Open Flight Stabilizer Version 3.1 (FlightStab)

Updated January 20, 2018

This document is derived from the original document published by noobee along with input from RCGroups ckleanth and is currently maintained by JohnRB.

This document is for builds compiled 03/22/2014 and latter.

Attention - Users of Attitude Hold Mode

A user reported a problem where occasionally when Attitude Hold mode was enabled, the surfaces would immediately move to their limits. He saw this problem both on an RX3S-V3 and an RX3SM. The problem was eventually traced to slow RX connection time to the TX data stream and the fact that Attitude Hold was enabled when the Bind to the RX was performed. This problem is not unique to OrangeRX devices and could happen with any brand of RX that supports fail safe and takes a long time to connect to the TX data stream.

OFS was ready to operate (Wag's completed) before the RX connected to the TX. This means that OFS used the fail safe values sent by the receiver while waiting for TX connection instead of the actual current TX control positions. When the actual TX data started to be emitted by the RX, OFS thought that the aircraft position had changed and was attempting to make corrections. An Inflight Calibration at this point corrects the problem but it could be far too late to prevent a crash.

Make sure when you bind your TX to the RX that Attitude Hold is not active and that your sticks are in the desired position along with any trim. If you are using any of the mixing modes like Delta, V-Tail or Duck, Bind the RX again after any major trim changes.

Open FlightStab News

January 20, 2018
- Documentation only, format corrections in Program Box Options and Stick Config Configuration Chart.

October 30, 2017
- Documentation only, update CFG version number in Program Box Options and Stick Config Configuration Chart.

August 23, 2017
- The author of the SRXLH and SRXLM support has discovered an intermittent bug that only occurs during power on. The problem causes all of the outputs to be forced to center channel except for throttle which is held at minimum. The Flight Controller will not respond to any inputs from the transmitter. All users of SRXLH and/or SRXLM are encouraged to apply this new firmware.
- All other versions of the firmware have also been updated without any changes made to the source code so there is no need to update if you are not using SRXLH or SRXLM firmware versions.

April 5, 2016
- Updates to RX3SM DSMx Configuration Chart.

February 28, 2016
- Stick Configuration does not allow selection of MAX MIXER_EPA mode
- Update only required for users of Stick Configuration.

February 9, 2016
- Problem discovered with the nanoWii and Flip 1.5 driving the THR signal line positive during the OFS initialization code when operating in one of the serial modes (CPPM, SBUS etc.).
- There is no need to update if you are not using these

Switch Settings.

September 19, 2015

- Documentation change Only. FrSky Taranis Inflight Calibration sequences added.

August 21, 2015

- August 18,2015 had a bug in Delta and V-Tail mode caused by incorrect subtraction order which caused axis to be swapped. This release just corrects that problem.

August 18, 2015

- Fixed Inflight Calibration with Delta and V-Tail mixes. Sub Trim and Trim can now be used on the individual servos and OFS will account for the change with Inflight Calibration of at power on (all sticks must be centered and aircraft must remain still during power on. Only small amounts of Sub Trim or Trim should be used as the servo travel is limited by OFS based on the setting of MIXER EPA MODE.
- Two new mounting modes have been added, Rotate 90 degrees left and Rotate 90 degrees right.
- Programming Box firmware **must** be updated to support this and future versions.

August 1, 2015

- The **only** change in this release is to add Dual Aileron support for the RX3SM DSMx device. The RX3SM DSM2 device **cannot** support dual ailerons due to lack of connectors.
- The required rework for the RX3S DSMx is not major, however the parts are extremely small so SMD rework skills and tools are required.

April 14, 2015

- Force MPU-6050 gyro (Mini MWC, nanoWii & Flip 1.5) to use 5Hz Digital Low Pass Filter as default. The other devices already default to 5Hz
- Added code to Programming Box to allow setting the gyro Digital Low Pass Filter bandpass if desired.
- Programming Box firmware **must** be updated to support this and future versions.
- EEPROM version updated to 11.

February 21, 2015

- Add AUX2 output on Flip 1.5 board.
- Allow Master Gain to have values less than 25, previous releases went from 0 immediately to 26 or higher.

February 07, 2015

- Change made on 2015-02-02 moved zero gain as well as Rate/Hold hysteresis to 1550us +/- 25us. Restore hysteresis for gain back to 1500us +/- 25us.

February 03, 2015

- When using RUD 2-ELE 2-AIL wing mode, MIXER_EPA settings were not being properly applied to ELE-R. Correct bug in FlightStab.ino
- Only users of RUD 2-ELE 2-AIL wing mode require this update.
- If you have not already updated Programming Boxes to the **20150130 version**, they will require updating as well.

February 02, 2015

- Rate/Hold mode hysteresis region was straddling the 0 gain range (1500us +/- 25us). This was causing a problem resetting Hold mode corrections. When Hold mode was used and gain reduced to 0 Hold mode was still engaged. When Hold was enabled again there might have been corrections left over from the previous use. The hysteresis region has been moved to 1550us +/- 25us so that then when the gain is set to 0 (1500us - 1525us) Rate mode will be engaged forcing a reset to the Hold mode corrections.
- All users should update to this version of the firmware.
- If you have not already updated Programming Boxes to the **20150130 version**, they will require updating as well.

January 30, 2015

- Dual Aileron with Dual Elevator Support
  - Channel input named E is Elevator left
  - Channel input named F is Elevator right (If Flaps are required Channel name 2 must be used or connector directly to desired channel at the RX).
- See Configuration/Pinout Tables for supported devices
  - Elevator servos must be installed such that the stabilizer correction is applied in the proper direction on both surfaces. There is no mechanism available to the user to alter the correction direction independently - Pot setting or EEPROM value is applied to both

- Some new features have been added to Open FlightStab that are described below. If you do not require these new features then you may continue to use the builds from September 1, 2014. This release does not include any bug fixes.

- The latest version of Open FlightStab is always available at this Link.

Changes since the September 1, 2014 release

- A new #define (MAX_TRAVEL) has been added to FlightStab.ino. Including this #define in a build will increase the servo limits from 1000us-2000us to 900us-2100us. This may be required if you have servos that do center at 1500us-1520us.
  - This build only includes one Hex file using this feature for an S.BUS version of the Flip 1.5.
- User ingo_s on RCGroups has contributed firmware to support SRXL protocol used by Multiplex (12 and 16 channel) and Graupner HoTT. The Hex files that include SRLX in the name provide this support. If you plan to use any of the SRXL files not listed below, please do through ground testing before attempting a flight - they have **not** been tested..
  - The only Hex files that have been tested are:
    - 20141216_RX3S_V2_SRXLH.hex (HoTT)
      - Tested with GR-12 HoTT Receiver Link
    - 20141216_RX3S_V2_SRXLM.hex (Multiplex)
      - Tested with Receiver RX-4/16 FLEXX M-LINK ID Link
      - Tested with Receiver RX-7-DR light M-LINK 2.4 GHz Link

Table of Contents

Quick Links to Discussion and Files

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

Files
The links below are to the latest version of Open FlightStab. If you wish to obtain Hex files for older version, use the build images link. Once that page opens, look near the right side of the page and you will find the text **History**. Click on **History** and you will be presented with a list of all of the commits since the very beginning. Search the list for the date of interest and then click on **browse code** and you will be presented with the Hex files altered on that date.

noobee's Master branch release (last update March 22, 2014):
source code https://github.com/noobee/FlightStab/
build images https://github.com/noobee/FlightStab/tree/master/Builds
Download the whole repository as zip Noobee FlightStab Master zip

JohnRB's Master branch release (updated clone of noobee's):
source code https://github.com/John-RB/FlightStab
build images https://github.com/John-RB/FlightStab/tree/master/Builds
Download the whole repository as zip John-RB FlightStab Master zip

Notes:
noobee has been extremely busy doing his "real job". The most recent updates are at the JohnRB link. If and when noobee has some free time, the JohnRB version will be merged back into the noobee Master Branch. The master branch always is the Stable version of Open Flight Stabilizer. Any new mixes will appear as a branch and will be merged once the branch has matured and verified by flight tests.

There are a number of changes and support for new boards like the FLIP 1.5.
The pinout could have changed for some boards, so please check the manual again to confirm that your setup is fine.

