

Seriously Pro

SP Racing H7 NANO Flight Controller

About

The Seriously Pro Racing H7 NANO Flight Controller (SPRacingH7NANO) was designed to give you all the CPU performance you need for a low-latency control-loop in a tiny form factor for ultra-light builds.

Analyze your race and flight telemetry/blackbox logs using the built-in 128MB flash.

The SPRacingH7NANO gives you all the features you need for the heart of your nano build, when paired with a VTX/OSD combo, such as the TBS Unify EVO VTX/OSD via CRSF on a UART, you get an ultra-light, easy-to-wire and full-featured FPV system with on-screen-display system in just two boards.

Software

The SPRacingH7NANO runs the open-source Betaflight flight control (FC) software which is based on Cleanflight. It has an ever-growing community of friendly developers and users. Being open-source means that you too can contribute to the system.

For latest and detailed software setup refer to the Betaflight wiki.

History

The hardware was designed by the lead developer of Cleanflight to be more capable than the previous-generation STM32F7/4/3-based boards and to set the benchmark for a nano-sized STM32H750 based board. It has other features not seen on most other nano-sized FCs, other than the blisteringly fast 400Mhz H7 CPU, including support for 8 motor outputs and integrated 2-6S BEC.

The SPRacingH7NANO is the second H7-based flight controller that runs Betaflight. The SPRacingH7EXTREME being the first.

Features

- Next-generation STM32 H7 processor with hardware floating point unit for efficient flight calculations and faster ARM-Cortex M7 core running at 400Mhz.
- 128MB External Flash for black box flight log recorder - optimize your tuning and see the results of your setup without guesswork.
- 2-6S integrated 5V BEC for powering directly from battery.
- Features the latest 32khz capable Accelerometer, Gyro sensor devices connected via the fast SPI bus.
- External On Screen Display (OSD) support via CRSF, etc.
- Solder-from-top design makes installation and maintenance simple.
- 8 Motor outputs. DSHOT/OneShot/PWM for ESCs and Servos.
- Supports direct connection of SBus, SumH, SumD, Spektrum1024/2048, CRSF, XBus receivers. No external inverters required (built-in). Legacy PWM 1 wire per channel receivers are not supported.
- Supports direct connection of 3.3v Spektrum Satellite receivers - use 3v3 pad for Satellite RX.
- 6 Serial Ports - None shared with the USB socket.
- Micro USB socket.
- Dedicated output for programmable LEDs - great for orientation, racing and night flying.
- Dedicated I2C port for connection of OLED display or MAG without needing flight battery.
- Battery monitoring for voltage and current.
- RSSI monitoring (analog only).
- Buzzer port for audible warnings and notifications.
- Developer friendly debugging port (SWD).
- Symmetrical design for a super tidy wiring.
- Optional connectors on NANO-E variant (2x JST-SH sockets + 2x Picoblade sockets for IO (I2C, SWD, UARTS, Motors, Power).
- Solder pads for all IO on NANO-S variant.
- Firmware upgradable via USB.
- Stackable design (Only NANO-E variant has stacking connector)
- Standard 20mm mounting hole pattern with 4mm holes and grommets for 3mm screws.
- LEDs for 3v, 5v and Status for easy diagnostics.
- Cleanflight, Betaflight and SP Racing logos.

WARNINGS

Failure to adhere to these warnings will void your warranty and destroy your flight controller.

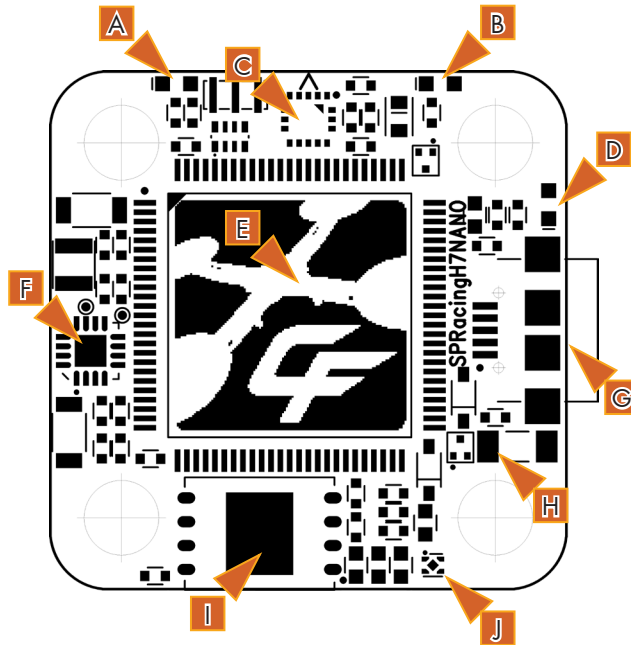
- Observe polarity at ALL TIMES. **Check** and **DOUBLE CHECK** before applying power. Do not rely on wire color-coding alone.
- **POWER OFF** before unplugging, plugging in or making any connections.
- Do not connect SOURCES of power to the pins marked with **5v**. They are **OUTPUTS** for supplying power to other devices.
- The **3.3v** supply is for low-current use only. 50mA MAX.
- Do not connect GND, **5v**, **VBAT** or **3.3v** to each other (short circuit).
- Do not connect GND, **5v**, **VBAT** or **3.3v** to any inputs or outputs unless specifically stated.
- Do not connect any input or output to any other input or output unless specifically stated.
- Do not connect all three pads of the 3 pin solder bridges, connect the center pad to one other pad **ONLY**.
- Do not allow dirt/dust/glue/liquid/etc into the pressure sensor (barometer) and microphone.
- Keep magnets away from the flight controller.
- Keep all conductive liquids away from the flight controller.
- Do not attempt to clean using an ultra-sonic cleaner - all the sensors be damaged.
- Do not use excessive force when inserting or removing MicroSD cards.
- Always align USB plug and socket when inserting/removing USB cables to prevent damage to the USB socket.
- Never power up any VTX without an antenna connected otherwise you will **DESTROY** the VTX.
- Do **NOT** rely on cable color-coding due to manufacturer variations.

