

WIP WIP WIP manual being revamped WIP WIP WIP

Open Flight Stabilizer (FlightStab)

updated 07/072013

rcgroups thread <http://www.rcgroups.com/forums/showthread.php?t=1794672>

source code <https://github.com/noobee/FlightStab>

build images <https://github.com/noobee/FlightStab/tree/master/Builds>

Key Features

- 3-axis gyro-based stabilization.
- WING mode mixing (single aileron and flaperons).
- DELTA mode mixing.
- VTAIL mode mixing.
- CPPM mode.
- throttle and flap channels pass through in CPPM.
- gyro-based attitude hold.
- configurable mixer limits to prevent servo binding.
- stick-based configuration for options.
- EEPROM storage for configurations (can be reset through jumpers on bootup)
- LED flashes for status and errors.
- RX calibration during startup to determine neutral stick positions.
- IMU calibration during startup to remove sensor offsets.
- AUX channel controls master gain linearly from 0% to 100%
- AUX channel also controls RATE and HOLD modes
- VRs on RX3S control axis gain and direction.
- DIP switches on RX3S control the mix mode.
- HOLD mode for attitude hold
- stick-controlled rotation rate in HOLD mode, optional in RATE mode
- inflight RX calibration to reset RX neutral values
- PID parameters adjustable through program box
- stick position controls gain, which prevents overcorrection or "fighting" commanded stick movement during manoeuvres.
- allows side mounting on flat fuselage foamies (by rolling 90° left or 90° right and then mounting to the fuselage side)
- optional AVRrootloader to allow firmware update without usbasp ISP.
- use pre-built firmware or build your own from the source code.
- support the AQUASTAR or DLUX programming box

note that some features are currently only in the dev/beta image.

Supported Devices

- RX3S V1
http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=25448

- RX3S V2 or V3 (they use the same firmware)
http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=28456
http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=31586
- NanoWii
- ITG-320x, MPU-6050 sensors
- Aquastar Programming Box
http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=25492
- DLUX Programming Box
http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=17619

Flashing the Firmware (for non Arduino-based boards)

- use an USBasp programmer capable of operating at 3.3v (highly recommended)
- search ebay for “**usbasp 3.3v**”. ensure that it supports both 5v or 3.3v to the target.
- RX3S require setting to 3.3v (via jumper setting on the USBasp). it is possible to use a 5v USBasp, but you need to power it differently.
- btw, i highly recommend this tool, especially if you have the RX3S V2 or V3
http://www.hobbyking.com/hobbyking/store/_27195_Atmel_Atmega_Socket_Firmware_Flashing_Tool.html

Flash using eXtreme Burner

Setting up eXtremeBurner (first time)

- install eXtreme Burner - AVR (<http://extremeelectronics.co.in/category/software/>)
- edit the properties of `C:\Program Files\eXtreme Burner - AVR\Data` to allow the user full control over the directory. otherwise, the contents are read-only.
- edit the file `C:\Program Files\eXtreme Burner - AVR\Data\chips.xml` to add support for the new **ATmega168PA** chip type.
- add the following XML section just after the ATmega168 section

```
<CHIP>
    <NAME>ATmega168PA</NAME>
    <FLASH>16384</FLASH>
    <EEPROM>512</EEPROM>
    <SIG>0x000B941E</SIG>
    <PAGE>128</PAGE>
    <LFUSE layout="2">YES</LFUSE>
    <HFUSE layout="3">YES</HFUSE>
    <EFUSE layout="2">YES</EFUSE>
    <LOCK>YES</LOCK>
    <CALIB>YES</CALIB>
    <PLACEMENT>.\Images\Placements\ZIF_DIP_40.bmp</PLACEMENT>
</CHIP>
```

- save the file. if there is an error saving, then check the directory access properties in the second step above again.

Checking and erasing the device before flashing for the first time

- run eXtreme Burner and select Chip type as ATmega168PA
- connect the ISP connector to the RX3S

- select **Read All**. the program should recognize the chip and proceed to read the flash, EEPROM and fuse/lock bits
- the flash and EEPROM will be read back as 0xff (since the lock bits LB1/2 are set). that is correct.
- the fuse/lock bits should read the following (the calibration value does not have to match):

```
lo=0xf7
hi=0xdf
ext=0xf9
lock=0xfc
```

- DO NOT PROCEED IF THE VALUES DO NOT MATCH OR IF EXTREME BURNER CANNOT RECOGNIZE THE CHIP (AFTER 2-3 TRIES)
- FROM THIS POINT ONWARDS, YOU WILL ERASE THE CHIP AND REFLASH WITH NEW FIRMWARE. THERE IS **NO WAY** TO RESTORE THE ORIGINAL FIRMWARE.
- select **Chip Erase**. the program will erase the chip and reset the lock bits.
- select **Read All** again, this time the lock fuse should read lock=0xff.

Flashing the device (each time you want to upgrade the firmware)

- open the firmware hex file
- either select the appropriate one from **.\\FlightStab\\Builds*.hex**, or
 - select the generated **FlightStab.cpp.hex** (if you built the firmware from sources, which is described later)
- select **Write** from the menu bar then **Flash**. the program will write the updated firmware to the flash. sometimes it will fail to recognize the chip ID. try again 2-3 times and it would usually succeed.
- disconnect the ISP connector from the RX3S.

