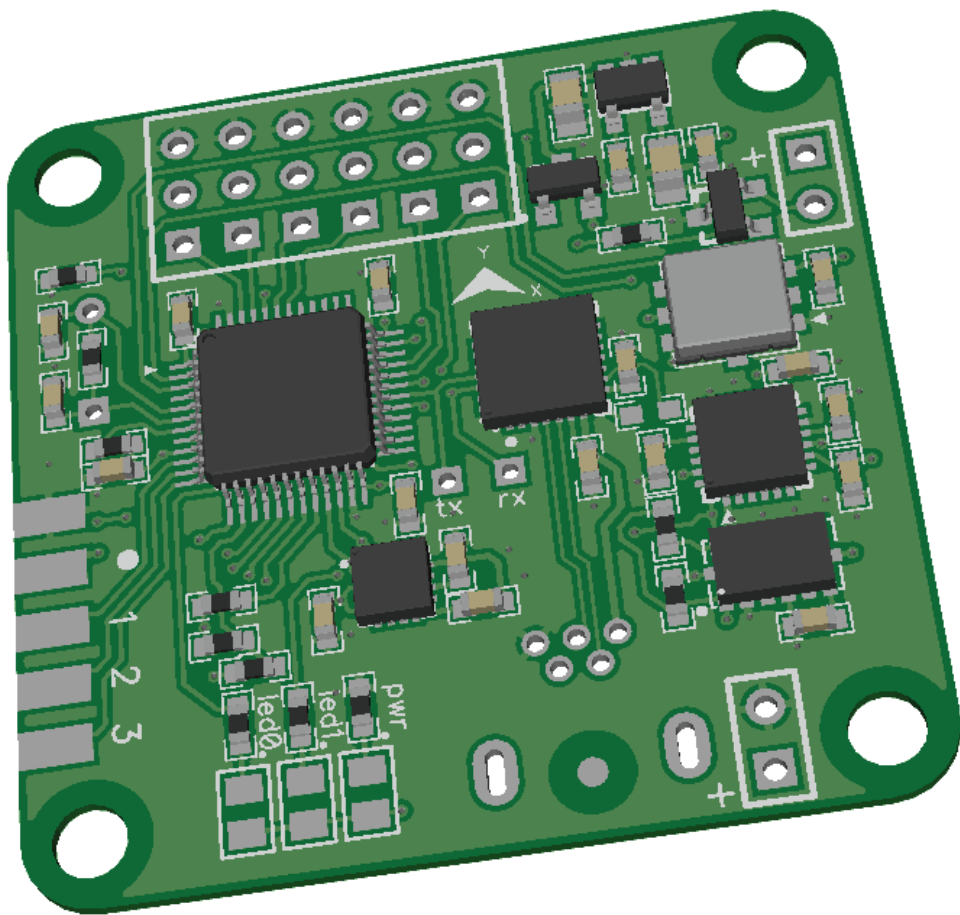


Flight Controller 「なぜ」 「Naze32」

Mini Fun Fly



Create by AbuseMark 2012

Mini FunFly Flight Controller “Naze32” (なぜ) Introduction

Mini Flight Controller 「なぜ」 (pronounced 「Naze」)

36x36mm.

6 grams (no headers, 8 grams with).

2000 degrees/second 3-axis MEMS gyro.

auto-level capable with 3-axis MEMS accelerometer.

3-axis magnetometer.

Pressure sensor.

Flexible motor outputs, support various airframe types -

Quad/Hexa/Tri/Bi/Y4/Y6/octo/camera gimbal.

Battery voltage monitoring and low voltage alarm.

Up to 8 ch RC input - supports standard receivers or CPPM receiver.

Modern 32-bit ARM processor running at 3.3V/72MHz.

Onboard USB for setup and configuration.

MultiWii-based configuration software for easy setup.

Designed for use with small indoor or small to mid-sized outdoor multirotor craft, or as a standalone camera stabilizer, the Naze32 flight control is simple to setup, with configuration based on the familiar “MultiWii” software.