Key Features
- 3-axis gyro-based stabilization.
- Two stabilization modes - "**RATE**" and "Attitude **HOLD**"
- PID values are adjustable with a supported programing box (AQUASTAR or dlux programming box)
- Remote On/Off and Master Gain Control using the AUX1 channel
  - AUX1 channel controls master gain linearly from 0% to 100%
  - AUX1 channel also controls RATE and HOLD modes
- DIP switches on RX3S or Programming Box control the mix mode.
- Fixed WING mode mixing
  - Single aileron
  - Two servo ailerons (enables you to use flaperons).
- DELTA mode mixing.
- VTAIL mode mixing.
- Support for Spektrum Serial and S.BUS (RX3SM not supported).
- CPPM mode (RX3SM not supported).
  - Throttle and flap channels pass through in CPPM mode
- Configurable mixer limits to prevent servo binding.
- Stick-based configuration or Programming Box (for changing from default options)
- EEPROM storage for configurations (can be reset through jumpers on bootup)

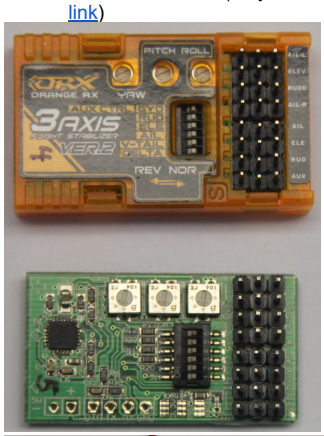
○ VR POT gain value can be overridden using the programing box.

● Stick-controlled rotation RATE in HOLD mode, optional in RATE mode.
● Option for Inflight RX calibration to reset RX neutral values.
● Stick position controls gain, which prevents overcorrection or "fighting" commanded stick movement during manoeuvres.
● Device can also be mounted sideways on the fuselage (i.e. flat fuselage foamies) by rolling 90° left or 90° right and then mounting to the fuselage side.
● Optional AVRootloader to allow firmware update without usbasp ISP.
● Use pre-compiled firmware or compile your own version from the source code.

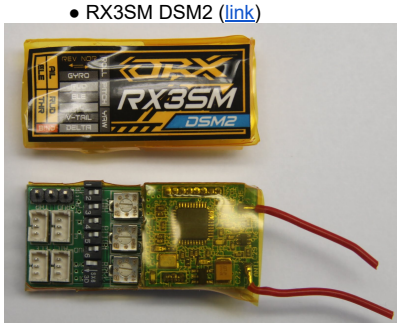
Supported Devices

● RX3S V1 (link)

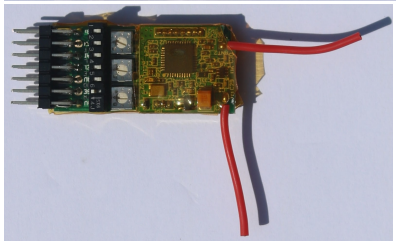
● RX3S V2 or V3 (they use the same firmware) (V2-link or V3-link)



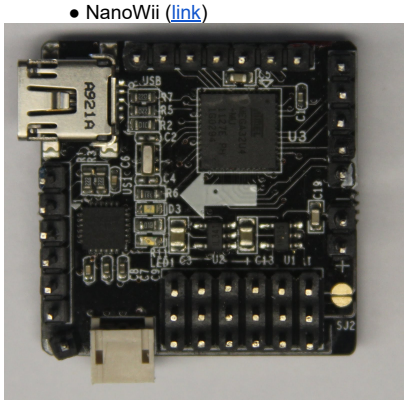
● RX3SM DSM2 (link)



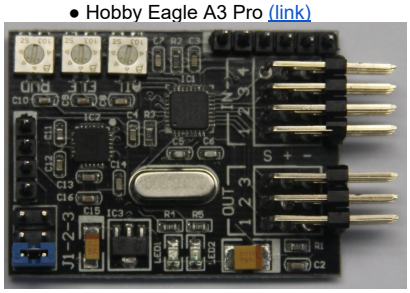
● RX3SM DSMx (link)



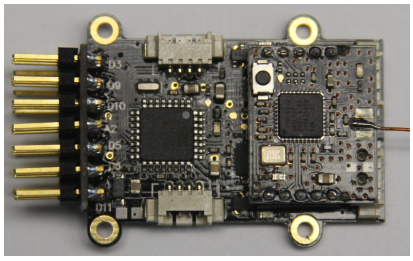
● NanoWii (link)



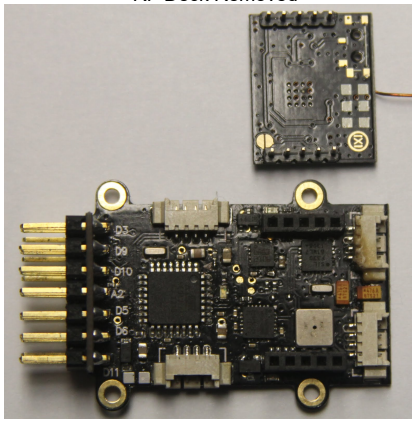
● Hobby Eagle A3 Pro (link)



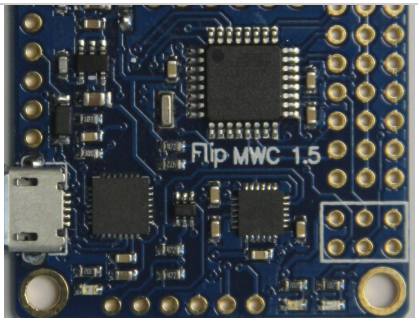
● Hobby King MINI MWC with DSM2 (link)
With RF Deck



RF Deck Removed



● Flip MWC 1.5 (Link)



- Gyros
  - ITG-320x with RX3S
  - MPU-6050 with NanoWii (gyro only) sensors
- Aquastar Programming Box (link)
- DLUX Programming Box (link)

Programming Overview

This section describes the layout of the ATMEGA flash memory and how it is utilized both with and without a Bootloader.

Without Bootloader

- The factory supplied HK devices (RX3S V1, RX3S V2 & RX3S V3) reserve the entire Flash memory space for flight stabilization firmware.
- This Flash memory space is factory programmed by HK using a 6 pin In Circuit Programming header (ISP). The ISP is designed to be connected to something like a 3.3V usbasp programmer.
- Each time you wish to update the firmware the usbasp will be required again as described below in the **Flashing the Firmware (for non Arduino-based boards)** section.
- The only programmer required is a 3.3V usbasp

Using the Bootloader

- This method reconfigures the Flash memory into two segments.
  - A very small segment at the very end of the Flash memory is used to hold the Bootloader. A Bootloader is a small piece of firmware that gets control each time the device is reset (such as with a power cycle). This firmware will wait for a specified period of time looking for a unique programming tool that may be connected by some means other than an ISP connector. If a programming device is not found in the specified period of time, the Bootloader firmware passes control to the flight stabilizer firmware. A usbasp programmer is required to load initially load the Bootloader and will not be required to be used again unless the Bootloader is damaged or removed.
  - The remainder of the Flash memory not being used by the Bootloader is dedicated for the flight stabilizer firmware. This section of memory is **not** programmed by an ISP. In the case of Open FlightStab, it is programmed by a 3 wire programmer connected to one of the 3 pin connectors normally used for RX inputs or servo outputs.
- The use of a Bootloader enables very easy updates of the firmware without removal of the flight controller from the aircraft using the same connector that is used for the Programming Box.
- Two programmers are required, a 3.3V usbasp to initially install the Bootloader (AVRootloader) and a 3 wire programming tool such as those described in the **USB-Serial programming** section.

Flashing the Firmware (for non Arduino-based boards)

This section deals with flashing the Open Flight Stabilizer firmware onto your device. Two types of usbasp devices have been tested and these are described in USB-Serial programming adapters section.

The MINI MWC & Flip MWC 1.5 processors operates at 5V and must be flashed with a programmer configured for 5V operation.

The RX3S family of devices are best programmed using a programmer configured for 3.3V operation.

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 (see below).
- This optional socket firmware flashing tool (link) is highly recommend especially if you have the RX3S and intend to use an Aquastar programing box (different chip type but the tool fits the cpu).

Flashing with a 5V usbasp when a 3.3V usbasp is not available

Caution - doing things in an incorrect order can damage the ATmega168PA which will require replacement. There is still a minor risk even if everything is done in the proper order as the signals from the usbasp will be 5V signals that the ATmega168PA will try to clamp to about 4V.

