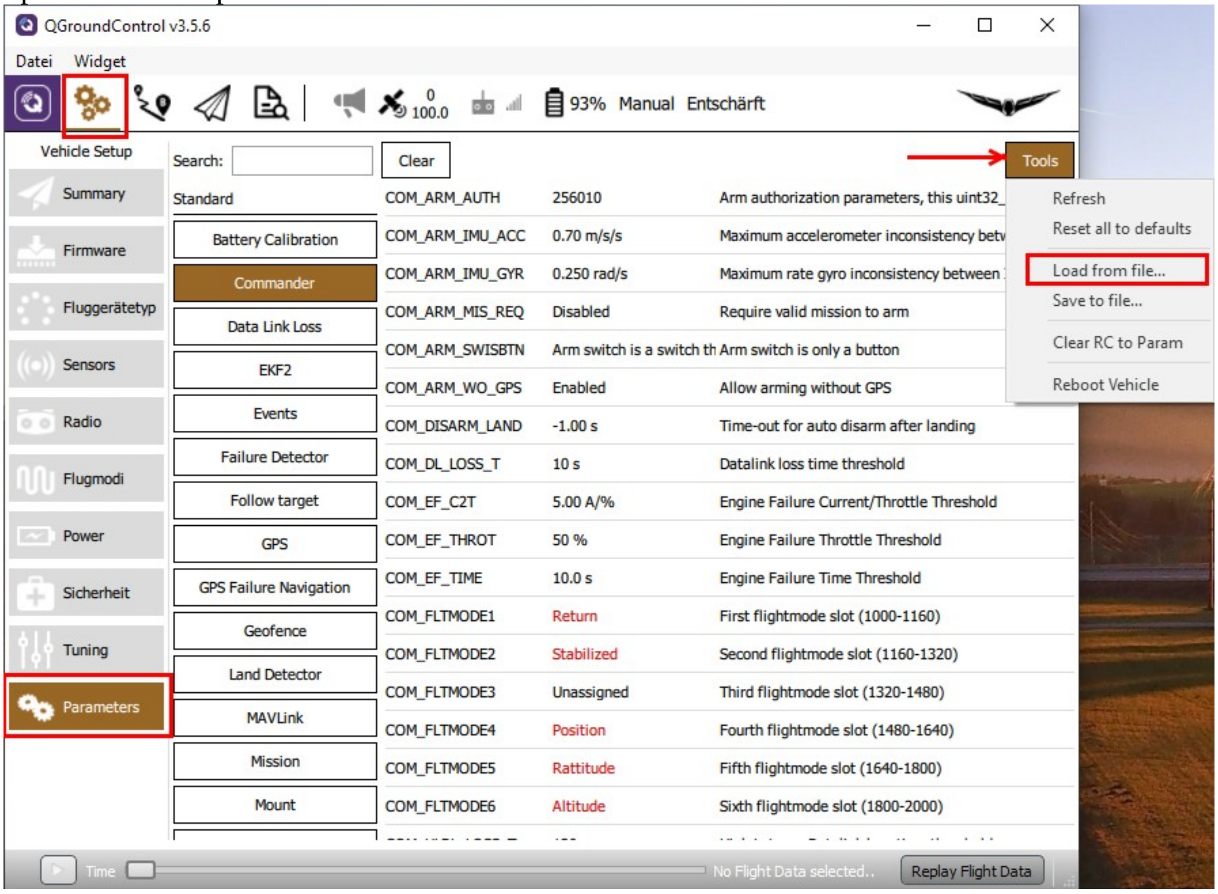
Flugmodi	
Modus Schalter	Channel 5
Flug Modus 1	Return
Flug Modus 2	Stabilized
Flug Modus 3	Unassigned
Flug Modus 4	Position
Flug Modus 5	Rattitude
Flug Modus 6	Altitude

Power	
Battery Full	4.05 V
Battery Empty	3.50 V
Number of Cells	4

Sicherheit	
Low Battery Failsafe	Return mode at critical...
RC Loss Failsafe	Return mode
RC Loss Timeout	0.5 s
Data Link Loss Failsafe	Disabled
RTL Climb To	30.0 m
RTL, Then	Land immediately

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.