WARNING Using this hardware with any DJI frames is not supported. This includes: FlameWheel 330, 450, 550, or any future or past frames released by DJI.

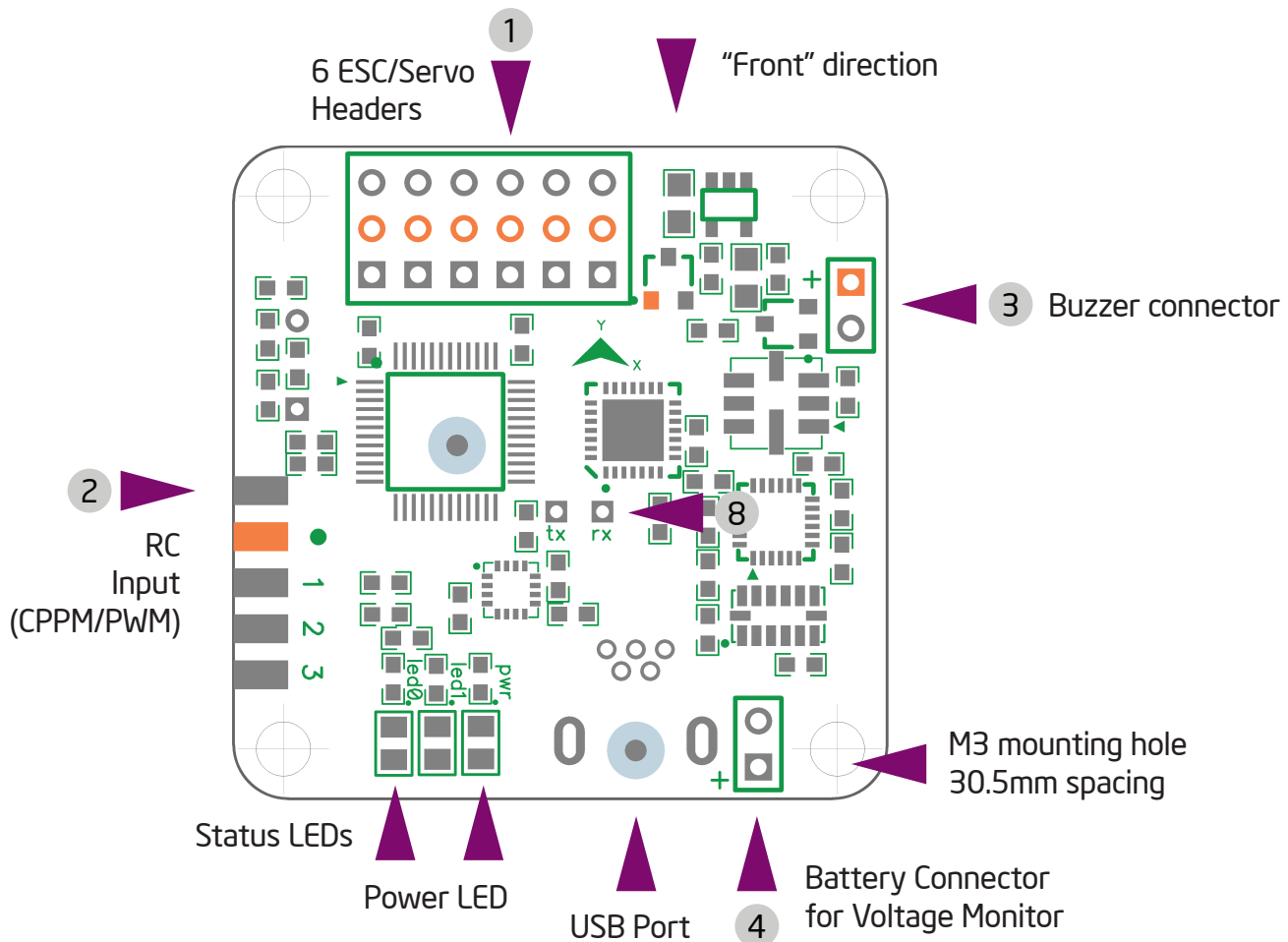
WARNING Do not use this hardware for serious business, commercial aerial photography, or anything else where its usage could endanger the end user, spectators, inanimate objects, aircraft flying overhead, etc.