Flash using avrdude

- avrdude recognizes the ATmega168PA with -p m168p
- please check the docs on how to read and verify the fuse bits, and how to erase the chip.
- apply the same caution in the previous section to verify the fuse/lock bits before erasing the chip.
- to flash the device, i used the following command (in a single line):

```
avrdude -C "C:\\Program Files\\arduino-1.0.3\\hardware\\tools\\avr\\etc\\avrdude.conf" -c usbasp -p m168p -U flash:w:<filename>.hex
```

Flash using avrdude on NanoWii (which has the arduino/leonardo bootloader)

```
avrdude -p m32u4 -P COM7 -c avr109 -b 1200  
TIMEOUT 3  
avrdude -p m32u4 -P COM5 -c avr109 -D -Uflash:w:%1:i  
TIMEOUT 3
```

Save above code to a Text file and rename something handy for you, example NanoFlash.bat. Open Explorer and type into address bar '%APPDATA%\Microsoft\Windows\SendTo' and put NanoFlash.bat in that folder. You'll need to swap details in Blue/Red to suit your FC.
How to find Trigger and Active Comports:

- Open device manager. Plug in NanoWii that port number will be Trigger Port. Replace Blue with this number.
- You'll manually reset the board while watching device manager. Will change to the Active Port number for few sec's. Change Red with this number.

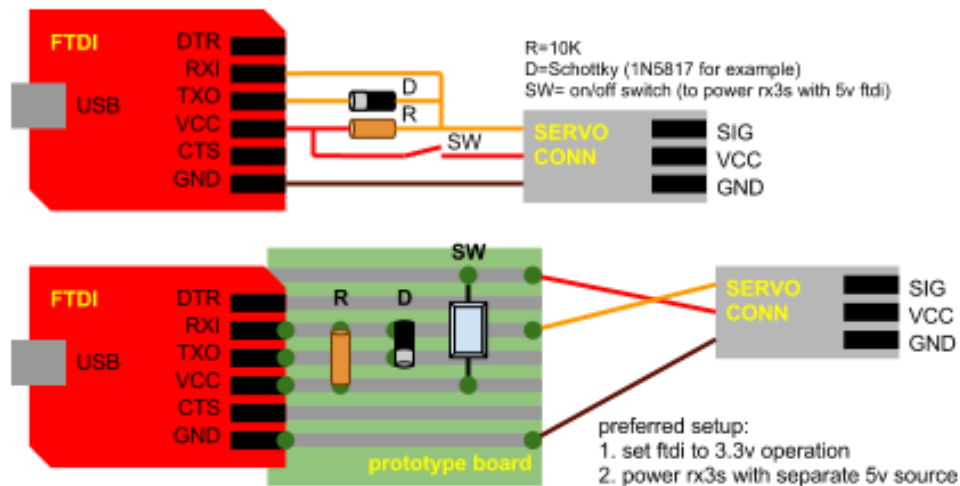
How to use

- Right click on *_NanoWii.hex file and go to option 'Send to' chose NanoFlash and your done.

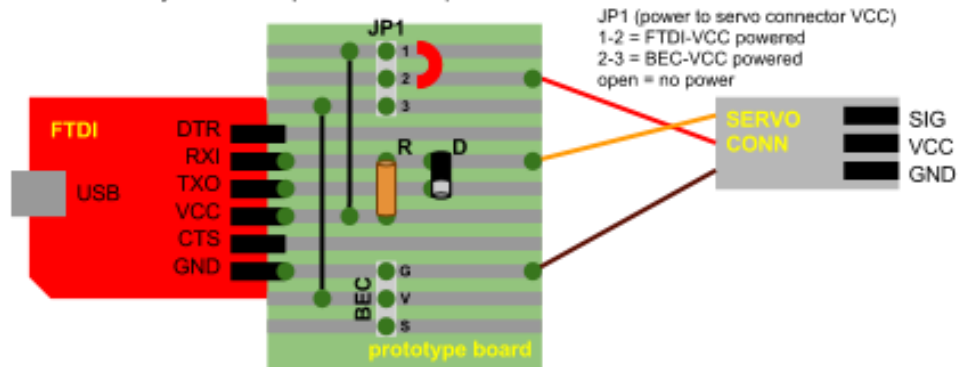
Optional: Flashing the AVRrootloader

It is possible to flash a bootloader via usbasp on the RX3S. If the bootloader is installed, then flashing the main stabilizer firmware does not require the usbasp. Instead, an FTDI USB-serial adapter with a small circuit mod connects to the **AILR_OUT** channel, which is more convenient than opening up the device to upgrade the firmware via the usbasp ISP connector. The circuit mod follows:

https://docs.google.com/drawings/d/1WhYAdw_9QpfYJWmOuu2lhCJNw6rNNagTw_OZEqrTo8/pub?w=867&h=439



Alternate layout with optional BEC power



Steps to flash the bootloader (note: for RX3S only):

- if flashing for the first time, follow earlier instructions to erase the chip and clear the lock bits.
- flash the hex file at
`.\FlightStab\AVRootloader\AVR\default\AVRootloader_AILR_OUT.hex`
- update the **EXTENDED FUSE** setting from **0xF9** to **0xFC**. This enables the atmega168pa to boot a 512-byte bootloader on reset.

Steps to flash the stabilizer image with a bootloader-enabled device:

- connect the servo connector from the FTDI + circuit mod to **AIL_OUT** channel
- run `.\FlightStab\AVRootloader\Windows\AVRootloader.exe`
- verify that baud rate is 115200 (optional, bootloader auto detects), “erase device during programming” is checked, “verify device after programming” is checked.
- click “**connect to device**”, which should establish a session with the device. If you have trouble connecting, try pressing “connect to device” then turn on the switch to power up and boot the device.
- select the flightstab hex file as the flash image file.
- click “**program**” to flash the stabilizer firmware. it should flash and verify the image, which completes in about 4 seconds at 115200 baud rate.