GENERAL ADVICE

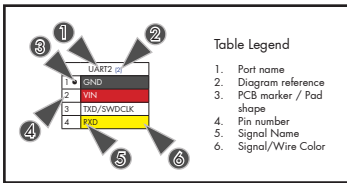
Follow the advice below for best performance and long-life of your flight controller:

- Apply resin/glue to reinforce JST-SH connectors - helps if you crash your aircraft.
- To further protect the board from crashes you can add a some additional solder to the edges of the JST-SH and USB sockets to reinforce them.
- Support connector sockets when inserting cables.
- Install open-cell foam over the pressure sensor for more accurate readings.
- If the noise from motors is too loud you can also install open-cell foam over the microphone.
- Do not cover the hole in baro and microphone sensors (e.g. with glue, resin, etc) or allow foreign object to enter it.
- Using color-coded wires and connectors is recommended, especially for BATTERY connections.
- For optimum performance do everything you can to prevent vibrations reaching the accelerometer/gyro sensor. e.g. balance motors, props use rubber isolation grommets and secure everything. Do not allow any wires to vibrate against the FC.
- If any incorrect flight behaviour is observed check props, motor bearings and other sources of vibrations **BEFORE** adjusting any software filters or other settings.

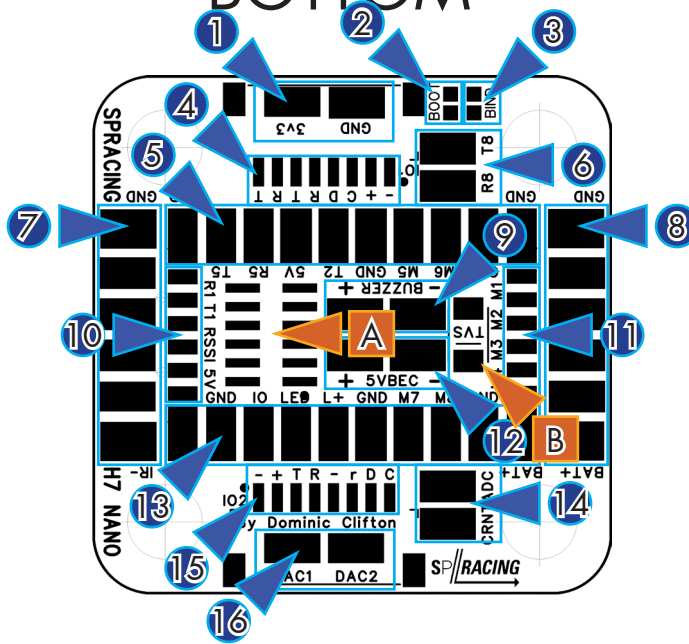
TOP



- A. Status LED (Red).
- B. 3V OK LED (Blue).
- C. Gyro/Accelerometer Sensor.
- D. 5V OK LED (Green).
- E. H7 400 Mhz CPU.
- F. 5V BEC.
- G. MicroUSB socket.
- H. Transponder Circuit.
- I. 128MB Flash.
- J. 3v3 Voltage Regulator.



BOTTOM



- A. Stacking connector (NANO-E only)
- B. TVS protection diode (fitted on NANO-E, user installable on NANO-S)

1. 3v3 + GND pads
2. Boot pads
3. Bind pads
4. IO-1 pads for JST-SH 8 pin socket.
5. IO-5 pads.
6. UART8 pads.
7. IO-RX pads.
8. IO-ESC pads.
9. Buzzer pads
10. IO-3 pads for Picoblade 6 pin socket.
11. IO-4 pads for Picoblade 6 pin socket.
12. 5V BEC pads.
13. IO-6 pads.
14. ADC pads.
15. IO-2 pads for JST-SH 8 pin connector.
16. DAC pads.

3.3v Pads (1)	
1	3.3v
2	GND

3.3v Pads (only accessible on NANO-S).

This pad is for use when connecting to 3.3v receiver, e.g. Spektrum Satellite receiver. **MAXIMUM 20mA**

Boot pads (2)	
1	3.3v (Bottom)
2	BOOT0 (Top)

Boot Pads.