NOTE While the flight controller firmware and configuration software is based on the MultiWii system, the processor is not Atmel AVR, and this hardware cannot be programmed through the Arduino development environment or any AVR development tools. For more information on STM32 development, see the following link:

<http://code.google.com/p/afrodevices/wiki/STM32Development>

NOTE This hardware is provided as-is and end-user is expected to have reasonable technical knowledge to complete set-up and reasonable R/C experience to operate multi-rotor aircraft.

Mini FunFly Flight Controller "Naze32" (なぜ) Hardware and Connection (Top)

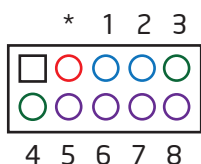


Board is 36x36mm square, with mounting holes for M3 screws, spaced 30.5mm.

[1] ESC / Servo Headers

6 motors or 4 motors and up to 2 servos can be connected here. The pins towards center of the board are signal pins. Following that is 5V, and then Ground/GND pins.
WARNING Incorrect or reverse connection to these connectors will instantly destroy the hardware.

[2] RC Input / Servo Output (CPPM / PWM) / GPS connector



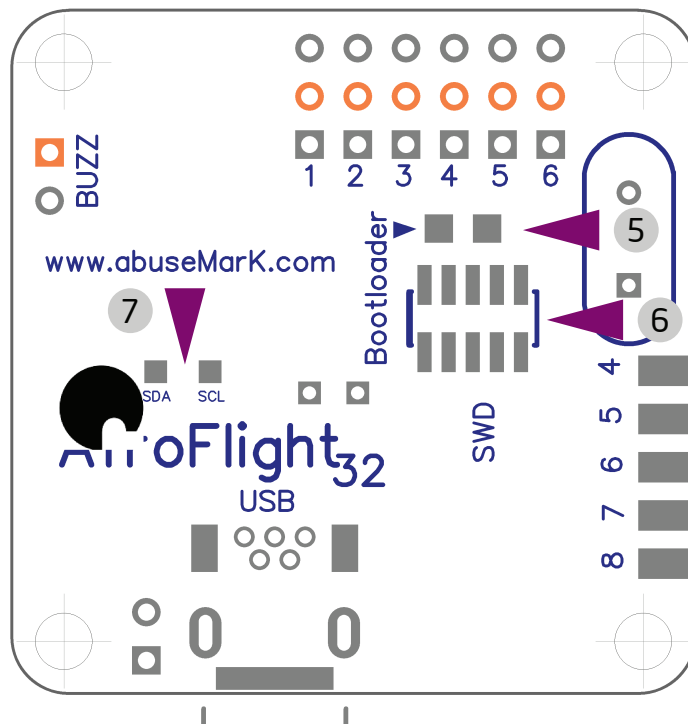
With the connector facing out, the pins are used as follows. Unmarked pin (left-most top side) is ground, pin marked with dot is 5V, the rest are RC signal pins for CPPM (channel 1) or channels 1 through 8. Standard male to male servo cable can be used to connect a CPPM receiver, then individual connectors going to 'signal' pins for any additional PWM channels.

In CPPM input mode, Channels 3 and 4 can be used for **3.3V** GPS connection. (3:TX, 4:RX), and Channels 5 to 8 can be used for motor outputs.

[3] Buzzer

5V Buzzer, connect a header here, + and - are marked and should be followed.

Mini FunFly Flight Controller "Naze32" (なぜ) Hardware and Connection (Bottom)



[4] Battery Voltage Monitor

Optionally connect + and - from battery pack here to be able to use in-flight battery voltage monitoring and alarm. Battery cell count is automatically detected, and cell warning voltage can be configured. For LiPo cells, up to 6S monitoring is supported. Buzzer will activate when voltage falls below configured per cell threshold.