OPTIONAL: Building the firmware image from source code

Building firmware from sources is optional, you can also use prebuilt firmware in `.\FlightStab\Builds*.hex`

Building the firmware via arduino IDE

- install arduino v1.0x (<http://arduino.cc/en/Main/Software>)
- installing the i2cdevlib is optional unless you want to enable this library with `#define USE_I2CDEVLIB`. the default is `USE_I2CLIGHT`
 - download i2cdevlib (<http://www.i2cdevlib.com/usage>)
 - place i2cdev directories in `C:\users\<user>\My Documents\Arduino\Libraries`

```
C:\users\<user>\My Documents\Arduino\Libraries\_Stub
C:\users\<user>\My Documents\Arduino\Libraries\I2Cdev
C:\users\<user>\My Documents\Arduino\Libraries\ITG3200
...
```

- run `Arduino.exe` and select Board as ``Arduino Pro or Pro Mini (5V, 16MHz) w/ ATmega 168``
- load ``FlightStab.ino``
- uncomment ``#define RX3S_V1`` for RX3S V1 and check all other devices are commented out.
- verify/compile the program.
- locate the generated hex file at ``C:\users\<user>\AppData\Temp\build<numbers>.tmp\FlightStab.cpp.hex``. sort the directory by modified-date to help find the most recent ``build<numbers>.tmp`` entry.
- verify the timestamp of ``FlightStab.cpp.hex`` matches current time.

Building the Firmware via cmake (recommended)

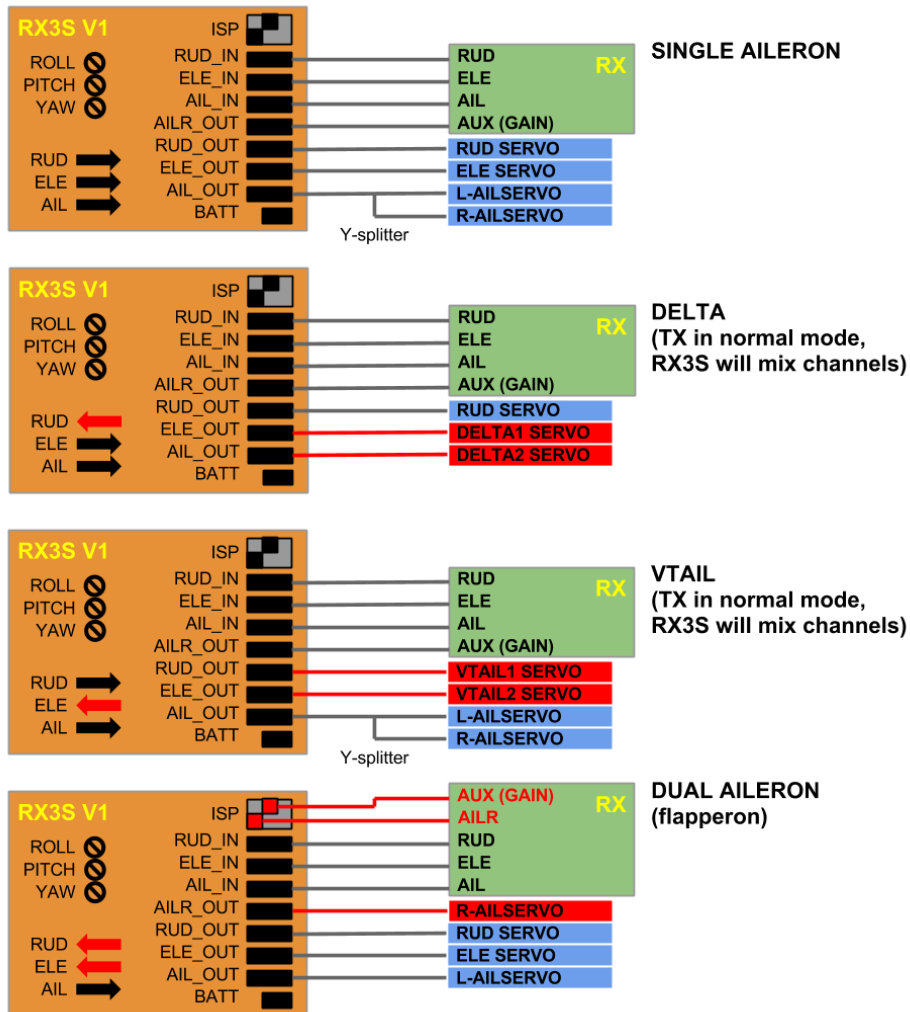
- install `C:\Program Files (x86)\CMake 2.8`
- ensure PATH includes `C:\Program Files (x86)\CMake 2.8\bin`
- install `C:\Program Files\arduino-1.0.3` (this exact path is important!)
- ensure PATH includes `C:\Program Files\arduino-1.0.3\hardware\tools\avr\utils\bin`
- run the following commands:

```
cd ./Builds
rd /s . (this optional step wipes out all intermediate files)
cmake [-DTODAY=yyyymmdd] -G"MSYS Makefiles" .. (-DTODAY is
optional to override the system date)
make [VERBOSE=1]
```

- a list of `.hex` files will be created in the current directory for all the different devices and with different compile-time options.

Connecting the RX3S V1

<https://docs.google.com/drawings/d/1UlpZlqXQydiNrwG9xPUMn7-c8fgz9dmPhzMvmBZxM3g/pub?w=1036&h=1138>