- There is usually a jumper on the usbasp that can be set (or removed) to eliminate sending power to the device to be programmed. Set that jumper as needed to prevent sending 5V power to the RX3S.
  Placing 5V on the ISP connector power pin will most likely result in damage to the gyro.
- Power on the RX3S from the BEC of and ESC **before** connecting the usbasp. **Connecting the usbasp while the RX3S is not powered will most likely destroy the ATmega168PA.**
- Program and verify the firmware and change fuse bits if required.
- Disconnect the usbasp
- Remove power from the RX3S

Flash using eXtreme Burner

Setting up eXtremeBurner (first time)
- install eXtreme Burner - AVR. Latest version 1.4.3 is available at this Link. Look near the PayPal donate button. This version is still missings the updates for the ATmega168PA described below.
- 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 (for the RX3S devices)
- connect the ISP connector to the device. **Do not connect your USBASP to your computer yet. Please make sure all connections to chip are appropriate before providing power. If you are using the Hobbyking Atmel Atmega Socket Firmware Flashing Tool, try to be still especially during flashing.**
- if you are not powering the ATmega168PA via the USBASP then Power on the RX3S from the BEC of and ESC **before** connecting the usbasp.
  Connecting the usbasp while the RX3S is not powered will most likely destroy the ATmega168PA.
- **Only when you are satisfied that everything is connected correctly then connect your USBASP to your computer.**
- select **Read All**. the program should recognize the chip and proceed to read the flash, EEPROM and fuse/lock bits. **If not, disconnect the USBASP from your computer and repeat the steps above.**

not be changed):

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

○ the fuse/lock bits for the Flip 1.5 devices should read the following:
The calibration value is not important (and should not be changed):

```
lo=0xff
hi=0xda
ext=0xf9
lock=0xff
```

● DO NOT PROCEED IF THE VALUES DO NOT MATCH OR IF EXTREME BURNER CANNOT RECOGNIZE THE CHIP (AFTER 2-3 TRIES).
○ If the values do not match because the ATmega168PA has been replaced then write the lo, hi and ext with the values listed above, write **lock = 0xff** and it should not be necessary to do the **Chip Erase** listed below.
● FROM THIS POINT ONWARDS, YOU WILL ERASE THE CHIP AND REFLASH WITH NEW FIRMWARE. THERE IS **NO WAY** TO RESTORE THE ORIGINAL FIRMWARE unless you have been able to read and save the factory firmware (only possible if lock bits were 0xff)..
● 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)

● Verify programmer configuration, 3.3V for RX3S family of devices, 5V for MINI MWC & Flip MWC 1.5.
● 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. Building firmware from sources is optional 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 device.

Flash using avrdude

● install arduino v1.0.3 (http://arduino.cc/en/Main/Software)
Note :Arduino must be installed in: C:\Program Files\arduino-1.0.3
● 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 as in the previous section to verify the fuse/lock bits before erasing the chip.
● to flash the device, use 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)

```
REM Trigger Programming Port
avrdude -p m32u4 -P COM7 -c avr109 -b 1200
REM Delay for 3 seconds
PING -n 4 127.0.0.1>nul
REM Program Flash
avrdude -p m32u4 -P COM5 -c avr109 -D -
```

How to find Trigger and Active Comports:

- Open device manager. Plugin NanoWii and watch for new COM port to appear. That port number will be Trigger Port. Replace Blue with this number.
- Manually reset the NanoWii board while watching device manager. This will change to the Active Port number for few sec's. Change Red with this number.

Save the above code after you have updated the COM port values to a Text file and rename to 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.

If you are using Windows XP the above command may not work. Your SendTo directory may be found at c:\Documents and Settings\your_user_name\SendTo.

How to use

- Right click on *_NanoWii.hex file you wish to program into the NanoWii and go to option 'Send to' chose NanoFlash and your done.

Optional: Flashing the AVRootloader

It is possible to flash a bootloader via usbasp on the RX3S, MINI MWC, Flip MWC 1.5 and/or the Programming box. If the bootloader is installed, then flashing the main stabilizer firmware or Programming box does not require the usbasp. Instead, a USB-serial adapter may be connected to the **AILR_OUT** channel of the stabilizer with **RX3S** devices, **RUD_OUT** channel with the **RX3SM** devices or the **A0** connector on the **MINI MWC** or **Flip MWC 1.5.** The Programming box may also be connected to this connector. This is more convenient than opening up the device to upgrade the firmware via the usbasp ISP connector.

Steps to flash the bootloader using the USBASP (note: for RX3S, RX3SM, MINI MWC & Flip MWC 1.5 only):

- if flashing for the first time, follow earlier instructions to erase the chip and clear the lock bits.
- **RX3S & RX3SM Devices**
  - Verify Programmer is configured for 3.3V operation
  - use the flash 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. Make sure to put a tick mark in the checkbox for the fuse you wish to alter. After you do the write, read the fuse bits again to verify they are correct.
- **MINI MWC & Flip MWC 1.5**
  - Verify Programmer is configured for 5V operation
  - use the flash hex file at .\FlightStab\AVRootloader\AVR\default**AVRootloader_MINI_MWC_A0.**hex
  - update the **HIGH FUSE** setting from **0xDA** to **0xDE**. This enables the atmega328p to boot a 512-byte bootloader on reset.. Make sure to put a tick mark in the checkbox for the fuse you wish to alter. After you do the write, read the fuse bits again to verify they are correct.

Caution: Once the AVRootloader has been installed, the ISP connector should **not** be used again unless the AVRootloader is being replaced or updated. If the ISP connector is used for programming, the AVRootloader will be erased.

Instructions for flashing the AVRootloader into the Programming box can be found in the Programming Box Support section.

USB-Serial programming adapters

Two different devices for flashing the firmware have been tested to date as described in the two sections that follow:

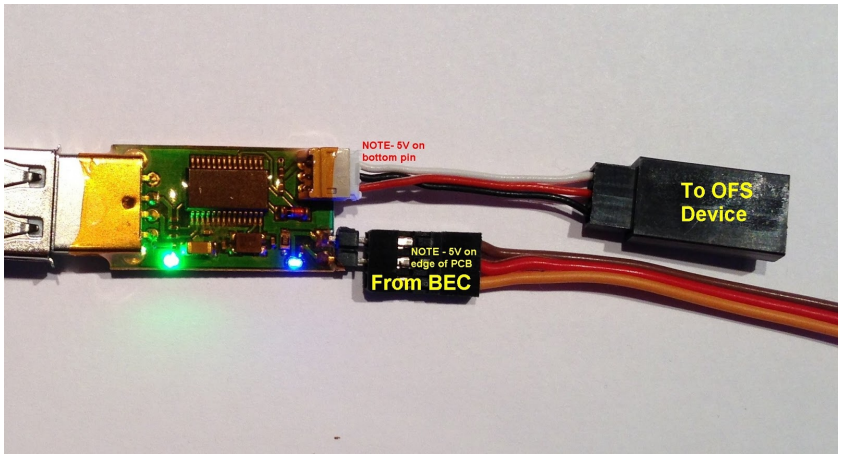
OrangeRX USB Firmware Kit for JR/Futaba Style Transmitter Module

Programmer
Cable

This device requires a device driver which can be obtained here:

10. In addition, if the board you have uses an older version of the Prolific chip it might not even work with Windows 8 and above. A tool is provided in the drive download package to check what chip is on the board you have.

The programmer is supplied as 2 PCB's. We will only use the PCB with the USB connector. The cable listed above will plug into the socket on this PCB and the BEC from your ESC plugs into the 2 pins on the edge of the board. Photo of components used shown below.



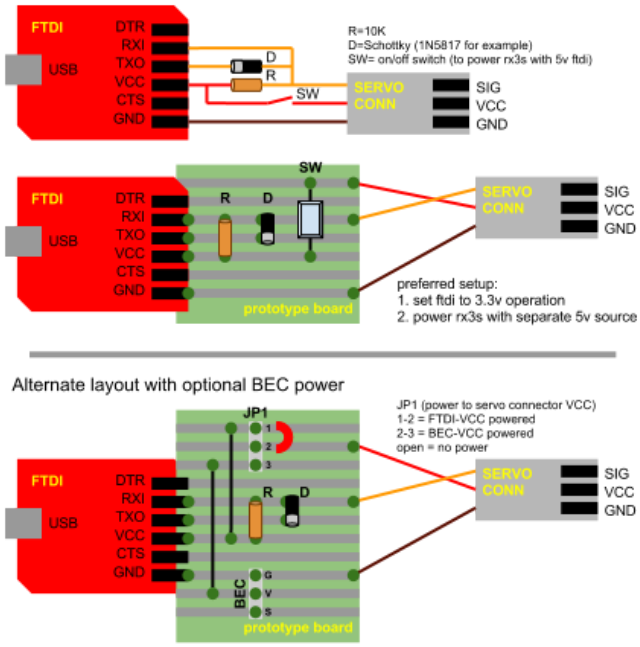
Note the orientation of the BEC connector, the orange wire is not connected. The green LED indicates that the USB is connected and the blue LED indicates that the BEC is providing power. Never connect your device to the programming cable unless **both** LED's are on.