[5] Bootloader pads

In case automatic firmware update fails or for upgrading from older firmware. Short pads together and connect power to the board. Firmware update tool can then be used to reflash.

[6] Cortex Debug Connector

10 pin 0.05" debug connector connected to 'SWD' port of STM32.
Pin 1 is bottom left.

[7] I2C Connector

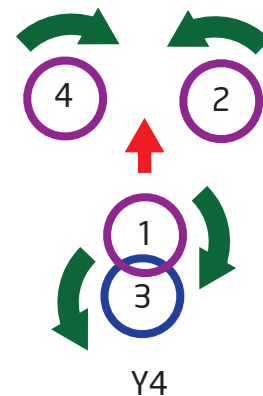
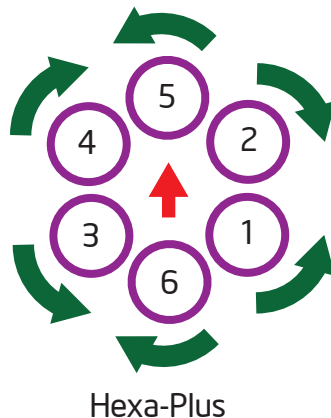
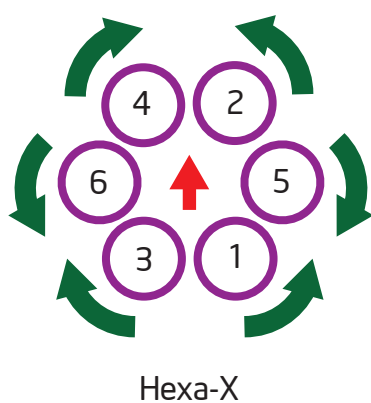
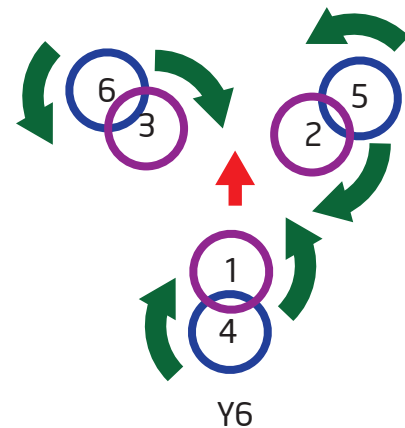
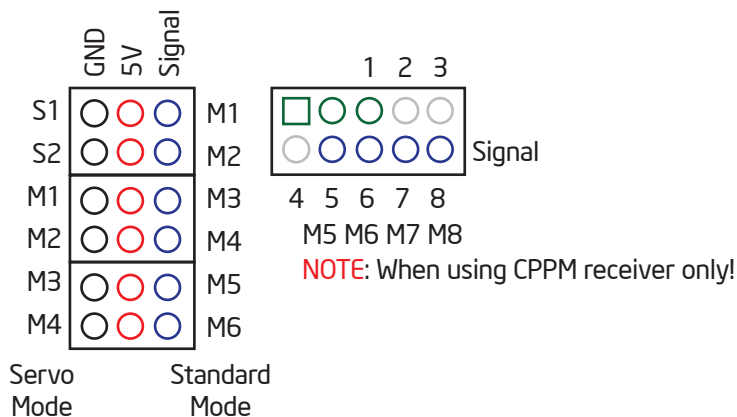
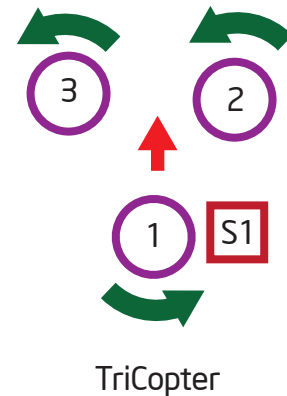
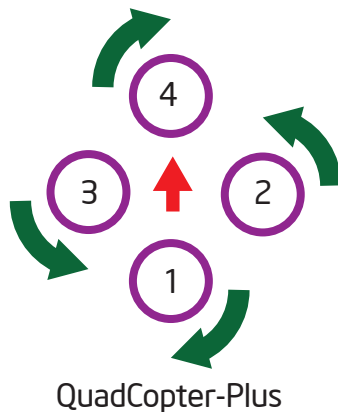
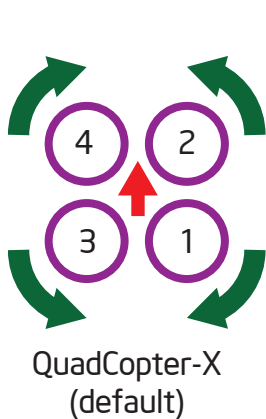
SDA and SCL from the sensor I2C bus are connected here. Used for debugging.

