Thunderbird



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

The firmware is Open Source. **Please join the community and contribute!** The project can be found here: <u>https://github.com/tonirosendahl/Thunderbird</u> PX4 Autopilot: <u>https://docs.px4.io/</u>

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	
Preparation	
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: <u>https://github.com/tonirosendahl/Thunderbird/blob/Typhoon_H_480/Thunderbird_19122019_FT.zip</u>

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



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. <u>https://www.st.com/en/development-tools/stsw-link004.html</u>

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/

<u>∕</u>Motes:

- → 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.

Option Bytes				×
- Read Out Protection	r	BOR Leve		
Level 0	~ F	OFF	~ R	~
User configuration op IWDG_STOP WWDG_SW nSRAM_Parity SRAM2_RST SRAM2_PE nRST_SHDW nRST_STOP nRST_STOP NRST_MODE Security option bytes SEC_SIZE 0x0	IWDG_ST IWDG_UL FZ_IWDG FZ_IWDG PCROP_R NBoot0_SV NSWB001 VDDA_Mo	P _STOP _STDBY DP W_Cfg F0	nBoot0 nBoot1 nDBOOT DB1M B1M IRHEN SDADC12_VE	nBOOTO BOOT1 nBFB2 nBOOT_SEL DUALBANK BOREN DD_Monitor
Boot address option b BOOT_ADDO (H) BOOT_ADD1 (H) User data storage opt Data 0 (H) Flash sectors protection	ion bytes	Boot from Boot from Da		
Sector 0 Sector 0 Sector 1 Sector 2 Sector 3 Sector 3 Sector 4 Sector 5 Sector 6 Sector 6 Sector 7 Sector 8 Sector 9 Sector 10 < Unselect all	Start address 0x08000000 0x08004000 0x080020000 0x08010000 0x08020000 0x08040000 0x08060000 0x08060000 0x08060000 0x08060000 0x08060000 0x08060000 0x08060000 0x08060000	16 K 16 K 16 K 16 K 64 K 128 K 128 K 128 K 128 K 128 K	Protection No Protection	*
			Apply	Cancel

If success it looks like this:

🛓 STM32 ST-LII	NK Utility							_		\times
ile <u>E</u> dit <u>V</u> iev	/ <u>T</u> arget ST-L	INK External	Loader <u>H</u> elp							
 	🤹 🤣 💡	š 👔 🔜								
Memory display Address: 0x0	8000000 🗸 Sia	ze: 0x1000) Data Wi	idth: 32 bits	~	Device Device ID Revision ID Flash size	STM32F405xx/F40 0x413 Rev 2.0 Unknown	7xx/F415xx/F	417xx	
Device Memory @	0x08000000 :	Binary File				TIGST SIZE	ONNIOWI		LiveU	lpdat
arget memory, A	ddress range: [0x	08000000 0x080	01000]							
Address	0	4	8	С	ASCII					
x08000000	FFFFFFF	FFFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿ	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ÿÿÿ			
x08000010	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿ	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ÿÿÿ			
x08000020	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿ	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ÿÿÿ			
x08000030	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿ	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ÿÿÿ			
x08000040	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿ	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ÿÿÿ			
x08000050	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿ	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ÿÿÿ			
x08000060	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿ	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ÿÿÿ			
x08000070	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿ	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ÿÿÿ			
x08000080	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿ	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ÿÿÿ			
[>
3:09:30 : v2329i 3:09:50 : Conne										-
3:09:50 : SWD F 3:09:50 : Conne 3:09:50 : Debug 3:09:50 : Device 3:09:50 : Device 3:09:51 : Can no Disable	requency = 4,0 M ction mode : Norm in Low Power mod ID:0x413 family :STM32F40	al. de enabled. 05xx/F407xx/F4: tion and retry.	15xx/F417xx							
contract in option	-,									

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

Download [om	nibusf4sd_bl.bin]	×
Start address	0x08000000	
File path Extra options	D:\Yuneec_Kopter\Thunderbir Omr	d\Flashing_YTH_MCU\Thi Browse
	📃 Skip Flash Erase	Skip Flash Protection verification
Verification	• Verify while programming	O Verify after programming
Click "Start" to p	program target.	
After program	ning ☑ Reset after programming	Full Flash memory Checksum
	Start	Cancel

If success it looks like this:

5 STM32 ST-LIN	IK Utility						_		\times
<u>F</u> ile <u>E</u> dit <u>V</u> iew	/ <u>T</u> arget ST-L	INK External L	.oader <u>H</u> elp						
- 🖬 🖖	V 🌾 🤣	\$ 🙆 🔜							
Memory display						Device	STM32F405xx/F407xx/F415xx/F4	17xx	
Address: 0x08	8000000 V Siz	e: 0x2690	Data Wig	lth: 32 bits 🗸	1	Device ID	0x413		
]	Revision ID	Rev 2.0		
Device Memory @	0x08000000 :	File : omnibusf4so	d bl.bin			Flash size	Unknown	LiveU	ndate
Target memory, Ad									public
Address	0	4	8	С	ASCII				^
0x08000000	20020000	08001BF5	08001BF3	08001BF1	õ	óñ			
0x08000010	08001BF1	08001BF1	08001BF1	00000000	ññ	ñ			
0x08000020	00000000	00000000	00000000	08001BF3		ó			
0x08000030	08001BF3	00000000	08001BF3	080002D1	ó	óÑ			
0x08000040	08001BF1	08001BF1	08001BF1	08001BF1	ññ	ññ			
0x08000050	08001BF1	08001BF1	08001BF1	08001BF1	ññ	ññ			
0x0800060	08001BF1	08001BF1	08001BF1	08001BF1	ññ	ññ			
0x08000070	08001BF1	08001BF1	08001BF1	08001BF1	ññ	ññ			
0x08000080	08001BF1	08001BF1	08001BF1	08001BF1	ññ	ññ			
<									>
13:09:50 : Device 13:09:50 : Device 13:09:51 : Can no Disable 13:10:27 : Option 13:13:01 : [omnibu 13:13:01 : [omnibu 13:14:21 : Memor 13:14:21 : Verifica 13:14:21 : Program	family :STM32F40 t read memory! Read Out Protect bytes updated su usf4sd_bl.bin] ope usf4sd_bl.bin] che y programmed in 0 tionOK	tion and retry. ccessfully. ened successfully cksum : 0x000E5 Is and 563ms.	189						^
Debug in Low Powe	er mode enabled.		Device ID:0x4	13			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 [yun	eec_typhoon_h.bin]	×
Start address	0x08008000	
File path	D:\Yuneec_Kopter\Thunderbi	rd\Flashing_YTH_MCU\Thi Browse
Extra options	yuneed	_typhoon_h.bin
	🔄 Skip Flash Erase	Skip Flash Protection verification
Verification	 Verify while programming 	O Verify after programming
Click "Start" to p	program target.	
After program	ning ☑ Reset after programming	Full Flash memory Checksum
	Start	Cancel

If success it looks like this:

🖷 STM32 ST-LIN	NK Utility							_		\times
<u>File E</u> dit <u>V</u> iew	/ <u>T</u> arget ST-L	INK External L	.oader <u>H</u> elp							
🖹 🖥 👘	V 🌾 🏈 🧋	💈 🧟 🕺								
Memory display						Device	STM32F405xx/F407xx	/F415xx/F4	17xx	
Address: 0x0	8008000 🗸 Siz	e: 0xF0404	1 Data Wie	dth: 32 bits		Device ID	0x413			
						Revision ID Flash size	Rev 2.0			
Device Memory @	0x08008000 :	File : yuneec typ	hoon h.bin			Flash size	Unknown		LiveU	Indate
Target memory, Ad										
Address	0	4	8	С	ASCII					^
0x08008000	200053E2	080081B1	080086D9	080086D9	âS.±	Ù+Ù+				
0x08008010	080086D9	080086D9	080086D9	080086D9	Ù+Ù)+Ù+Ù+.				
0x08008020	080086D9	080086D9	080086D9	080086D9	Ù+Ù	J+Ù+Ù+.				
0x08008030	080086D9	080086D9	080086D9	080086D9	Ù+Ù	J+Ù+Ù+.				
0x08008040	080086D9	080086D9	080086D9	080086D9	Ù+Ù	J+Ù+Ù+.				
0x08008050	080086D9	080086D9	080086D9	080086D9	Ù+Ù	J+Ù+Ù+.				
0x08008060	080086D9	080086D9	080086D9	080086D9	Ù+Ù	J+Ù+Ù+.				
0x08008070	080086D9	080086D9	080086D9	080086D9	Ù+Ù	J+Ù+Ù+.				
0x08008080	080086D9	080086D9	080086D9	080086D9	Ù+ù)+Ù+Ù+.				_
<					· · ·					> `
13:13:01 : [ommb 13:13:01 : [ommb 13:14:21 : Memor 13:14:21 : Verifica 13:14:21 : Progra 13:18:18 : [yunee 13:20:05 : Memor 13:20:05 : Verifica 13:20:05 : Progra	usf4sd_bl.bin] che y programmed in 0 itionOK mmed memory Che ic_typhoon_h.bin] ic_typhoon_h.bin] y programmed in 2 itionOK	ecksum : 0x000E5)s and 563ms. ecksum: 0x000E5 opened success checksum : 0x06 24s and 687ms.	189 189 fully. 6F205A							ŕ
						r				~
Debug in Low Powe	er mode enabled.		Device ID:0x4	413			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.