COM_ARM_AUTH	256010	Arm authorization parameters, this uint32_
COM_ARM_IMU_ACC	0.70 m/s/s	Maximum accelerometer inconsistency betv
COM_ARM_IMU_GYR	0.250 rad/s	Maximum rate gyro inconsistency between
COM_ARM_MIS_REQ	Disabled	Require valid mission to arm
COM_ARM_SWISBTN	Arm switch is a switch th	Arm switch is only a button
COM_ARM_WO_GPS	Enabled	Allow arming without GPS
COM_DISARM_LAND	-1.00 s	Time-out for auto disarm after landing
COM_DL_LOSS_T	10 s	Datalink loss time threshold
COM_EF_C2T	5.00 A/%	Engine Failure Current/Throttle Threshold
COM_EF_THROT	50 %	Engine Failure Throttle Threshold
COM_EF_TIME	10.0 s	Engine Failure Time Threshold
COM_FLTMODE1	Return	First flightmode slot (1000-1160)
COM_FLTMODE2	Stabilized	Second flightmode slot (1160-1320)
COM_FLTMODE3	Unassigned	Third flightmode slot (1320-1480)
COM_FLTMODE4	Position	Fourth flightmode slot (1480-1640)
COM_FLTMODE5	Rattitude	Fifth flightmode slot (1640-1800)
COM_FLTMODE6	Altitude	Sixth flightmode slot (1800-2000)

Step 5: Set hardware and calibrate sensors

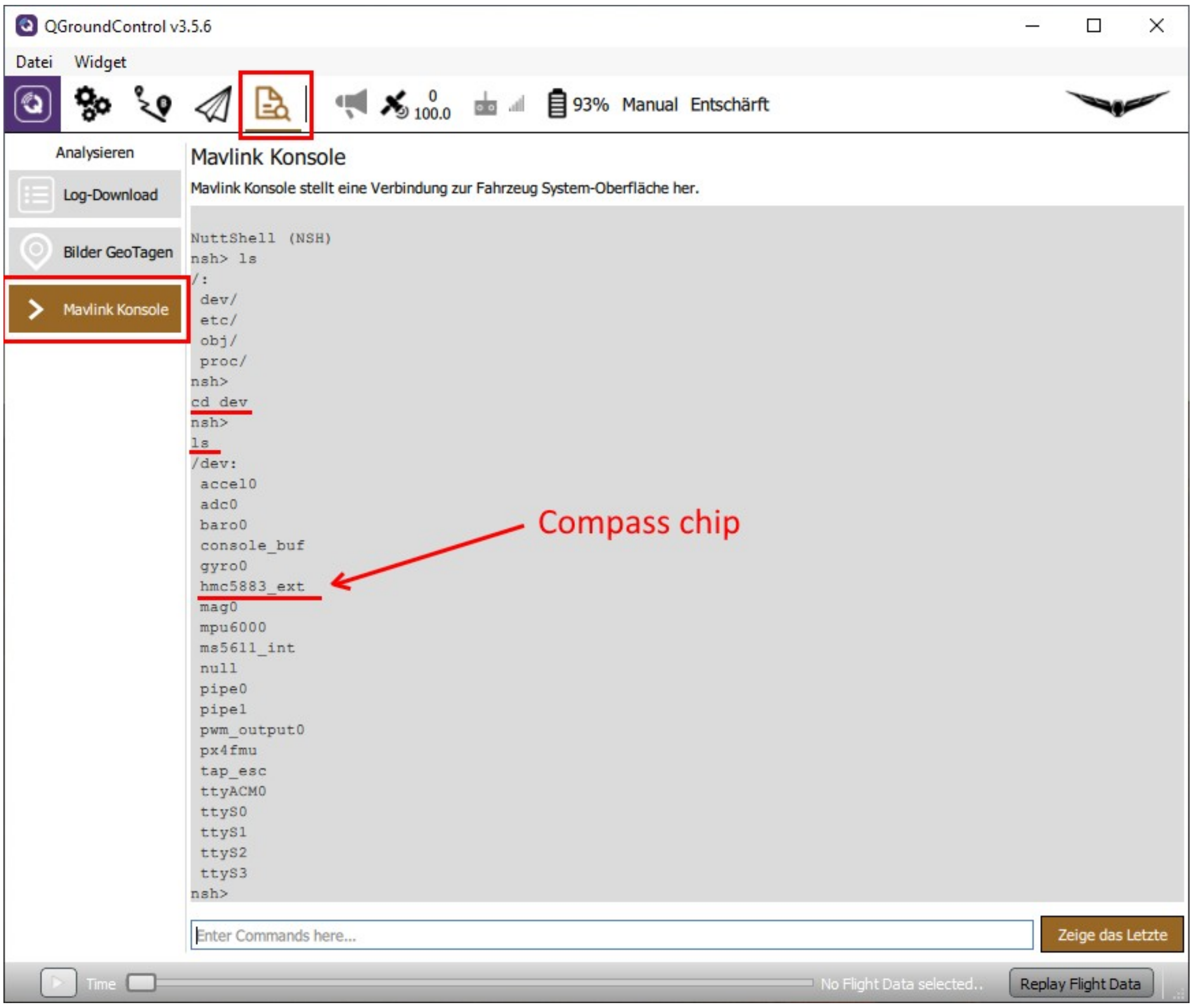
For Typhoon H exists two different compass chips. Older GPS boards have HMC5883, newer boards have IST8310 applied as compass chip. To find out which one you have, connect the drone to QGC and go to Mavlink console:

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.