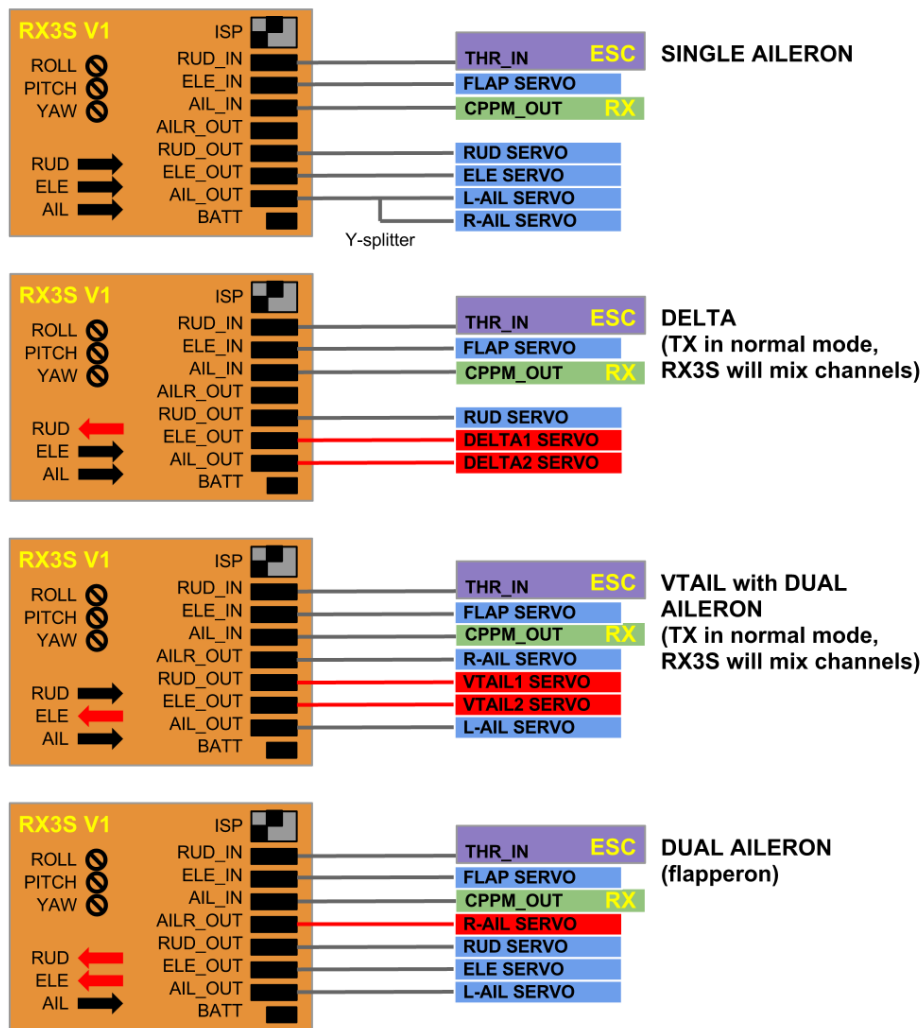
DUAL AILERON

- Only the signal wire on RX AUX and AILR channels should be connected to the specified pins on the ISP header..

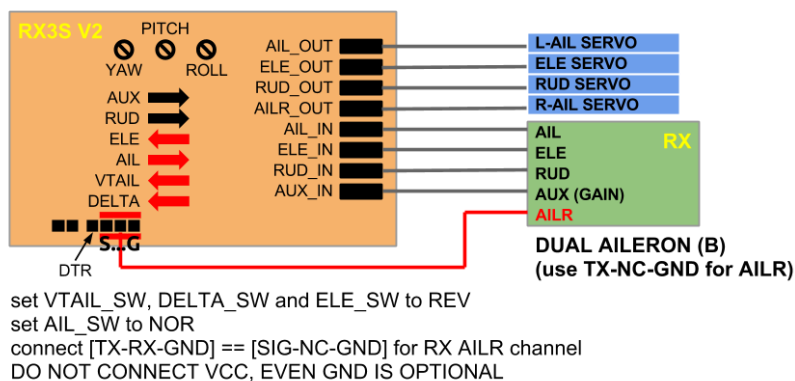
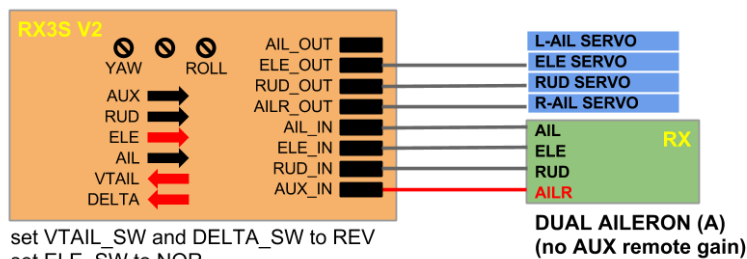
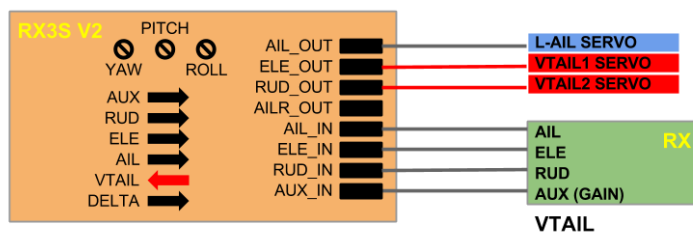
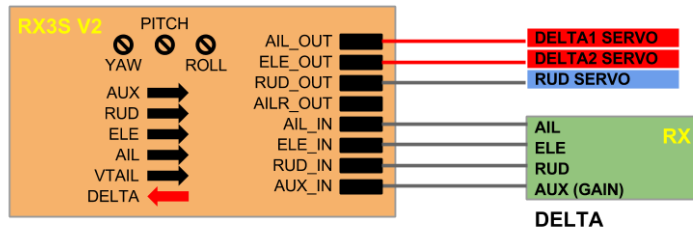
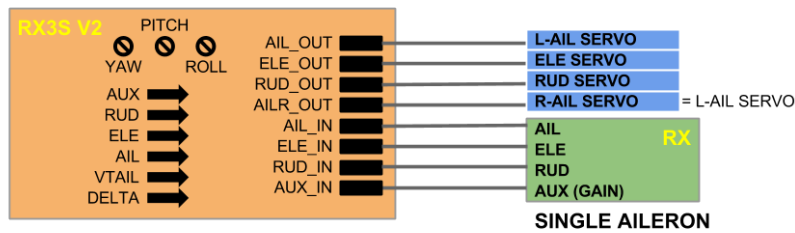
Connecting the RX3S V1 with CPM

https://docs.google.com/drawings/d/1OSFdJhFkyxzuxppBQuRJt8MLZkH4-uPX9_TsbjsJbFA/pub?w=1036&h=1138