O QGroundControl	v3.5.6			- 🗆 X
Datei Widget				
ية <mark>اي </mark> 🕲	🖉 🗟 🖪 🕷 🖞 📩	al 93% Manual Entschär	rft	
Vehicle Setup	Below you will find a summary	of the settings for your vehicle. To th	e left are the se	etup menus for each component.
A Summary	Fluggerätetyp 😑	Sensors	٠	Radio
Firmware	System ID 1 Fluggerätetyp Hexarotor x Fahrzeug Generic Hexarotor x geometry Firmware Version 1.10.0dev	Compass 0 Gyro Accelerometer	Ready Ready Ready	Roll 2 Pitch 3 Yaw 4 Throttle 3
Fluggerätetyp	Firmware Version 1.10.00eV			Throttle 1 Aux1 7 Aux2 Disabled
(c) Sensors				
o o Radio				
Flugmodi	Flugmodi 🔵	Power	٠	Sicherheit
Power	Modus Schalter Channel 5 Flug Modus 1 Return Flug Modus 2 Stabilized Flug Modus 3 Unassigned Flue Modus 4 Position	Battery Full Battery Empty Number of Cells	4.05 V 3.50 V 4	Low Battery Failsafe Return mode at critical RC Loss Failsafe Return mode RC Loss Timeout 0.5 s Data Link Loss Failsafe Disableo RTL Climb To 30.0 m
Sicherheit	Flug Modus 4 Position Flug Modus 5 Rattitude Flug Modus 6 Altitude			RTL, Then Land immediately
tuning				
Parameters				
				Flight Data selected.

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.

QGroundControl v Datei Widget	/3.5.6			- 0	×
ې 😓 😧	\land 🖪 🖪	× 100.0 📩 📶	93% Manual Er	ntschärft	-
Vehicle Setup	Search:	Clear		→	Tools
Summary	Standard	COM_ARM_AUTH	256010	Arm authorization parameters, this uint32_	Refresh
Firmware	Battery Calibration	COM_ARM_IMU_ACC	0.70 m/s/s	Maximum accelerometer inconsistency betv	Reset all to default
	Commander	COM_ARM_IMU_GYR	0.250 rad/s	Maximum rate gyro inconsistency between	Load from file
Fluggerätetyp	Data Link Loss	COM_ARM_MIS_REQ	Disabled	Require valid mission to arm	Save to file
Sensors	EKF2	COM_ARM_SWISBTN	Arm switch is a switch	th Arm switch is only a button	Clear RC to Param
с <i>т</i> ,		COM_ARM_WO_GPS	Enabled	Allow arming without GPS	Reboot Vehicle
Radio	Events	COM_DISARM_LAND	-1.00 s	Time-out for auto disarm after landing	
Fluqmodi	Failure Detector	COM_DL_LOSS_T	10 s	Datalink loss time threshold	
00 -	Follow target	COM_EF_C2T	5.00 A/%	Engine Failure Current/Throttle Threshold	1
Power	GPS	COM_EF_THROT	50 %	Engine Failure Throttle Threshold	and the second
Sicherheit	GPS Failure Navigation	COM_EF_TIME	10.0 s	Engine Failure Time Threshold	
	Geofence	COM_FLTMODE1	Return	First flightmode slot (1000-1160)	
Tuning	Land Detector	COM_FLTMODE2	Stabilized	Second flightmode slot (1160-1320)	
Parameters		COM_FLTMODE3	Unassigned	Third flightmode slot (1320-1480)	
	MAVLink	COM_FLTMODE4	Position	Fourth flightmode slot (1480-1640)	
	Mission	COM_FLTMODE5	Rattitude	Fifth flightmode slot (1640-1800)	
	Mount	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.



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.