This device does not have a voltage switch. It's normal output is a 4V signal which is high enough to program 5V devices and the RX3S family of devices have current limiting resistors for the signal line on all of the 3 pin connectors and will not be damaged by this programmer.

FTDI USB-Serial programming adapter

This device requires a device driver which can be obtained here:
Device Drivers

This device requires modification, the circuit mod follows:
https://docs.google.com/drawings/d/1WhYAdw_9QpfYJWmOuu2IhCJNw6rNNagTw_OZEqriTo8/pub?w=867&h=439



One source for this programmer is
http://www.tinyosshop.com/index.php?route=product/product&product_id=600

Modifications to allow the TinyOS programmer to supply 5V power and 3.3V signals can be found at the end of this document in the "**TinyOS FTDI Programmer Modifications**" section.

● connect the USB from the the OrangeRX USB or the FTDI + circuit mod programmer to your machine and wait until the drivers are installed (if not installed already)
● 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.
● prepare your wires and make sure you know where you need to connect the programming wire
  ○ RX3S V1/V2 **AILR_OUT** channel
  ○ RX3SM **RUD_OUT** channel
  ○ MINI MWC **A0** connector
  ○ Flip MWC 1.5 **A0** connector
● Set the programmer to 3.3V for the RX3S family of devices, set it to 5V for the MINI MWC & Flip MWC 1.5. **If you forget to do this you may not be able to connect to the device.**
● If the programmer will not supply power to the device,
● Click “connect to device” on the software and then connect the wire to the RX3S gyro or the Programming box, which should establish a session with the device. If the serial programmer is not providing power to the device, apply device power at this time. There is a finite time you can do this so if you fail, disconnect the wire from the gyro or the programming box (remove power from the device if not powered by the programmer) and try again.
● **Note** the software can’t connect if you first connect the gyro or the programming box to the usb programmer (or they are already powered up). The reason is because the gyro firmware waits for 0.5 sec to connect to the programmer and then it starts to initialize. There is a similar delay on the programming box firmware.
● Select the appropriate firmware (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.
● If programming a MINI MWC or Flip MWC 1.5, set the programmer back to 3.3V.

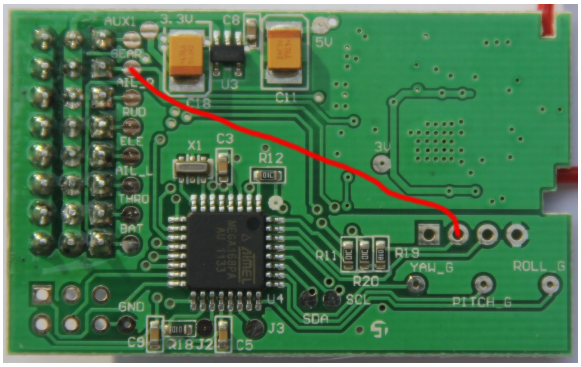
Flight Stabilizer Board Modifications

RX3S V3 Dual Aileron Support

Instructions for transmitters that support 2nd aileron on Channel 5 (Gear)

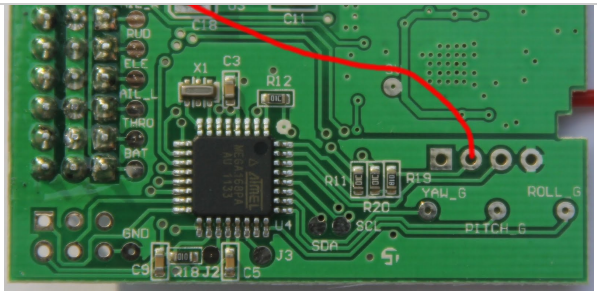
This modifications adds 1 wire to the bottom of the PCB to connect the JP4 TX pin to the GEAR pad which is used as the 2nd aileron input to the Atmel chip. The wire should be held in place with small dabs of Hot Glue in several places.

The needed modification is shown in the photo below.



Instructions for transmitters that force 2nd aileron on Channel 6 (Aux)

The first modification connects the Aux input (2nd Aileron) to the Atmel chip. Add a wire on the bottom of the PCB to connect the TX pin of JP4 to the AUX1 pad as shown in the photo below. Again use several dabs of Hot Glue on the wire

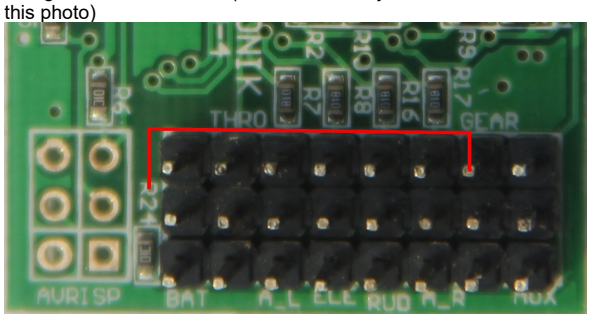


The second modification is to move the Mode/Gain control from channel 6 to channel 5. You must also modify your transmitter setup to use channel 5 for Mode/Gain control.

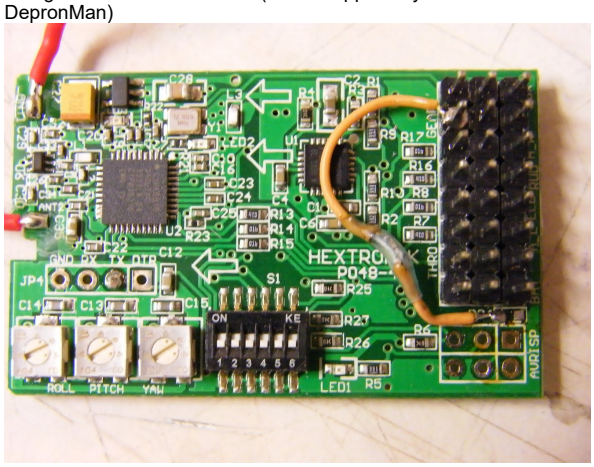
This change requires the removal of R24. Then R24 can be installed such that one end connects to the original R24 pad closest to the to the center of the board with the resistor body covering the R24 silk screen. Then connecting a wire to the free end of R24 to the pad labeled GEAR. Cover the resistor and several places on the wire with a small amount of Hot Glue that is easily removed with alcohol.

The other option is to remove and discard surface mount R24 and replace it with a discrete 10K resistor connected from the R24 pad closest to the center of the board and the pad labeled GEAR. Again use several dabs of Hot Glue on the wire and resistor to hold them in place. Both options are shown in the photos below, choose only 1 option.

Using surface mount R24 (resistor has not yet been moved in this photo)



Using a discrete 10K resistor (Photo supplied by user DepronMan)



Modification required for Dual Aileron Support for the Hobby King RX3SM DSMx

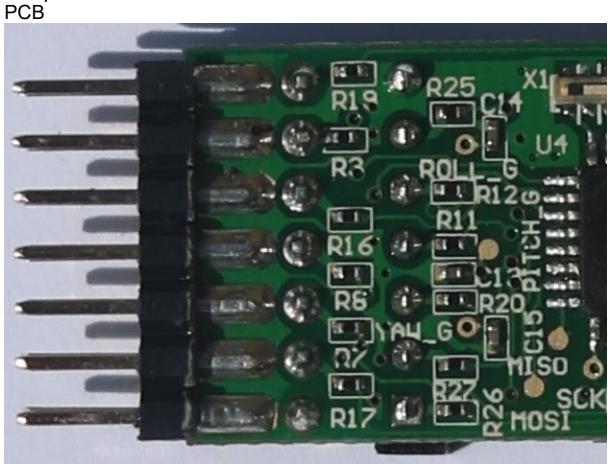
NOTE - This rework only applies to the DSMx version of the RX3SM

This rework will remove the Gear Channel output from the end connector and reroute the gear signal to to Atmel processor as the second aileron (AILR) input for stabilization. The output end connector label GEAR will now be used for the right Aileron servo signal.