CPPM ENABLED



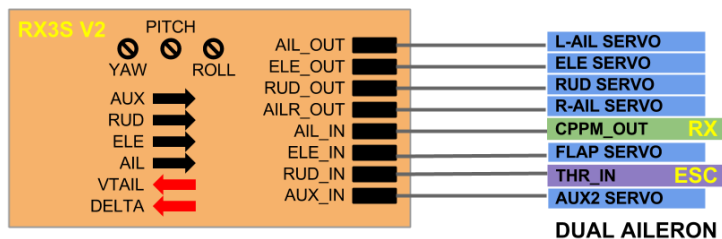
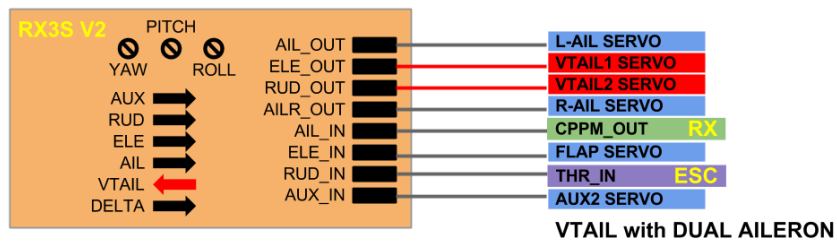
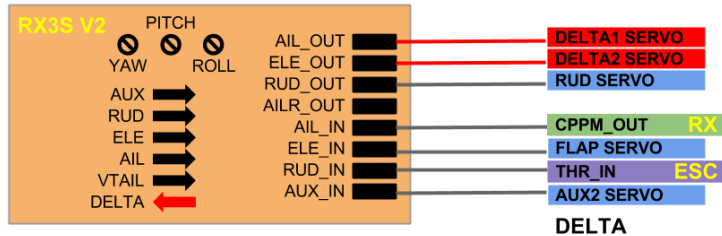
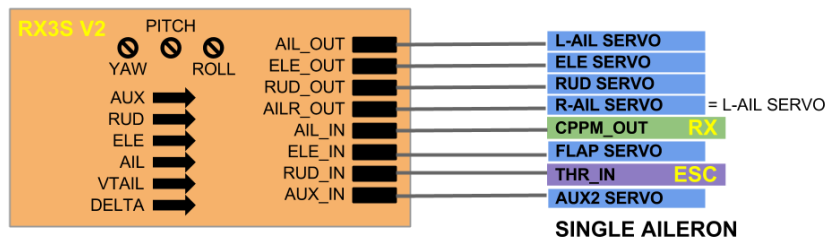
Connecting the RX3S V2



Connecting the RX3S V2 with CPM

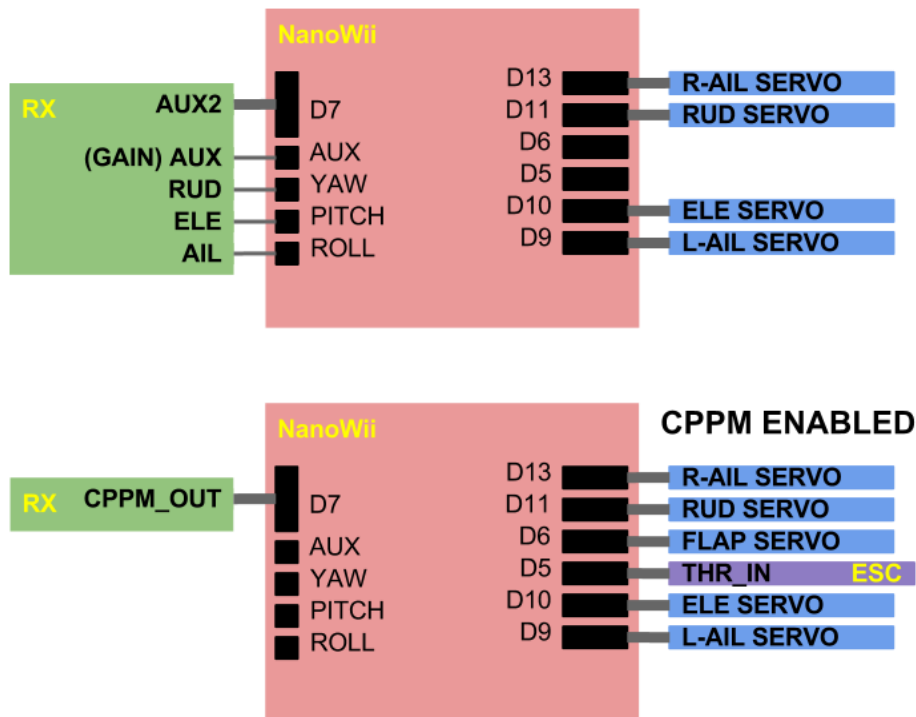
https://docs.google.com/drawings/d/17n4bOpNZL-B_cwbcNCIPpM4CLkEAdrxoobJhFI5-300/pub?w=1086&h=1272

CPPM ENABLED



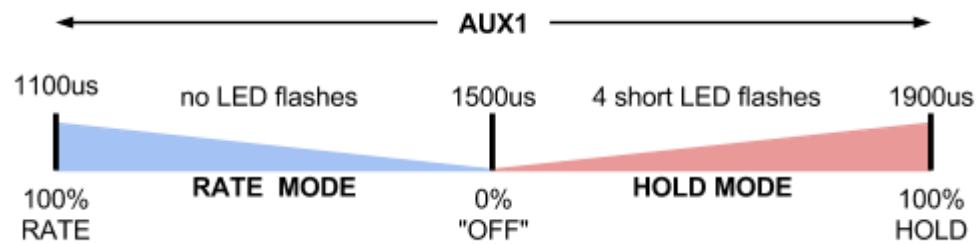
Connecting the NanoWii

https://docs.google.com/drawings/d/10f_17nVnECrE8oCDNfB_Dw8DAffwjyNOaXWew73Y3qs/pub?w=739&h=637