Temporarily bridge these pads when powering on the board to enter CPU ROM bootloader mode. Do **NOT** use these pads for flight controller firmware upgrades.

Bind pads (3)	
1	3.3v (Bottom)
2	BIND (Top)

Bind Pads.

Temporarily bridge these pads when powering on the board to enter BOOTLOADER DFU mode. BOOTLOADER DFU mode can also be started from the flight controller. You can also use the pads for boot-loader selections other flight controller functions, but they are not designed for repeated use.

IO-1 JST-SH8 socket (4)	
1	● GND (right)
2	5v
3	SCL
4	SDA
5	RXD (R4)
6	TXD (T4)
7	RXD (R5)
8	TXD (T5) (left)

IO1 socket for I2C/UART4/UART5 (Only fitted on NANO-E)

Use this socket to connect a GPS receiver with I2C MAG sensors to use GPS features.

IMPORTANT: The SCL and SDA are 3.3v signals

"IO-1 / 5V" (2) supplies ~5V when powering via USB or BATTERY

IMPORTANT: 5.0v to 3.3v logic level converters are REQUIRED for SCL/SDA if your sensors use 5.0v signals.

UART4 R4/T4 and UART5 R5/T5 can use 3.3v/5.0v signals.

IO-5 pads (5)	
1	■ Battery - / GND (right)
2	■ GND
3	■ Motor Signal (M6)
4	■ Motor Signal (M5)
5	■ GND
6	■ TXD (T2)/SmartPort/HoTT
7	■ 5v
8	■ RXD (R5)
9	■ TXD (T5)
10	■ GND (left)

IO5 pads for Motor 5/6 and UART2/UART5

Use these pads for motor and IO signals.

UART2 TXD (T2) is bi-directional capable and is the preferred pin for SmartPort/HoTT or other single-wire bi-directional telemetry protocols.

UART5 RXD/TXD can be used for as a standard UART port, e.g. for GPS, SmartAudio, etc.

"IO-5 / 5v" (7) supplies ~5V when powering via USB or BATTERY

UART2 T2 and UART5 R5/T5 can use 3.3v/5.0v signals.

The IO5/GND (1) pad can be connected to Battery - and used to power the FC along with IO6/Battery+(1).

UART8 pads (6)	
1	■ TXD (T8) (top)
2	■ RXD (R8) (bottom)

UART8

Use for connecting VTX/ESC Telemetry/OSD/GPS/BlueTooth/RX Telemetry/Etc. 3.3v/5.0v signals.

IO-RX pads (7)	
1	■ GND (top)
2	■ RXD (R1)/PPM
3	■ TXD (T1)/TLM
4	■ RSSI
5	■ 5v
6	■ IR - (bottom)

IO-RX pads for UART1 / RSSI / IR (Only accessible on NANO-S)

For receiver connections.

Note: Some receivers like the FrSky XSR/X8R require the use of two serial ports for RX and Telemetry in this case use UART1 RXD for RX and UART2 TXD for Telemetry/SmartPort.

RSSI is for 0 - 3.3v Analog RSSI ONLY.

"IO-RX / 5v" (5) supplies ~5V when powering via USB or BATTERY

IR - is for connection to the SHORT leg of an IR Transponder LED.

IO-ESC pads (8)	
1	■ Battery - / GND (top)
2	■ Motor Signal (M1)
3	■ Motor Signal (M2)
4	■ Motor Signal (M3)
5	■ Motor Signal (M4)
6	■ Battery + (bottom)

IO-ESC pads for Battery and Motors 1-4 (Only accessible on NANO-S)

For motor signals and power.

When using 4S/5S/6S battery the TVS protection diode **must** be installed otherwise the high voltages that occur when batteries are connected will destroy the on-board BEC.

The "IO-ESC/GND" (1) and "IO-ESC/Battery+" (6) pads can be connected to the Battery -/+ to power the FC.

BUZZER (9)	
1	■ BUZZER- (B-) (right)
2	■ BUZZER+ (B+) / 5v (left)

BUZZER

Use 5V and BUZZER- to connect to an external buzzer.

"BUZZER / BUZZER+ (B+) / (5v)" (2) supplies ~5V when powering via USB or BATTERY

IO-3 socket (10)	
1	■ GND (top)
2	■ RXD (R1)/PPM
3	■ TXD (T1)/TLM
4	■ RSSI
5	■ TXD (T2)/SmartPort/HoTT
6	■ 5v (bottom)

IO3 socket for RX UART1 / UART2 / RSSI (Only fitted on NANO-E)

For receiver connections.

Note: Some receivers like the FrSky XSR/X8R require the use of two serial ports for RX and Telemetry in this case use UART1 RXD for RX and UART2 TXD for Telemetry/SmartPort.

RSSI is for 0 - 3.3v Analog RSSI ONLY.