The RF deck on the DSMx version of this device is soldered to the

0402 SMD parts on hand. Mouser part numbers for these parts are:
0402 1K ohm resistor - 667-ERJ-2GEJ102X
0402 10K ohm resistor - 667-ERJ-2RKF1002X (only needed for device restoration)

The photo below shows the area to be reworked on an unmodified PCB



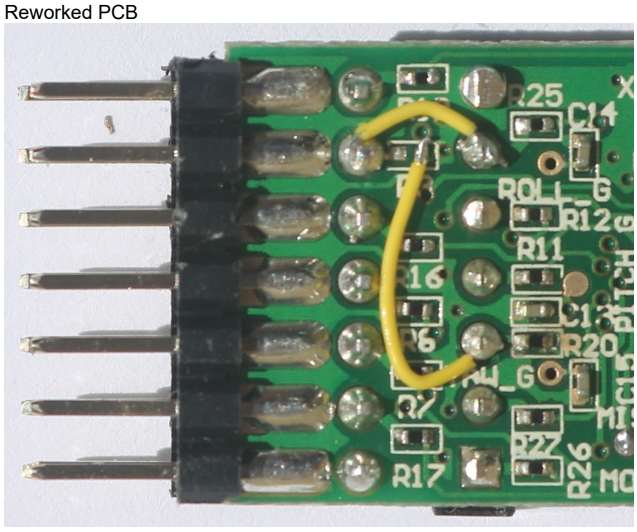
The first step of the rework is to remove R25 and set it aside (if you can find it after removal). This is a 10K resistor that will no longer be used but will be required if you ever decide to restore the board to the factory wiring.

The second step is to carefully remove R3 and then install it at the R25 location. Hopefully you have a spare on hand in case R3 was lost in the removal process.

The third step is to add a short piece of insulated wire from the pad used for R3 that is closest to the center of the PCB to the switch pin that connects to R20. It is much easier to solder to the switch pin than to the end of R20.

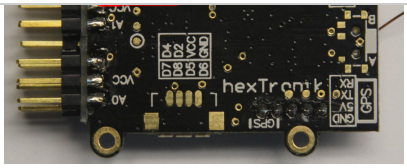
The last step is to add a short piece of insulated wire from the signal pin of the gear connector (near the other end of R3).

Reworked PCB



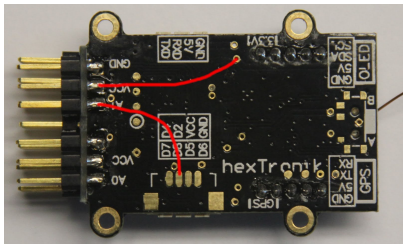
Modification required for the Hobby King MINI MWC

This board was designed for MultiRotor use and does not support servos. The reason for this is that the recommended power is a 2S LiPo connected to the A2 connector. The center pin of all of the connectors are tied together on the PCB and with 7.4V connected to A2, all of the servo jacks will have 7.4V which may damage most servos. Thus for Aircraft use with servos, a minor modification is required on the bottom of the PCB. This modification will tie the output of the onboard 5V regulator to the center pins of all the connectors. A2 is then connected to the 5V output of your BEC to power the MINI MWC and the connected servos. The photo below depicts the modification required for all configurations of this board..



Modification for the Hobby King MINI MWC using External CPPM

This mode requires that an additional wire be added to the bottom of the PCB. This wire does not interfere with other configurations so it may be left connect for all configurations.

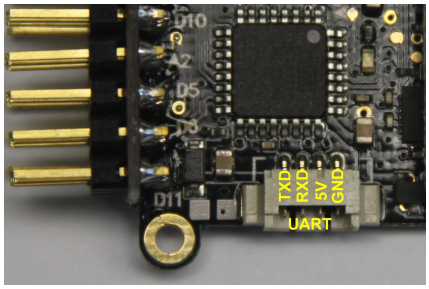


The RF deck should be removed. CPPM cable from RX connected to EXT CPPM connector in photo for end connectors shown above. The CPPM cable can also supply 5V in which case nothing needs to be connected to A2.

Hobby King MINI MWC SBUS Connections

The SBUS signal must be connected, through an inverter, to the RXD signal of the UART connector shown in the photo below. This requires a 1.25mm JST connector. One source for this connector is here (link). Other sources may be available. The GND signal for the SBUS cable should be connected to GND on the UART connector.

In addition a cable with 5V and ground must also be connected to the A2 connector. This is because the signal labeled 5V on the UART connector is not the 5V signal on the PCB, it is actually the input to the 5V regulator on the PCB.



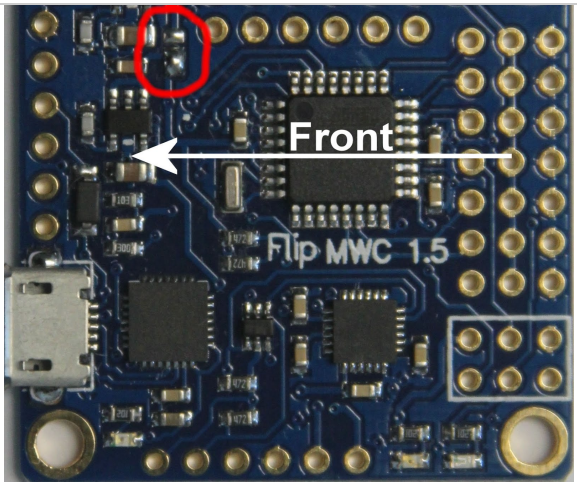
Modification for the Flip MWC 1.5

This board has an onboard 5V regulator. If you plan to use a 6V UBEC with 6V servos you can use the board without modification. If you wish to use 5V BEC and 5V servos the board requires a minor modification.

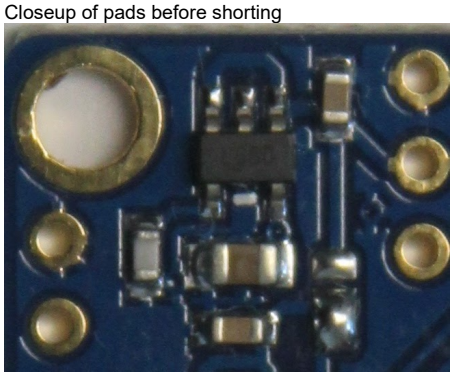
Modifications - Original Flip MWC 1.5

There are two adjacent pads on the PCB where you can place a short jumper or just a solder “blob” to bypass the 5V regulator. Note that there are actually 3 pads in the circled area and the bottom 2 already have a solder “blob” from the factory. All 3 pads need to be connected by a single solder “blob”.

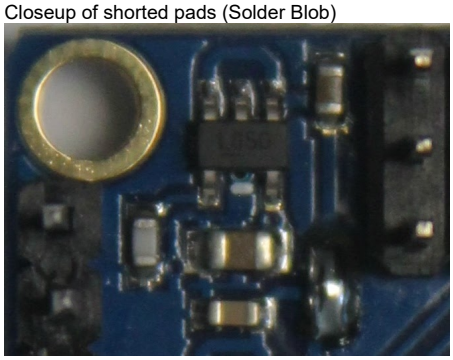
Location of aircraft front and pads to bypass 5V regulator (red circle area).



Closeup of pads before shorting

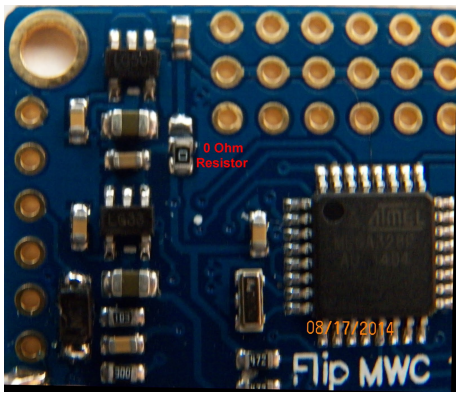


Closeup of shorted pads (Solder Blob)



Modifications - New Flip MWC 1.5

Some boards have a 0 Ohm resistor connecting the 2 bottom pads. That resistor should be left in place with a solder “blob” added between the top of the 0 Ohm resistor and the open pad above it to bypass the 5V regulator.