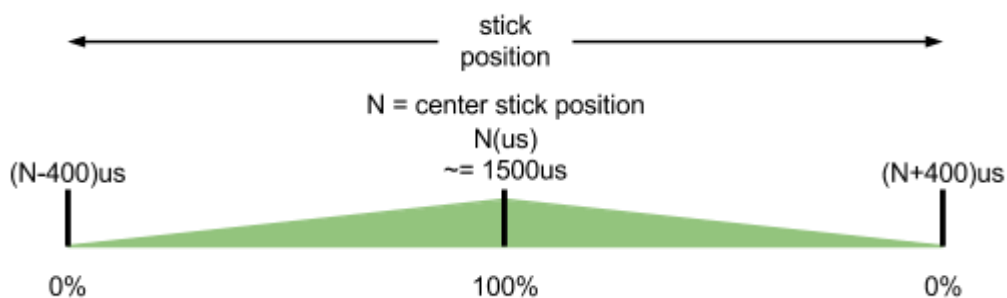


Basic Instructions

- connect the device based on the connection diagrams. for RX3S, the DIP switches set the WING mode (single aileron, dual ailerons or flapperons, delta or vtail). for NanoWii, the stick configs will set WING mode.
- if you change any DIP switch settings, power cycle the device to enable the new setting to take effect.
- the VR gain POTs on the RX3S control the correction gain and direction for each of the pitch (ELE), roll (AIL) and yaw (RUD) axis.
 - 7 o'clock = max gain in one direction
 - 12 o'clock = zero gain
 - 5 o'clock = max gain in opposite direction
- the AUX channel controls the master gain linearly. It can be set with a knob/slider (ideally) or a 2/3-position switch. It can also be left unconnected, which defaults to MAX gain in RATE mode.
- the AUX channel also controls RATE mode vs HOLD mode. The following diagram shows the AUX1 pulse width determining RATE and HOLD master gain values. To calibrate the TX AUX1 for 1500us, apply subtrim to AUX1 (with 3P switch in middle position for example) until you are at the point between 4 flashes and no flashes, that is effectively zero stabilization.



- the stick position also controls the gain to reduce the stabilizer from over correcting your controls during manoeuvres.
 - neutral position = max gain
 - extreme position = zero gain
 - in between = linear between max and zero.



LED Status

- there are 4 LED message slots. each slot is a series of flashes to indicate a condition. LONG flashes are 600ms pulses (600ms on and 600ms off), SHORT flashes are 200ms pulses and VERY_SHORT pulses are 30ms pulses. the number of pulses and duration of each pulse determines the message.

LED pulses	Message
2 LONG	WING mode = single aileron
3 LONG	WING mode = delta
4 LONG	WING mode = vtail
5 LONG	WING mode = dual ailerons (flapperons)
1 SHORT	RX calibrating
2 SHORT	IMU calibrating
3 SHORT	both RX and IMU calibrating
4 SHORT	HOLD stabilization mode (otherwise RATE stabilization mode)
5 SHORT	device init error (gyro)
20 VERY SHORT	low SRAM

50 VERY SHORT	EEPROM has reset, power cycle device now
---------------	--

RX and IMU Calibration on Startup

- after powering up, the device will try to calibrate the RX input and IMU sensor simultaneously.
- during RX calibration, keep the AIL/ELE/RUD sticks in the neutral (centered) position. for flapperons, it does not matter if they are in the UP or DOWN position. once the device has detected that the sticks have been still for a period, it will record those readings as the neutral stick positions.
- during IMU calibration, keep the plane still on the ground. **it does not have to be wings level, just have to be still.** once the device has detected that the IMU readings are stable, it will record those readings as the “zero” rotation rate values.
- the LED will flash accordingly if the RX or IMU (or both) calibration is still in progress.
- Stabilization will engage once both RX and IMU calibrations complete. You can tell it is the case by any one of the following:
 - waiting for the control surfaces to wiggle back and forth 3 times
 - looking at the LEDs (should not see calibration status)
 - rocking the plane to see if there is correction (with proper gain setting)
- If there is a power cycle in the air, the device will undergo calibration again. But it is unlikely for the plane or the TX to be still enough for calibration complete in the air successfully. However, the RX signals still pass through to the servos, allowing you to control and land the airplane, but without any stabilization.

Mixer EPA Mode

- there are 3 mixer EPA modes that controls the limits of all the servos.

Mixer EPA mode	Notes
FULL (default)	1000-2000 us
NORMAL	1100-1900 us
TRACKING	start with 1250-1750us, then track and never exceed the RX

- with **tracking** mode, the servos will never be driven past the point that the RX would drive if there was no stabilization correction. this would prevent servo binding. however, this means that you should “cycle” the sticks to the limits on the ground each time before flying for the device to learn the limits from the RX. otherwise, it would apply a smaller correction than it could based only what it learnt from the RX.
- the mixer EPA mode can be set from stick configuration.

CPPM Mode

- The device supports CPPM mode for reduced connections to the RX and to enable more channels (AUX2, THROTTLE and FLAP) on the RX3S V1 and V2 devices. The RX3S V3 device has an integrated RX, which cannot be set to CPPM mode.

- Refer to the connection diagram for CPPM connection. note that the THROTTLE and FLAP channels are passed through and available through two pins since the RX is unlikely to output CPPM and individual channels at the same time.
- currently, only the “RETA1a2F” CPPM channel order is supported.

Attitude Hold

- Attitude HOLD allows the plane to fly with a fixed attitude (as long as there is sufficient power and control surface authority).
- The AUX1 channel determines if the plane is in RATE (stabilization only) mode or HOLD mode.
- ~~When attitude hold is enabled, the device will~~**engage**~~, or hold the attitude, whenever the sticks are near their neutral position. Attitude hold will~~**disengage** ~~when the stick moves away from the neutral position.~~
- ~~Pitch and roll are separately controlled from yaw. For example, the device can hold a high alpha attitude (with pitch and roll) while you use the rudder to steer the aircraft. For another example, the device can hold a knife edge (with yaw) while you use the aileron and elevator stick to steer the aircraft.~~
- Attitude hold is short term as the gyros drift after a brief period. When you move the sticks to correct for the drift, the device will disengage the hold and then re-engage when you return the sticks to the neutral position.