[8] Telemetry / Bluetooth / XBee serial port

RX and TX connectors from the USB Serial converter are available here. Can be used to connect Bluetooth / XBee / etc to allow changing settings without connecting USB. Cannot be used at same time with USB connection.

Mini FunFly Flight Controller "Naze32" (なぜ) Motor Mixer Settings

Images below show motor numbering and propeller rotation for various supported mixes. The number corresponds to the motor index (M1 to M8) from the diagrams below. Note, configurations with more than 6 motors require usage of CPPM receiver. In all cases, the "front" arrow on flight control board should point in same direction as red arrow in these illustrations. For Y4 and Y6 mixes, purple motors are top, and blue is bottom. In **servo mode** (Tricopter/Bicopter/Camera Stabilization), motor numbers change according to the diagrams below. When using CPPM receiver additional 4 motor outputs are available for Hexa + Camera, or Octo configuration.

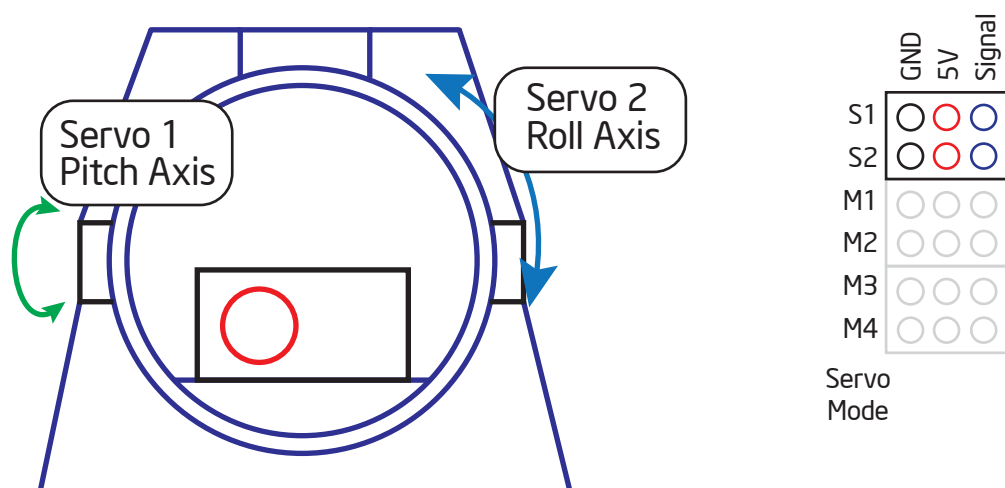


NOTE In Tri-Copter mode, tail tilt servo connects to S1, and motors M1..M3 as shown on the diagram above. When camera stabilization is enabled, gimbal pitch/roll servos connect to S1/S2, and motor connectors shift as well.

Mini FunFly Flight Controller “Naze32” (なぜ) Camera Stabilization and Gimbal Mode

Naze32 can be used as a standalone camera gimbal stabilization unit. By setting multirotor type to “Standalone Gimbal Stabilization”, servo outputs 1 and 2 can drive camera gimbal servos. Both analog (50Hz refresh) and digital (200Hz+ refresh) servos are supported. Servo gain (amount of servo travel for given angle of tilt) is configurable for both pitch and roll axis.