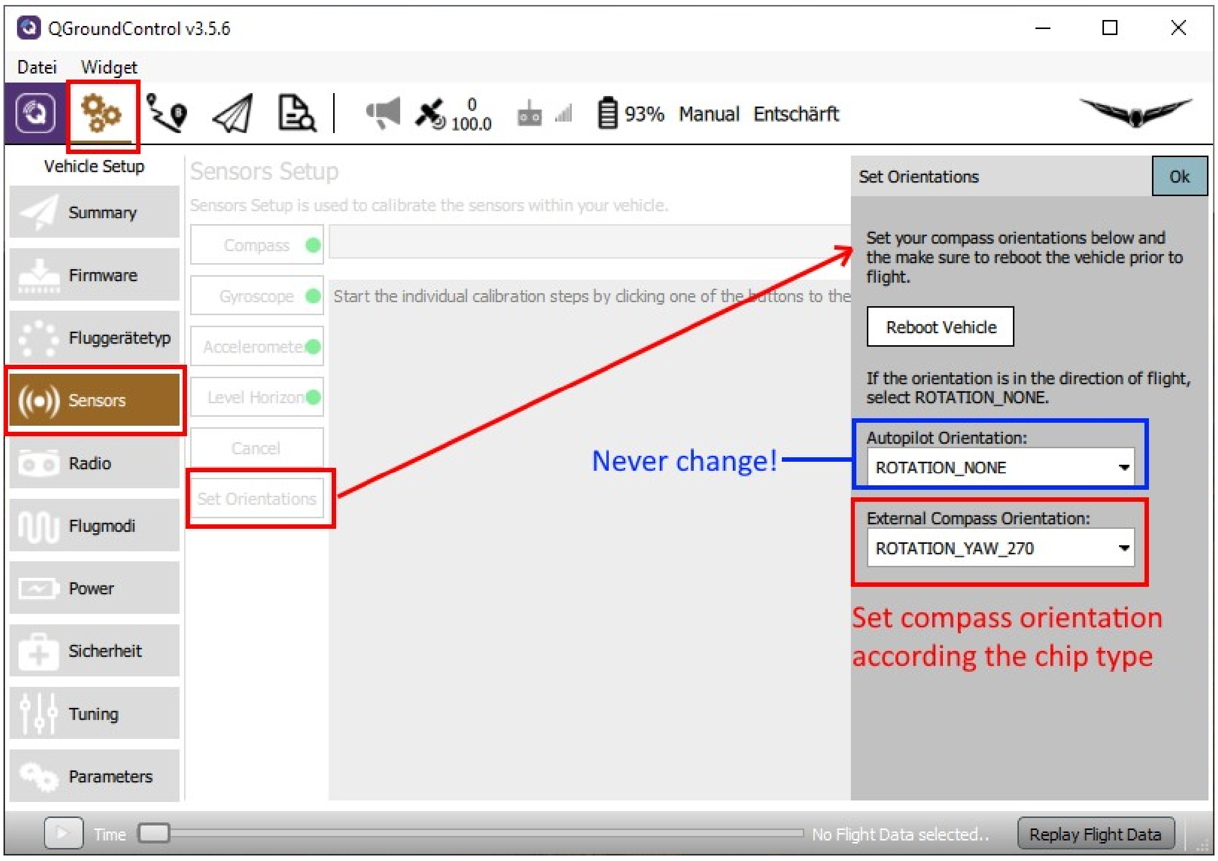
```
NuttShell (NSH)
nsh> ls
/:
 dev/
 etc/
 obj/
 proc/
nsh>
cd dev
nsh>
ls
/dev:
 accel0
 adc0
 baro0
 console_buf
 gyro0
 hmc5883_ext
 mag0
 mpu6000
 ms5611_int
 null
 pipe0
 pipe1
 pwm_output0
 px4fmu
 tap_esc
 ttyACM0
 ttyS0
 ttyS1
 ttyS2
 ttyS3
nsh>
```

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



Sensors Setup is used to calibrate the sensors within your vehicle.

Start the individual calibration steps by clicking one of the buttons to the

Set Orientations

Set your compass orientations below and the make sure to reboot the vehicle prior to flight.

If the orientation is in the direction of flight, select ROTATION_NONE.

Autopilot Orientation: ROTATION_NONE

External Compass Orientation: ROTATION_YAW_270

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.
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.