Side Mounting

- The device supports mounting to the side of a flat fuselage foamy. Assuming you start from a normal flat orientation, you can either roll the device 90 degrees to the left and mount it to the left side of the fuselage or 90 degrees to the right and mount it to the right side of the fuselage. in this way there is no need to rearrange all the RX and servo channel wires.
- Please see the stick config section for updating the device mounting orientation and storing it into the EEPROM.

Stick Configuration Mode

- To enter stick configuration mode, move the AIL/ELE stick to **7-9-7-9-7** positions **within 15 seconds** after powering up the device (assuming mode 2 TX).
- The position numbers correspond to the telephone pad. 1 = top left (roll left nose down), 9 = bottom right (roll right nose up). **Start from position 7 (roll left nose up):**

1	2	3
4	5	6
7	8	9

- Note that the device assumes that the stick will **visit the bottom left corner (position 7) first** and will calibrate its orientation from that reference point. you could also enter stick configuration with 9-7-9-7-9, 1-3-1-3-1 or 3-1-3-1-3 stick transitions but the up/down and/or left/right directions will be reversed while in configuration mode.

- Once in configuration mode, the ELE servo toggles to indicate the config item and the AIL servo toggles to indicate the config value:

Config Item	ELE servo toggles	AIL servo toggles	Notes
WING mode	1	1 = single aileron [default] 2 = delta 3 = vtail 4 = dual ailerons (flapperons)	NanoWii only
Roll Gain	2	-4 to 0 to +4 [+4 default]	NanoWii only
Pitch Gain	3	-4 to 0 to +4 [+4 default]	NanoWii only
Yaw Gain	4	-4 to 0 to +4 [+4 default]	NanoWii only
Mixer EPA mode	5	1 = full (1000-2000)us [default] 2 = normal (1100-1900)us 3 = tracking the RX	
CPPM Mode	6	1 = No CPPM [default] 2 = RETA1a2F order	
Mount Orientation	7	1 = Normal (flat) mounting 2 = Roll 90° left and mount on left side 3 = Roll 90° right and mount on right side	
Exit	8	1 = no-op 2 = write EEPROM and reboot	

- Move the AIL stick **left or right** to change the config item and the ELE stick **up or down** to change the config value.
- WING mode and Roll/Pitch/Yaw gains are **ignored** on the RX3S since they use the DIP switches and VR POTs for WING mode and axes gains, respectively.
- For the Pitch/Roll/Yaw gain, the AIL servo toggles one way to indicate positive gain and the opposite way to indicate negative gain. It does not toggle when the gain is zero. Try it to get a better idea.
- Selecting the last config item (Exit) and then changing the config value from 1 to 2 will write the updated config to the EEPROM and reboot the device. This is the only way to exit the stick configuration mode (other than power cycling the device).

Resetting the EEPROM

- To reset the EEPROM to default values, use a jumper to short the pin pairs for the device and then power up.

Device	Pin pair to apply jumper
RX3S	AIL_OUT and ELE_OUT
NanoWii	D6 and D5 (in the Motor Output group)

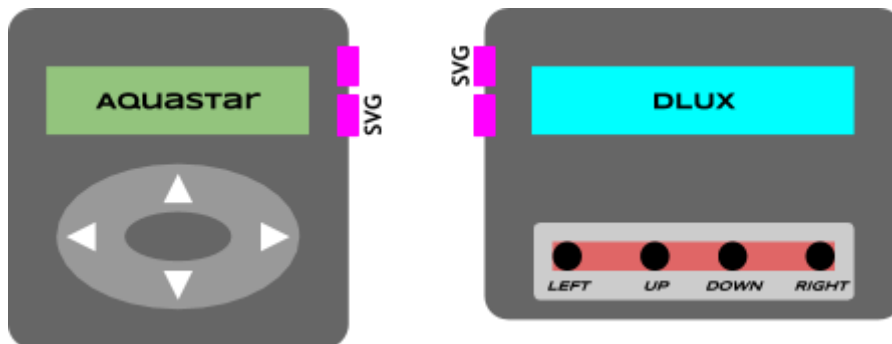
- On reboot, the device will clear the EEPROM and flash the LED rapidly in 3 sec bursts, which indicates that you can now power off the device and remove the jumper. On the next power up, the EEPROM should start with the default “factory reset” values.
- Normally, the plane will wiggle the surfaces 3 times to indicate that calibration has completed. If the EEPROM was reset to factory default (either on first boot or on jumper reset), the plane will wiggle the surfaces 9 times instead. **This can be used to indicate a problem with EEPROM corruption when you do not expect the EEPROM to be reset.**

Programming Box Support

FlightStab supports configuration through the Aquastar or DLUX programming boxes. Using the programming box is more convenient and will allow more options than using the stick configuration.

The steps to enable programming box support are:

- Use the USBASP ISP programmer to erase the chip, update fuse settings and flash the **AVRootloader_AQUASTAR.hex** bootloader
- Use the AVRootloader to update the main programming box firmware:
 - **2013xxxx_AQUASTAR.hex** for the AQUASTAR box
 - **2013xxxx_DLUX.hex** for the DLUX box
- Connect the programming box to the FlightStab device through the “one-wire” servo connector and power them up.



Flashing the Aquastar/DLUX with AVRootloader

On the Aquastar, the CPU is exposed, so it is highly recommended to use the HK chip adapter for the ISP cable to the USBasp. The ISP pads are also exposed on the PCB if you want to solder the signals instead.