In addition to standalone stabilization, camera outputs can also be enabled when used as a standard multirotor controller. In case of regular (PWM) receiver, this limits to Quad mixer. In case of CPPM receiver, up to Octo can be supported while still allowing for camera control. Channels AUX3/AUX4 can be assigned to tilt/roll the camera mount in addition to stabilization.



WARNING If using high-current-draw servos for camera stabilization, consider powering them from a separate BEC. Only connect Signal and GND wire to the board, and 5V wire from servos to a separate power source.

Mini FunFly Flight Controller "Naze32" (なぜ) Firmware Upgrade / MultiWiiConf

To update firmware,

1. Download Flash Loader Demonstrator and latest baseflight firmware from <http://code.google.com/p/afrodevices/downloads/list>
2. Plug in USB or apply power.
3. Send 'R' (no enter/line feed needed) over serial port at 115200, 8n1.
4. Run Flash Loader, choose USB port, click next until asked for firmware hex file.
5. Upload.
6. When finished, unplug/replug USB or reconnect power.

If flash fails or sending 'R' does not enter bootloader mode, extended method is used:

1. Temporarily short bootloader pads (5) on the bottom of the board.
2. Plug in USB or apply power.
3. Follow instruction steps 4-6 above.

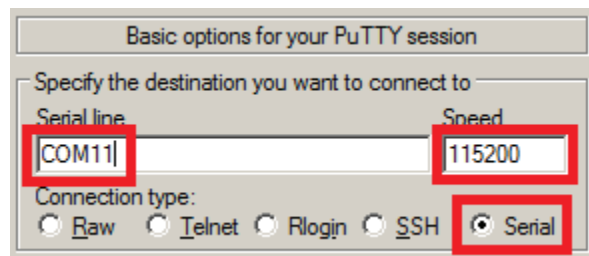
MultiWiiConf usage is standard, and all the basics are covered at MultiWii website, <http://www.multiwii.com/software>. There are no special requirements for this board, and default settings are sufficient for stable flight.

Mini FunFly Flight Controller "Naze32" (なぜ) Command Line Interface / configuration

Command Line Interface (hereafter CLI) is used to configure most functions of the board. A terminal program (such as PuTTY or HyperTerminal) should be used.

Download PuTTY from <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

Connect to USB Serial port at 115200 baud, 8bit, no parity, 1 stop bit. These are default settings for most terminal programs. To enter CLI, type '#' into the terminal.



A message will be printed,
Entering CLI Mode, type 'exit' to return, or 'help'

Command line is interactive, and most commands will print something in return.
Available commands:

help	print out a list of all commands with short description of each
defaults	reset all settings to built in defaults, and reboot
feature	enable, disable, or list enabled features (such as CPPM input, GPS, etc)
map	configure RC channel mapping for CPPM and regular receiver
mixer	set or show current multirotor mixer (such as Quad/Hexa/etc)
set	set or list available parameters. Many settings are available.
status	print out system status (voltage, uptime, enabled sensors, etc)
version	print out firmware version and build date/time
save/exit	save settings and reboot

WARNING Any changes to settings in cli, in particular related to motor output (throttle values, etc) are accepted REALTIME and may result in motors starting up without warning. Always configure with props disconnected or when model is only powered by USB.

Common usage examples:

To enable CPPM receiver, enable PPM feature:

```
# feature ppm
# save
```

To disable battery voltage monitoring, disable VBAT feature:

```
# feature -vbat
# save
```

To enable camera stabilization and change mixer to Quad-Plus:

```
# mixer quadp
# feature servo_tilt
# save
```

To configure CPPM receiver for EATR channel order and swap AUX1/2:

```
# map EATR2134
#save
```

Settings are applied real-time, however not saved until 'save' or 'exit' command is executed.