"IO3 / 5v" (6) supplies ~5V when powering via USB or BATTERY

IO-4 socket (11)	
6	■ Battery - / GND (top)
5	■ Motor Signal (M1)
4	■ Motor Signal (M2)
3	■ Motor Signal (M3)
2	■ Motor Signal (M4)
1	■ Battery + (bottom)

IO-4 pads for POWER and Motors 1-4 (Only fitted on NANO-E)

For motor signals and power.

When using 4S/5S/6S battery the TVS protection diode **must** be installed otherwise the high voltages that occur when batteries are connected will destroy the on-board BEC.

5V/GND (12)	
1	■ GND (right)
2	■ 5v BEC (left)

5V BEC/GND

5V BEC output. e.g. for 5V camera/VTX systems or LED strip power.

"5V BEC/GND / 5v BEC" (2) only supplies ~5V when powering via BATTERY. MAXIMUM 1A

IO-6 pads (13)	
1 ■	Battery + (right)
2 ■	GND
3 ■	Motor Signal (M8)
4 ■	Motor Signal (M7)
5 ■	GND
6 ■	5v BEC
7 ■	LED STRIP
8 ■	IO
9 ■	GND
10 ■	IR +

IO5 pads for Motor 7/8 and LED STRIP, IO and accessory power.

Use these pads for motor and IO signals.

There are additional pads for 5V, 5V BEC and Battery+ power which can be used to power accessories/peripherals.

The Battery+ pad can be used to power the FC along with "IO-5/ GND" (1).
Do **NOT** connect Battery- to "IO-6/GND" (2/5/9).

"LED STRIP" is for connection to the DATA IN of a compatible LED Strip. Note that some LED strips require a MAXIMUM voltage LESS than 5V. In this case you may need to connect a DIODE between the LED strip + and the 5V BEC pad to drop the voltage to around 4.5v. The LED Strip signal pad is marked "LED" and closest to the middle of the board. When using LED strips ensure you do not use more than 1A total power from the 5V BEC

"IO-6 / 5v BEC" (6) only supplies ~5V when powering via BATTERY.
"IO-6 / IR +" (10) supplies ~5V when powering via USB or BATTERY.

IR + is for connection to the LONG leg of an IR Transponder LED.

ADC pads (14)	
1 ■	ADC (top)
2 ■	CRNT (bottom)

ADC

"ADC" is a spare ADC input.

"CRNT" is for connection to a 0-3.3v current sensor output from a PDB/4in1 ESC.

IO2 JST-SH8 socket (15)	
1 -	GND (left)
2 +	5v
3 T	TXD (T3)
4 R	RXD (R3)
5 -	GND
6 r	NRST
7 D	SWDIO
8 C	SWDCLK (right)

IO1 socket for UART3 and SWD debugging (Only fitted on NANO-E)

Use this socket to connect a GPS receiver with I2C MAG sensors to use GPS features.

"IO-2 / 5V" (2) supplies ~5V when powering via USB or BATTERY

UART3 R3/T3 can use 3.3v/5.0v signals.

For debugging use a compatible SWD debugger. Black Magic Probe, ST-Link, Segger J-Link, etc.

IMPORTANT: Do not attempt to reflash the firmware on the H7 CPU via SWD. Firmware is stored on external flash.

ADC pads (14)	
1 ■	ADC (top)
2 ■	CRNT (bottom)

DAC (only accessible on NANO-S)

"ADC" is a spare ADC input.

"CRNT" is for connection to a 0-3.3v current sensor output from a PDB/4in1 ESC.

Receiver Connections

The H7NANO has many methods to allow receiver connections, the main ones are as follows (listed in order of popularity):

- 1) One-wire Serial RX on UART1 via IO-3/IO-RX + Telemetry on UART2 via IO-3/IO-5
- 2) Two-wire Serial RX on UART1 via IO-3/IO-RX (e.g. FPort)
- 2) Serial RX on UART1 via IO-3/IO-RX + 3v3 PAD (Spektrum Satelelite)
- 3) PPM via IO-3/IO-RX + Telemetry on UART1 via IO-3/IO-RX (e.g. FrSky D4R-II)

TVS protection diode

The NANO-E has the TVS protection diode pre-installed.

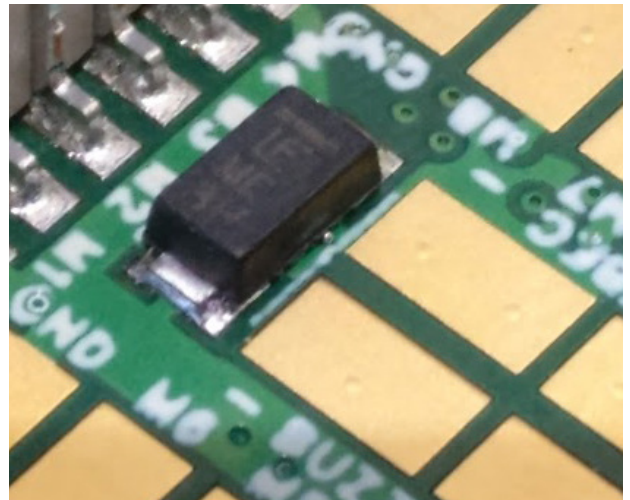
The NANO-S requires the fitting of the supplied TVS protection diode when using 4S/5S/6S.