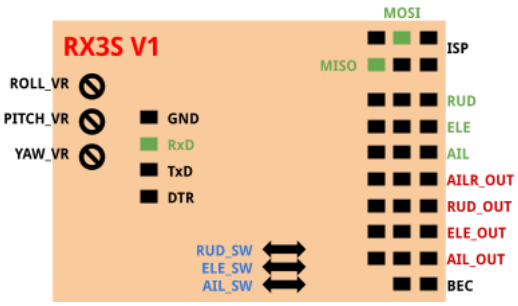
Connection/Pinout Tables

Mix Mode	Description
RUDELE 1-AIL	Rudder + Elevator Tail (pitch and yaw). Single Aileron.
DELTA 1-AIL	Elevon Tail (pitch and roll). Single Aileron and Rudder.
VTAIL 1-AIL	Vee Tail (pitch and yaw). Single Aileron. No Rudder.
RUDELE 2-AIL	Rudder + Elevator Tail (pitch and yaw). Dual Ailerons (Flapperons).
DELTA 2-AIL	Elevons with “FLAPPERONEVATORs”. No Rudder. RUD_IN is the LINKED input. RUD_VR is pitch gain on FLAPPERONEVATORs. Click here for description
VTAIL 2-AIL	V Tail with Flapperons.
DUCKERON	4 split wing control surfaces (pitch, roll, yaw and brake). Yaw by differential braking. AILR_IN is the BRAKE input.
RUD 2-ELE 2-AIL	Rudder + Dual Ailerons and Dual Elevators

- Mix mode set by DIP switches, stick config or program box. Some modes in RX3S V1 must be set via the config CFG (stick config or program box).
- with SERIALRX_*, the AUX1 channel is used for MODE
- Spektrum satellites can only tolerate 3.3V, 5 volts may cause damage. A 3.3V connection on the Stabilizer must be found or a 3.3V regulator added to power the satellite.
- Warning - The Spektrum Satellites do not include any failsafe function. They were never intended to be used standalone.
- For S.BUS, short cable end of Tarot Signal Inverter connects to Receiver.

Legend

Cell Color	Pin Type
Blue	DIP Switch
Orange	Servo or RX Pin
Red	Output to Servo or ESC
Green	Input from RX



RX3S V1

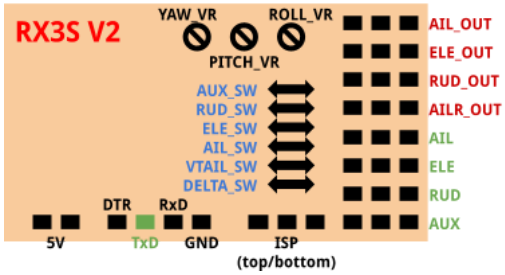
	RudEle 1-AIL	Delta 1-AIL	VTail 1-AIL	RudEle 2-AIL	Delta 2-AIL	VTail 2-AIL	Duck	Rud 2-ELE 2-AIL	
RUD_SW	NOR ▶	REV ◀	NOR ▶	REV ◀	via CFG	via CFG	via CFG	N/A	
ELE_SW	NOR ▶	NOR ▶	REV ◀	REV ◀	via CFG	via CFG	via CFG	N/A	
AIL_SW (TxD)								N/A	
AIL_OUT	AIL	DELTA-2	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	N/A	
ELE_OUT	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	N/A	
RUD_OUT	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	N/A	
AILR_OUT	MODE	MODE	MODE	AIL-R	AIL-R	AIL-R	RIGHT-1	N/A	1-WIRE
AIL_IN	AIL	AIL	AIL	AIL-L	AIL-L	AIL-L	AIL-L	N/A	
ELE_IN	ELE	ELE	ELE	ELE	ELE	ELE	ELE	N/A	
RUD_IN	RUD	RUD	RUD	RUD	**LINKED**	RUD	RUD	N/A	

LED flashes	2	3	4	5	6	7	8	9	

RX3S V1 CPPM/Serial (SPEKTRUM/SBUS/SRXL)

	RudEle 1-AIL	**Delta 1-AIL**	**VTail 1-AIL**	**RudEle 2-AIL**	**Delta 2-AIL**	**VTail 2-AIL**	**Duck**	**Rud 2-ELE 2-AIL**	
RUD_SW	NOR ▶	REV ◀	NOR ▶	REV ◀	via CFG	via CFG	via CFG	via CFG	
ELE_SW	NOR ▶	NOR ▶	REV ◀	REV ◀	via CFG	via CFG	via CFG	via CFG	
AIL_SW (TxD)									
AIL_OUT	AIL	DELTA-2	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	AIL-L	
ELE_OUT	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	ELE-L	
RUD_OUT	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	RUD	
AILR_OUT	AIL	AIL	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	AIL-R	1-WIRE
AIL_IN	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	
ELE_IN	FLAP	FLAP	FLAP	FLAP	FLAP	FLAP	FLAP	ELE-R	
RUD_IN	THR	THR	THR	THR	THR	THR	THR	THR	
ISP ↑ pin (mosi)									
ISP ↙ pin (miso)									
RxD pin	serial	serial	serial	serial	serial	serial	serial	serial	
LED flashes	2	3	4	5	6	7	8	9	

- See these NOTES on Spektrum and SBUS connections.



RX3S V2/V3

	RudEle 1-AIL	**Delta 1-AIL**	**VTail 1-AIL**	**RudEle 2-AIL**	**Delta 2-AIL**	**VTail 2-AIL**	**Duck**	**Rud 2-ELE 2-AIL**	
AUX_SW (RxD)								N/A	
RUD_SW								N/A	
ELE_SW	NOR ▶	NOR ▶	NOR ▶	NOR ▶	REV ◀	REV ◀	REV ◀	N/A	
AIL_SW (TxD)				NOR ▶	NOR ▶	NOR ▶	NOR ▶	N/A	
VTAIL_SW	NOR ▶	NOR ▶	REV ◀	REV ◀	NOR ▶	REV ◀	NOR ▶	N/A	
DELTA_SW	NOR ▶	REV ◀	NOR ▶	REV ◀	REV ◀	NOR ▶	NOR ▶	N/A	
AIL_OUT	AIL	DELTA-2	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	N/A	
ELE_OUT	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	N/A	
RUD_OUT	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	N/A	
AILR_OUT	AIL	AIL	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	N/A	1-WIRE
AIL_IN	AIL	AIL	AIL	AIL-L	AIL-L	AIL-L	AIL	N/A	

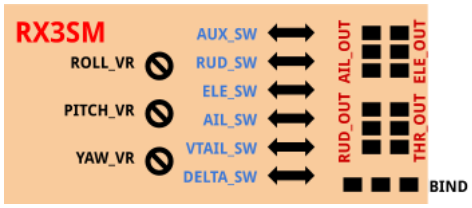
TxD pin				AIL-R	AIL-R	AIL-R	**BRAKE**	**N/A**	
LED flashes	2	3	4	5	6	7	8	9	

- if using TxD pin for input (AIL-R IN or BRAKE IN), AIL_SW must be set to NOR ▶
- V3 support has no *_IN and is limited to "* 1-AIL" modes without mods

RX3S V2 CPPM/Serial (SPEKTRUM/SBUS/SRXL)

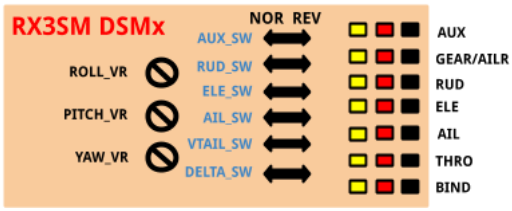
	RudEle 1-AIL	**Delta 1-AIL**	**VTail 1-AIL**	**RudEle 2-AIL**	**Delta 2-AIL**	**VTail 2-AIL**	**Duck**	**Rud 2-ELE 2-AIL**	
AUX_SW (RxD)	NOR ▶	NOR ▶	NOR ▶	NOR ▶	NOR ▶	NOR ▶	NOR ▶	NOR ▶	
RUD_SW									
ELE_SW	NOR ▶	NOR ▶	NOR ▶	NOR ▶	REV ◀	REV ◀	REV ◀	NOR ▶	
AIL_SW (TxD)									
VTAIL_SW	NOR ▶	NOR ▶	REV ◀	REV ◀	NOR ▶	REV ◀	NOR ▶	NOR ▶	
DELTA_SW	NOR ▶	REV ◀	NOR ▶	REV ◀	REV ◀	NOR ▶	NOR ▶	NOR ▶	
AIL_OUT	AIL	DELTA-2	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	AIL-L	
ELE_OUT	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	ELE-L	
RUD_OUT	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	RUD	
AILR_OUT	AIL	AIL	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	AIL-R	1-WIRE
AIL_IN	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	
ELE_IN	FLP	FLP	FLP	FLP	FLP	FLP	FLP	ELE-R	
RUD_IN	THR	THR	THR	THR	THR	THR	THR	THR	
AUX_IN	AUX2	AUX2	AUX2	AUX2	AUX2	AUX2	AUX2	AUX2	
TxD pin									
RxD pin	serial	serial	serial	serial	serial	serial	serial	serial	
LED flashes	2	3	4	5	6	7	8	9	

- Rud 2-ELE 2-AIL must be selected by CFG (Stick or Programming Box)
- If using SPEKTRUM/SBUS via serial port (RxD pin), AUX_SW must be set to NOR ▶
- See these NOTES on Spektrum and SBUS connections.