On the DLUX, the CPU is hidden behind the LCD board, but the pads are exposed just above the 3rd and 4th buttons. The ISP pads have the following definitions (from left to right):

GND	29	VCC	17	16	15
GND	RESET	VCC	SCK	MISO	MOSI

After connecting the USBasp programmer to the program box via the ISP connector, the next step is to erase the chip and set the fuse for bootloader support.

- Select the target chip type as **ATMega8A** chip (and not ATMega168pa) on the programmer (eXtremeburner, AVRDude, etc).
- Read the fuse settings. The following table shows the expected fuse values. The calibration value is not important (and should not be changed):

Fuse	Aquastar	DLUX
LOW	0xAE	0xAE
HIGH	0xCF	0xCF
LOCK	0xFF	0xFC

On the Aquastar, the LOCK fuses are **NOT** set (0xFF), so you can read and save the flash image at this time in case you want to restore the factory functionality. On the DLUX, the LOCK fuses **ARE** set (0xFC), so you can only erase the chip to proceed.

DO NOT PROCEED if the values are unexpected.

- Erase the chip. This should also reset the LOCK fuse for the DLUX to 0xFF.
- Change the HIGH fuse from 0xCF to **0xCC**. This will enable the chip to boot a 512-byte (256-word) bootloader. Remember to restore the HIGH fuse back to 0xCF if you rolled back to the factory image on the Aquastar.
- Flash the AVRRootloader bootloader. Both Aquastar and DLUX use the same bootloader hex file at
.\AVRRootloader\AVR\default\AVRRootloader_AQUASTAR.hex
- Close up the box.

Use the AVRRootloader instructions to flash the main programming box firmware image (including using the “one-wire” serial connection). The Aquastar firmware is at **.\Builds\20130505_AQUASTAR.hex** and the DLUX firmware is at **.\Builds\20130505_DLUX.hex** (dated 20130505 or later).

Using the Aquastar/DLUX Programming Box

Connect the programming box to the FlightStab device. The RX3S devices uses AILR_OUT as the default channel. Power them on at the same time (or the RX3S after the program box). If the device recognizes the program box, the program box will display the recognized device ID on the status page. Press left/right to change pages and up/down to change sub-options within the page.

The status page (first page) has several items. The device ID, device version, and EEPROM statistics (1/2/R). 1 means the number of times the device detected an error and reset copy #1 of the config. 2 is the same for the redundant copy. R is the number of times both copies have been reset, usually as an outcome of the jumper-based EEPROM reset procedure or by choosing “erase cfg” through the program box.

The eeprom page (last page) has several actions. Update cfg justs updates the Flighstab device with the config that you modified with the program box. Erase cfg invalidates the

config so that the device will default to “factory settings” on the next restart, as if this was the first time it was booting up. Erase stats erases the 1/2/R stats.

20130702 dev onwards

<https://github.com/noobee/FlightStab/tree/newcfg/Builds>

20130702_RX3S_V1|2.hex

		stick config ail/ele toggles
WING_MODE	set WING mix mode	1 ail
USE_DIPSW *	based on DIP switches	1 ele
SINGLE_AIL	override to SINGLE AIL	2 ele
DELTA	override to DELTA	3 ele
V-TAIL	override to V-TAIL	4 ele
DUAL_AIL	override to DUAL AIL	5 ele
MIXER_EPA MODE	limit servo correction output to	2 ail
FULL *	1000-2000 us	1 ele
NORM	1100-1900 us	2 ele
TRACK	never exceed RX input	3 ele
CPPM_MODE	allow combined PPM input from RX	3 ail
NONE *	no CPPM, use separate RX (PWM) channels	1 ele
RETA1a2F	use CPPM with RETA1a2F	2 ele
TAER1a2F	use CPPM with TAER1a2F	3 ele
AETR1a2F	use CPPM with AETR1a2F	4 ele
MOUNT_ORIENT	support mounting sideways on flat fuselage	4 ail
NORMAL *	mount device flat “normally”	1 ele
ROLL_90_LEFT	roll device 90 deg left and mount on left side	2 ele
ROLL_90_RIGHT	roll device 90 deg right and mount on right side	3 ele
STICK_GAIN THROW	stick-position gain blends from max (1.0) to zero over	5 ail
FULL *	full stick range	1 ele
HALF	half stick range (ie. no correction from ½ stick onwards)	2 ele
QUARTER	quarter stick range	3 ele
MAX STICK_ROTATE	set stick-controlled rotation rate at full stick to	6 ail
49	49 deg/s	1 ele

98	98 deg/s	2 ele
196	196 deg/s	3 ele
391 *	391 deg/s	4 ele
782	782 deg/s	5 ele
RATE STICK_ROTATE	allow stick-controlled rotation rate in RATE mode	7 ail
DISABLE *	disabled	1 ele
ENABLE	enabled	2 ele
INFLIGHT CALIBRATE	toggle RATE/HOLD 3x within 0.5s to calibrate	8 ail
DISABLE	disable inflight RX calibration	1 ele
ENABLE *	enable inflight RX calibration	2 ele
VR_GAIN		TBD
AIL/ELE/RUD	POT = use device POT setting * -127 to 127 = override POT setting	
RATE_PID	P, I, D parameters (500, 0, 500) * for	NA
Ail P/I/D	ROLL axis P/I/D in RATE mode	
Ele P/I/D	PITCH axis P/I/D in RATE mode	
Rud P/I/D	YAW axis P/I/D in RATE mode	
HOLD_PID	P, I, D parameters (500, 500, 500) * for	NA
Ail P/I/D	ROLL axis P/I/D in HOLD mode	
Ele P/I/D	PITCH axis P/I/D in HOLD mode	
Rud P/I/D	YAW axis P/I/D in HOLD mode	
EEPROM	EEPROM Action	9 ail
Update Cfg	Write Config to EEPROM	2 ele
Erase Cfg	Erase Config in EEPROM to Default	NA
Erase Stats	Erase 1/2/R Stats in EEPROM to 0/0/0	NA