Failure to install the TVS will result in the FC being destroyed by the high voltages that occur when connecting batteries. The voltage spikes are often more than double the battery voltage!

The TVS diode must be installed in the correct orientation, the line on the diode must match the line on the PCB.

To install the diode follow these simple steps:

- Quickly apply solder only to one pad on the PCB, remove the tip, don't burn off the flux.
- Quickly re-heat the solder on the pad and slide one leg of the diode into the molten solder.
- Remove soldering iron tip and hold diode in place until solder solidifies. Then solder the other side of the diode to the pad.



Separate ESC Connections.

If you are using separate ESC and a PDB, with one ESC per motor, proceed as follows.

1. Connect a pair of wires with a MALE connector to a PDB's battery +/- pads. The battery itself should have a FEMALE connector.
2. Connect each ESC's +/- to a PDB's +/- pads.
3. Connect a pair of wires from the PDB's power out +/- pads to the FC's Battery +/- pads (opposite side from the USB connector on the FC)
4. Connect a GND and a Motor signal wire from the FC to each ESC.

There should be 4 wires to each ESC, 2 to the battery and 2 from the PDB to the FC, 20 total.

If your PDB has a current sensor then you can connect the PDB's current sensor 0-3.3v ADC output signal to the CRNT pad.

Not all GND connections on the FC are suitable for powering the FC even though they are all connected; so only use "Battery +" and "Battery -" pads listed above for battery/pdb connections. i.e. do not use a "Battery +" pad and a "GND" that doesn't have "Battery -" next to it in the tables listed above.

4in1 ESC Connection.

If you are using a 4in1 ESC/PDB combo, one ESC for 4 motors, then proceed as follows:

1. Connect FC battery wire to 4in1's Battery +/- pads. Use a MALE connector on the FC battery lead as the battery itself should have a FEMALE connector.
2. Connect a pair of wires from the 4in1's battery out +/- pads to the FC's Battery +/- pads (opposite side from the USB connector on the FC)
3. Connect Motor signals from 4in1 ESC to FC's motor signal pads.

If your 4in1 ESC has a current sensor then you can connect the 4in1 ESC's current sensor 0-3.3v ADC output signal to the CRNT pad.

OSD Connections

The H7NANO supports serial interface control of Camera or VTX-based OSDs. e.g. TBS Unify EVO OSD/VTX, some RunCam camera, etc. Check betaflyght wiki for supported devices and general configuraton.

The FC sends data to the devices using any spare serial port, UART8 is suitable for this.

The devices generally use a cross-over UART wiring system.

* FC UART(x) TXD to VTX/Camera UART RXD

* FC UART(x) RXD to VTX/Camera UART TXD

Please refer to Camera/OSD manuals before making connection.

Cables

IMPORTANT: Ensure that you leave some slack in your cable routing as this will help if you crash your aircraft.

Spare or replacement cables are available from your retailer.

Do **NOT** rely on cable color-coding due to manufacturer variations. Always check before applying power.

Soldering

VERY IMPORTANT! - Do NOT solder your flight controller until you have plugged in a USB cable and checked that the GREEN, BLUE and RED lights operate! Red will flash, GREEN and BLUE must be ALWAYS ON. Once you have checked the FC is OK then unplug it and solder your wires.

IMPORTANT:

- Use a high quality soldering iron and good solder.
- Tin/Lead solder is MUCH easier to use than other Lead-free solder.
- Use solder with flux and remove any flux residue after soldering. AVOID CORROSIVE FLUX!
- Check for and remove solder balls you may have created after soldering - use a magnifying glass.
- Ensure you have sufficient fume extraction when soldering.
- Pin headers that connect to the GND signal and large battery/esc pads will be more difficult to solder because the PCB will sink the heat from your soldering iron. Solder the signal pins, then the VIN pins, then the GND pin or large pads - by the time you get to the GND pins or large pads you will have heated the board and it will be easier to solder them.
- If you have never soldered before then DO NOT attempt to solder the flight controller, practice on something else first.
- Position your soldering iron tip so that you do not accidentally de-solder or burn other components on the board.
- After soldering check for short circuits between all combinations of GND, BATTERY+, 5V, 5V BEC, 3V, if a short is found DO NOT power up the FC. Find the short and remove it, re-check for shorts.

The flight controller may be supplied with some pin-headers and connectors. Choose very carefully which ones you want to use. You can solder headers to top or the bottom of the board.

Once you have soldered pin headers or connectors in place DO NOT attempt to remove them unless you are highly skilled in de-soldering and have the correct tools. Overheating the board or components will destroy it.

Blackbox Logging

The NANO has a 128MB flash chip, most of which is available for black-box logging.

For full details on how to enable logging, and how to view logs please see the Betaflight software manual/wiki.

Your aircraft will fly better if you analyze your logs and tune accordingly.

Getting started

Verify flight controller operation via the configuration software.