RX3SM DSM2

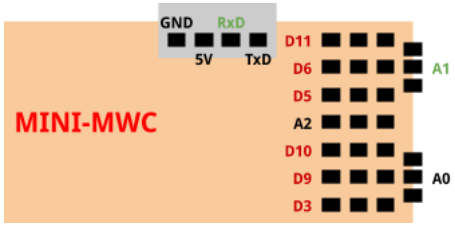
	RudEle 1-AIL	**Delta 1-AIL**	**VTail 1-AIL**	**RudEle 2-AIL**	**Delta 2-AIL**	**VTail 2-AIL**	**Duck**	**Rud 2-ELE 2-AIL**	
GYRO/AUX_SW									
RUD_SW									
ELE_SW									
AIL_SW (TxD)									
VTAIL_SW (RxD)	NOR ▶	NOR ▶	REV ◀						
DELTA_SW	NOR ▶	REV ◀	NOR ▶						
AIL_OUT	AIL	DELTA-2	AIL						
ELE_OUT	ELE	DELTA	VTAIL						



RX3SM DSMx

	RudEle 1-AIL	Delta 1-AIL	VTail 1-AIL	RudEle 2-AIL	Delta 2-AIL	VTail 2-AIL	Duck	Rud 2-ELE 2-AIL	
GYRO/AUX_SW	NOR ▶	NOR ▶	NOR ▶	NOR ▶	NOR ▶	NOR ▶			
RUD_SW									
ELE_SW									
AIL_SW (TxD)	NOR ▶	NOR ▶	NOR ▶	NOR ▶	NOR ▶	NOR ▶			
VTAIL_SW (RxD)	NOR ▶	NOR ▶	REV ◀	REV ◀	NOR ▶	REV ◀			
DELTA_SW	NOR ▶	REV ◀	NOR ▶	REV ◀	REV ◀	NOR ▶			
AIL_OUT	AIL	DELTA-2	AIL	AIL	DELTA-2	AIL			
ELE_OUT	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1			
RUD_OUT	RUD	RUD	VTAIL-2	RUD	RUD	VTAIL-2			1-WIRE
GEAR/AILR_OUT	GEAR	GEAR	GEAR	AILR	AILR	AILR			
LED flashes	2	3	4	5	3	4			

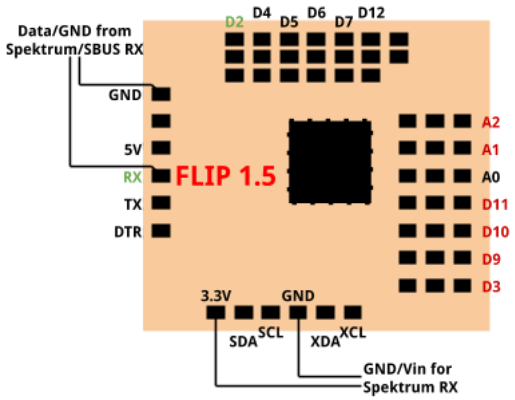
- AIL_SW and AUX_SW must be set to NOR ▶
- Columns with Green Headings are for Factory PCB
- Columns with Yellow Headings are for Modified PCB



MINI MWC CPPM/Serial (SPEKTRUM/SBUS/SRXL)

	RudEle 1-AIL	Delta 1-AIL	VTail 1-AIL	RudEle 2-AIL	Delta 2-AIL	VTail 2-AIL	Duck	Rud 2-ELE 2-AIL	
D11 (AILR_OUT)	AIL	AIL	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	AIL-R	
D6 (RUD_OUT)	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	RUD	
D5 (ELE_OUT)	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	ELE-L	
A2 (AUX2_OUT)	AUX2	AUX2	AUX2	AUX2	AUX2	AUX2	AUX2	AUX2	
D10 (AIL_OUT)	AIL	DELTA-2	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	AIL-L	
D9 (FLP_OUT)	FLP	FLP	FLP	FLP	FLP	FLP	FLP	ELE-R	
D3 (THR_OUT)	THR	THR	THR	THR	THR	THR	THR	THR	
A1 (CPPM_IN)	Ext CPPM	Ext CPPM	Ext CPPM	Ext CPPM	Ext CPPM	Ext CPPM	Ext CPPM	Ext CPPM	
A0									1-WIRE
LED flashes	2	3	4	5	6	7	8	9	

- The 5V pin on the FTDI connector should not be used for receiver power. That pin has a diode in series with the 5V net which may result in low receiver voltage.
- A0 = one-wire pin for upgrading and configuring
- A1 = EXT CPPM IN (for MINI_MWC_EXTERNAL_RX only). SHORT with PB0

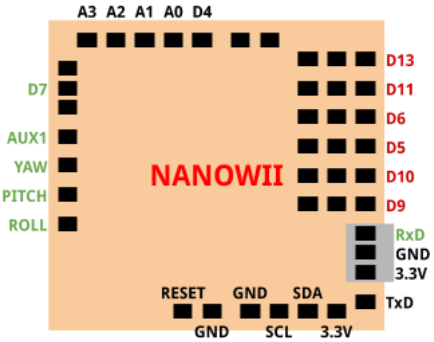


FLIP 1.5

	RudEle 1-AIL	Delta 1-AIL	VTail 1-AIL	RudEle 2-AIL	Delta 2-AIL	VTail 2-AIL	Duck	Rud 2-ELE 2-AIL	
D11 (AILR_OUT)	AIL	AIL	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	AIL-R	
D3 (RUD_OUT)	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	RUD	
D9 (ELE_OUT)	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	ELE-L	
D10 (AIL_OUT)	AIL	DELTA-2	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	AIL-L	
A2 (FLP_OUT)								ELE-R	
A1 (THR_OUT)									
D2 (THR/PPM)									
D4 (AIL)	AIL	AIL	AIL	AIL-L	AIL-L	AIL-L	AIL	AIL-L	
D5 (ELE)	RUD	RUD	RUD	RUD	**LINKED**	RUD	RUD	RUD	
D6 (YAW)	ELE	ELE	ELE	ELE	ELE	ELE	ELE	ELE-L	
D7 (AUX1)				AIL-R	AIL-R	AIL-R	**BRAKE**	AIL-R	
D12 (AUX2)	MODE	MODE	MODE	MODE	MODE	MODE	MODE	MODE	
A0								ELE-R	1-WIRE
LED flashes	2	3	4	5	6	7	8	9	

FLIP 1.5 CPPM/Serial (SPEKTRUM/SBUS/SRXL)

	RudEle 1-AIL	Delta 1-AIL	VTail 1-AIL	RudEle 2-AIL	Delta 2-AIL	VTail 2-AIL	Duck	Rud 2-ELE 2-AIL	
D11 (AILR_OUT)	AIL	AIL	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	AIL-R	
D3 (RUD_OUT)	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	RUD	
D9 (ELE_OUT)	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	ELE-L	
D10 (AIL_OUT)	AIL	DELTA-2	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	AIL-L	
A2 (FLP_OUT)	FLP	FLP	FLP	FLP	FLP	FLP	FLP	ELE-R	
A1 (THR_OUT)	THR	THR	THR	THR	THR	THR	THR	THR	
D12 (AUX2)	AUX2	AUX2	AUX2	AUX2	AUX2	AUX2	AUX2	AUX2	
D2 (CPPM_IN)	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	
RxD pin	serial	serial	serial	serial	serial	serial	serial	serial	
A0									1-WIRE
LED flashes	2	3	4	5	6	7	8	9	



NANO WII

	RudEle 1-AIL	Delta 1-AIL	VTail 1-AIL	RudEle 2-AIL	Delta 2-AIL	VTail 2-AIL	Duck	Rud 2-ELE 2-AIL	
D7				AIL-R	AIL-R	AIL-R	**BRAKE**	AIL-R	
AUX	MODE	MODE	MODE	MODE	MODE	MODE	MODE	MODE	
YAW	RUD	RUD	RUD	RUD	RUD	**LINKED**	RUD	RUD	
PITCH	ELE	ELE	ELE	ELE	ELE	ELE	ELE	ELE-L	
ROLL	AIL	AIL	AIL	AIL-L	AIL-L	AIL-L	AIL-L	AIL-L	
A3								ELE-R	
D13 (AILR_OUT)	AIL	DELTA-2	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	AIL-R	1-WIRE
D11 (RUD_OUT)	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	RUD	
D6 (FLP_OUT)								ELE-R	
D5 (THR_OUT)									
D10 (ELE_OUT)	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	ELE-L	
D9 (AIL_OUT)	AIL	AIL	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	AIL-L	
LED flashes	2	3	4	5	6	7	8	9	