- Disconnect the flight controller from your computer.
- Disconnect ALL connectors and headers from the flight controller.
- Install latest STM32 Virtual COM Port Driver
<http://www.st.com/web/en/catalog/tools/PF257938>
- Install and launch the Betaflight Configurator tool
<https://github.com/betaflight/betaflight-configurator>
- Connect flight controller to computer via USB cable.
- Select the correct COM port if it is not automatically detected.
- Verify all sensors on your board are giving correct readings.
- Disconnect and **upgrade the firmware** if new firmware is available.

For further software configuration see the getting started guide in the Betaflight wiki or manual.

The basic steps, after firmware upgrade, are as follows.

- Choose board alignment - you can mount it in any orientation, not just with the arrow facing forwards.
- Calibrate sensors.
- Configure serial ports.
- Choose model/mixer (default is Quad X)
- Enable features.
- Configure receiver, set channel mapping.
- Configure channel mid and endpoints (1000-2000) and trim channels on transmitter.
- Configure voltage/current monitoring.
- Configure outputs (servos/ESCs)
- Ensure ESC calibration matches ESC configuration, recalibrate ESCs if needed.
- Learn about flight modes and configure channels/switches to activate them as required.
- Learn how to arm/disarm.
- Bench-test failsafe without props attached.
- Read safety notes.
- Learn how to download and view your flight logs to help tune your aircraft.
- Enable blackbox logging for your first flight (so you have a log).
- Learn how to recognise un-tuned flight characteristics and the effects of a PID controller. (Watch some videos).
- First flight should be in Acro/Rate mode (the default mode when no other modes are active).
- Tune PIDs.
- Backup settings.
- Contribute to the Betaflight project with feedback, suggestions, code, etc.

Firmware Upgrade

VERY IMPORTANT:

It is **REQUIRED** that you immediately upgrade the firmware of the flight controller so that you have the latest features and bug fixes. **DO NOT ATTEMPT TO FLY** until you have installed the **latest firmware**. Due to the fast-paced development cycle of Betaflight and manufacturing lead-times the firmware installed in the factory may not have all features enabled and issues may have been fixed.

On Windows USB DFU drivers must be installed.

1. Click the Firmware Flasher tab.
2. Select the correct COM port and speed, use the default speed unless you have changed it on the FC.
3. Select the latest "SPRACINGH7NANO" stable release. (Do **not** use other targets). If no firmware is currently available or the target is not listed do **NOT** attempt to flash and refer to the betaflight wiki/issue tracker for firmware availability. **NOTE: Flashing the wrong firmware can cause the FC to be permanently damaged.**
4. Click 'Load firmware [Online]' and wait for firmware to download and read release notes before flashing.
5. Click 'Flash Firmware'. At this point the FC should reboot into DFU mode. DFU will appear in the list of ports and flashing via DFU will proceed. After flashing the virtual COM port will re-appear.

IMPORTANT: Verify operation using latest official firmware before attempting to use alternative firmware. Not all features may be supported by alternative firmware.

Official firmware:

<https://github.com/spracing/betaflight/issues/1>

Official source:

<https://github.com/spracing/betaflight/issues/2>

Bootloader

With the The SP Racing H7 NANO the Betaflight/Cleanflight software is NOT stored on the CPU itself, instead the software is stored on the external flash and loaded into the CPU's RAM when the FC is powered on.

The SP Racing H7 NANO has a small program, called a bootloader, which handles this task.

The bootloader will start as soon as the CPU powers on, and it will look for firmware and attempt to load it immediately, without interfaction.

If you need to enter the bootload you can do it from the configurator tool or by keeping the bind pads bridged when powering on the FC. When the bind pads are bridged on boot the status led will blink very fast until the bridge is removed. When the bridge is removed the bootloader software will flash the status led slowly and the FC will be in Bootloader DFU mode.

When the bind pads are kept bridged during SP Racing bootloader mode the LED will flash once every 2 seconds, each LED flash indicates a different function. keep the bind pad bridged until the LED has flashed the amount of times that correspond to the function number below, then un-bridge the pads to start the function.

You can bridge the pad with a pair of conductive tweezers, paperclip, etc. Be careful not to short anything else against the item used to bridge the pads!

Bootloader functions:

- 1 Load firmware (looks for valid firmware on external flash at offset 0x97CE0000)
- 2 Launch loaded firmware in RAM (for developers).
- 3 Clear firmware in RAM.
- 4 Enter CPU ROM DFU bootloader.
- 5 Reboot.
- 6 Erase config on external flash.
- 7 Erase firmware on external flash.
- 8 Erase entire external flash.
- 15 Erase CPU bootloader (**DO NOT DO THIS - You will have to return the FC to the factory for reprogramming**).
- 16 Erase external flash reserved area. (Do NOT do this unless instructed by Seriously Pro / SP Racing)

IMPORTANT: The bootloader is non-user-upgradable. If you erase it you will not be able to use the FC until a bootloader is restored to it by the factory, at your cost! If you are a developer you can erase it and write your own if you like; the bootloader has a function (15) which allows you to erase it for this purpose only.

Failure patterns:

When the bootloader can't do something, it will give a failure pattern. The patterns starts with a few quick flashes, then some slow flashes. Count the slow flashes and use the table below.

- 2 = Firmware verification failed. Corruption, hash mismatch.
- 3 = Firmware load failed. No valid firmware found.
- 4 = Firmware write failed (e.g. when flashing).
- 5 = External flash write failed.
- 6 = External flash read failed.
- 7 = SD write failed.
- 8 = SD read failed.

There are other codes, if you experience them contact SP Racing support with your symptoms.

Note: The betaflight failure patterns start with fewer and slower 'fast' flashes than the bootloader failure patterns which begin with much faster 'fast' flashes.

Firmware flashing for developers

Manual flashing can be performed when the SP Racing bootloader is in DFU mode. The SP Racing bootloader allows full access to the external flash. You can also download Blackbox logs from the external flash via DFU! - very fast!

Firmware Program

```
dfu-util -D obj/betaflight_4.0.0_SPRACINGH7NANO_EXST.exst -s 0x97CE0000:leave
```

Firmware Erase

```
dd if=/dev/zero ibs=1k count=448 of=ZERO_448K.bin  
dfu-util -D ZERO_448K.bin -s 0x97CE0000:0x70000
```

Firmware Read

```
dfu-util -U fw_448K.bin -s 0x97CE0000:0x70000
```

Blackbox Log Read

```
dfu-util -U bb_log.bin -s 0x90000000:0x7CC0000
```

Config Erase

```
dd if=/dev/zero ibs=1k count=128 of=ZERO_128K.bin  
dfu-util -D ZERO_128K.bin -s 0x97CC0000:0x20000
```

Config Read

```
dfu-util -U config.bin -s 0x97CC0000:0x20000
```

The SP Racing bootloader firmware supports SWD debugging of the betaflight software code, just set breakpoints in the Betaflight code as you would normally. Refer to the BF source for BL address space usage.

Troubleshooting

Q: Unable to flash firmware via DFU.

A: Likely the correct DFU drivers are not installed. Put the FC in SP Racing BOOTLOADER mode by bridging BIND pads and connecting the USB cable. Un-bridge BIND pads when the status LED flashes fast repeatedly. Then update DFU drivers. See USB Flashing section of Betaflight manual for details.

Q: The COM port does not show up.

A: Ensure latest USB STM VCP (Virtual Com Port) drivers are installed.

Q: DFU does not appear in the port list when flashing.

A: Try disconnecting the FC. See above for how to enter SP Racing bootloader. Connect the USB cable. Retry flashing when in SP Racing bootloader mode.

Q: I have no LEDs on at all.

A: Check 5v supply. (Battery Charged?, USB cable damaged?)

Q: The status LED never lights or is always on.

A: Follow recovery procedure in the configurator. Likely caused by flashing wrong firmware. Could also be caused by stuck-down BOOT switch - check button for dirt, etc. You can also use the SP Racing Bootloader to erase/reset the FC configuration to defaults which may also help.

Q: Unable to connect and a repeating light sequence occurs on the activity LED.

A: Check the Betaflight manual for how to interpret the error code (count the long flashes).

Q: Unable to connect to the board (and status LED shows non-repeating pattern at boot-up).

A: Close configurator, Close all Chrome browsers and Chrome apps, retry. COM Port drivers not installed? Try connecting via different UART OR Reset the board to defaults using buttons OR reflash firmware using 'full chip erase'. (Likely caused by mis-configuration of ports).

Q: Transponder code not recognised by receiver.

A: Check transponder enable bridge. Check IR LED orientation and light beam exit path. Ensure transponder feature enabled in software. Ensure transponder code correctly configured.

Q: The COM port does not show up after flashing firmware.

A: Wrong firmware was flashed; Use SP Racing bootloader to erase the firmware and config, reflash correct firmware.

Q: The receiver tab does not show any activity.

A: Check configuration tab - receiver mode. Center sticks, configure endpoints and trim on transmitter. Check ports tab if using Serial RX. Check wiring.

Q: You have a problem not listed here.

A: Reset and/or upgrade the firmware, try again, report issues via the forums - links are in the configurator on the 'Documentation & Support' tab. Before contacting your retailer reflash with latest Betaflight, reset FC to default setting and double check your issue - perhaps the firmware you are using is old or incorrect?

Q: Only 5V LED lights up, No BLUE 3V LED or RED STATUS LED.

A: Most common cause is misconnection or short of power/battery wires - FC will probably be destroyed! Check for short in Spektrum Satellite cable if using Spektrum Satellite RX. Normally only caused by destroyed CPU or overloaded/shorted voltage regulator.

Mounting

The SPRacingH7NANO is supplied with grommets. Install the grommets in each corner of each board.

The SP Racing H7 NANO has a directional arrow on it indicating the front of the board. It doesn't matter which way round or which way up it is installed, but if the arrow is not facing forwards then you will have to configure the board orientation in the configurator.

Ensure that no wires are loose or vibrate against the FC or gyro.

Use appropriate guage cable and solder neatly! Especially for the Battery / ESC power cables.

Read SOLDERING section of this manual, there's great advice in that section, even if you've soldered before.

Transponder

The SPRacingH7NANO features a IR LED transponder system. For optimum performance ensure you read the following section.

Mounting - Ensure that the IR LED can shine light from its installed location onto the track-side receiver, unobstructed. e.g. ensure that props, motors, arms, etc, are not in the way of the light shining from the LED.

Orientation - Ensure the LED points outwards from the aircraft towards the track-side receivers. The more receivers you use the better the code reception will be.

Verifying operation - Ensure that the TRANSPONDER feature is enabled in Betaflight. Ensure that the TRANSPONDER code has been configured via the Race Transponder configuration section in the Betaflight Configurator. Once this is done you can use a CMOS/CCD camera without IR block pointed directly at the IR LEDs and you should see them pulsing. A mobile phone camera works well for this, your FPV camera and screen/goggles may be OK if it doesn't block IR light.

Once you have verified that the IR LED pulses IR light then you can scan your SPRacingH7NANO past one of the iLAP receivers. The iLAP receiver just needs power, no software configuration is required to verify that the code is working.

The first time the iLAP receiver can read the transponder code the LAP light will pulse once, while the iLAP receiver can read the code then the activity light (ACT) will be flashing.

Each time the LAP light flashes the receiver transmits transponder code and timing information via it's COM port to the computer it is attached to.

By default the iLAP receivers will not transmit the transponder code to the PC twice in a row unless the transponder code has not been received for over one second.

For further iLAP receiver configuration and setup advice refer to the iLAP documentation.

Hardware Specifications

- STM32H750 CPU, 400MHz inc FPU
- 128MByte 1Gbit NAND flash via QuadSPI
- Low-noise ICM20602 accelerometer/gyro with dedicated filtering(connected via SPI)
- 1.0mm thick 4-layer copper gold-plated PCB
- 2-6S BEC 5V Switching regulator, 1A
- TVS protection diode (NANO-E: Fitted, NANO-S: optional extra)
- Transponder circuitry (LED and code available separately)
- Buzzer circuitry
- RSSI Analog input
- 8 motor outputs (NANO-S: 8 on pads, NANO-E: 4 on 4in1 connector, 4 on pads)
- 1x Dual SPI + GPIO breakout onto stacking connector (NANO-E only)
- 6 Serial Ports (5x TX+RX + 1x TX only bi-directional)
- 3 LEDs for 5V, 3V and STATUS (Green, Blue, Red)
- 26.5x26.5mm PCB with 20mm mounting hole pattern
- 4mm mounting holes for soft-mount grommets and M3 bolts
- MicroUSB socket for configuration and ESC programming
- Bootable from External flash.
- Supplied with 4x soft-mount grommets.
- Optionally supplied with 2x JST-SH 8-way IO cables. (NANO-E only)
- Optionally supplied with 2x PicoBlade 6-way IO cables. (NANO-E only)
- 1x solder pad for LED Strip
- 2x solder pad for DAC out (NANO-S only)
- 2x solder pad for ADC in (for 4in1 current sensor output, etc)
- 2x solder pads for UART8 RX/TX
- 2x solder pads for 5V/GND power
- 2x solder pads for Buzzer
- 2x solder pads for TVS Diode
- 1x Row of pads for Motor 1-4 + Battery wires (NANO-S only)
- 1x Row of pads for RX connections (UART1 RX+TX, RSSI, 5V, GND, IR) (NANO-S only)
- 2x Rows of pads for additional IO (UART2,UART5,IR,LED-Strip,etc)
- 2x 8pin JST-SH socket for GND/5V/I2C/UART4/UART5 (IO port, e.g. for external GPS module)
- 2x 8pin JST-SH socket for GND/5V/SWD/UART3 (IO port, e.g. for debugging, etc)
- 1x 6pin PicoBlade socket for RX (NANO-E only)
- 1x 6pin PicoBlade socket for 4in1 ESC (NANO-E only)
- 1x solder pads for BOOT
- 1x solder pads for BIND
- Cleanflight and Betaflight logos - they're on there, you just have to find them
- SP Racing logo
- 1x Additional easter egg!

Credits and acknowledgements

Hardware design and manufacturing by Dominic Clifton.

Manual by Dominic Clifton.

Seriously Pro website by Dominic Clifton.

Blackbox flight logging component, tools and SD Card support by Nicholas Sherlock.

The Cleanflight/Betaflight flight controller software is based on MultiWii by Alex Dubus.

The GUI tool was based on software by cTn.

Thanks to Painless360 / Lee Schofield for ideas and tutorial videos.

<https://www.youtube.com/user/Painless360/playlists>

Thanks to everyone who provided feedback on previous SP Racing products.

Software support and contributions from many, many authors. For a complete list see github contributors.

<https://github.com/cleanflight/cleanflight/graphs/contributors>

Community

Thanks to everyone from the AWESOME Cleanflight/Betaflight/iNAV communities for code, artwork, support, ideas, feedback, and everything else. Without you all this product would not exist.