NANO WII CPPM/Serial (SPEKTRUM/SBUS/SRXL)

	RudEle 1-AIL	Delta 1-AIL	VTail 1-AIL	RudEle 2-AIL	Delta 2-AIL	VTail 2-AIL	Duck	Rud 2-ELE 2-AIL	
D7	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	
AUX									
YAW									
PITCH									
ROLL									
D13 (AILR_OUT)	AIL	DELTA-2	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	AIL-R	1-WIRE
D11 (RUD_OUT)	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	RUD	
D6 (FLP_OUT)	FLP	FLP	FLP	FLP	FLP	FLP	FLP	ELE-R	
D5 (THR_OUT)	THR	THR	THR	THR	THR	THR	THR	THR	
D10 (ELE_OUT)	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	ELE-L	
D9 (AIL_OUT)	AIL	AIL	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	AIL-L	
RxD	serial	serial	serial	serial	serial	serial	serial	serial	
LED flashes	2	3	4	5	6	7	8	9	

- See these NOTES on Spektrum and SBUS connections.

A3 PRO (TBD TBD TBD)

AIL_SW	TBD	TBD	TBD	TBD	TBD	TBD	TBD	N/A	
ELE_SW	TBD	TBD	TBD	TBD	TBD	TBD	TBD	N/A	
RUD_SW	TBD	TBD	TBD	TBD	TBD	TBD	TBD	N/A	
AILR_IN				AIL-R	AIL-R	AIL-R	**BRAKE**	N/A	
AUX_IN	MODE	MODE	MODE	MODE	MODE	MODE	MODE	N/A	
RUD_IN	RUD	RUD	RUD	RUD	RUD	**LINKED**	RUD	N/A	
ELE_IN	ELE	ELE	ELE	ELE	ELE	ELE	ELE	N/A	
AIL_IN	AIL	AIL	AIL	AIL-L	AIL-L	AIL-L	AIL-L	N/A	
RUD_OUT	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	N/A	1-WIRE
ELE_OUT	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	N/A	
AIL_OUT	AIL	AIL-L	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	N/A	
AILR_OUT	AIL	DELTA-2	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	N/A	
LED flashes	2	3	4	5	6	7	8	9	

A3 PRO CPPM/Serial (SPEKTRUM/SBUS/SRXL) (TBD TBD TBD TBD)

	RudEle 1-AIL	**Delta 1-AIL**	**VTail 1-AIL**	**RudEle 2-AIL**	**Delta 2-AIL**	**VTail 2-AIL**	**Duck**	**Rud 2-ELE 2-AIL**	
AIL_SW (RxD)	serial	serial	serial	serial	serial	serial	serial	serial	
ELE_SW (TxD)	TBD	TBD	TBD	TBD	via CFG	via CFG	via CFG	via CFG	
RUD_SW	TBD	TBD	TBD	TBD	via CFG	via CFG	via CFG	via CFG	
AILR_IN									
AUX_IN	AUX-2	AUX-2	AUX-2	AUX-2	AUX-2	AUX-2	AUX-2	AUX-2	
RUD_IN	THR	THR	THR	THR	THR	THR	THR	THR	
ELE_IN	FLP	FLP	FLP	FLP	FLP	FLP	FLP	ELE-R	
AIL_IN	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	CPPM	
RUD_OUT	RUD	RUD	VTAIL-2	RUD	DELTA-2	VTAIL-2	RIGHT-2	RUD	1-WIRE
ELE_OUT	ELE	DELTA-1	VTAIL-1	ELE	DELTA-1	VTAIL-1	LEFT-2	ELE-L	
AIL_OUT	AIL	AIL-L	AIL	AIL-L	AIL-L	AIL-L	LEFT-1	AIL-L	
AILR_OUT	AIL	DELTA-2	AIL	AIL-R	AIL-R	AIL-R	RIGHT-1	AIL-R	
LED flashes	2	3	4	5	6	7	8		

- Some A3 PRO boards use ATmega88 Processors. Open FlightStab cannot be loaded on these boards because the Flash memory size is too small.
- Very Limited Testing of this device, do a through ground test.
- For SBUS/SPEKTRUM, WING_MODE must be configured via Program Box (AIL_SW is used for serial input).
- See these NOTES on Spektrum and SBUS connections.

Basic Instructions

Before attempting to use any stabilization mode it is advisable to double check and make sure you can switch the gyro off from your transmitter.

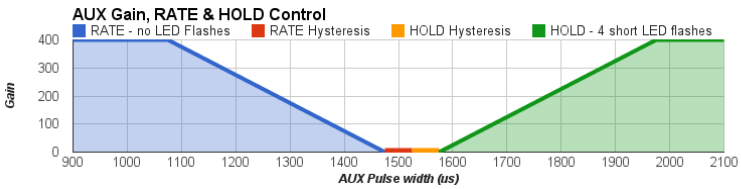
For help Programming your transmitter for switching between modes and adjusting the Master gain (and switch the gyro off) please see "Programming your transmitter for adjusting the Master gain" section.

For help adjusting PID gains please see "General guidelines for adjusting the PID gains" section.

Although numerous flights have been conducted using

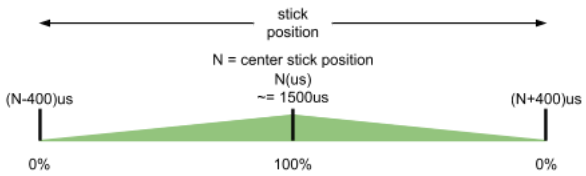
Therefore please use low VR POT gain and double check the correction direction (the gyro needs to oppose the movement/excitation). Fly to a safe altitude and use low master gain and gradually increase to full watching for any signs of oscillation.

- Connect the device based on the connection diagrams. The DIP switches (or programming box) set the WING mode (single aileron, dual ailerons or flaperons, delta or vtail) for the RX3S. The stick configs (or programming box) 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 NanoWii, MINI MWC & Flip MWC 1.5 do not have gain POT's. **A programming box is required for setup of these devices.**
- The AUX1 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 AUX1 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) to set the center position to for a pulse width of 1500us. If your TX does not display transmitted pulse width then do the following:
    - Increase the subtrim slowly for the AUX1 channel (3P switch or slider/knob in center position) until you see the 4 fast LED blinks that indicate Hold mode. At that point the TX pulse width will be about 1575us. Record the subtrim value as **maximum**.
    - Decrease the subtrim slowly until until the 4 fast LED blinks stop (Rate mode). This will be about 1525us. Record this subtrim value as **minimum** and then subtract it from the **maximum** recorded above. Save this result as **delta** which is the amount of subtrim required to change the pulse width by 50us.
    - The goal is to set the subtrim to produce a pulse width of 1500us. Divide **delta** by 2 (subtrim value for a 25us change) and record this as **offset**. Subtract **offset** from **minimum** and move the subtrim to the result of this subtraction, the TX pulse width should be close to 1500us.



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

6 LONG	WING mode = delta dual ailerons
7 LONG	WING mode = vtail dual ailerons
8 LONG	WING mode = duckeron
9 LONG	WING mode = dual ailerons & dual elevators
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 flaperons, 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, it just has 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.

Stabilization modes

The AUX1 channel determines if the plane is in **RATE** mode or **HOLD** mode. These are the only two modes for Open FlightStab. A pulse width of 1500us on the AUX1 channel is **RATE** mode with a gain of 0 which is effectively **OFF.**

- **RATE Mode**
  - In this mode the device will stabilize in pitch, roll and yaw axis. The VR gain POTs on the RX3S (or the EEPROM setting) control the correction gain and direction for each of the three axis.
  - Note this mode has stabilization only
  - The PID “I” term is essentially zeroed out so the PID is not able to accumulate errors.
  - Default PID values for this mode is 500,0,500 for all three axis.
- **Attitude HOLD Mode**
  - Attitude HOLD allows the plane to fly with a fixed attitude (as long as there is sufficient power and control surface authority).
  - Attitude hold is short term as the gyros drift after a brief period. When the the sticks are moved to correct for any drift, the stick displacement alters the PID “I” term. This causes the accumulation of the error between the desired position and the current gyro reading over time and makes the correction more aggressive until the error is reduced to zero..
  - Default PID values for this mode is 500,500,500 for all three axis.

Trimming the aircraft

Therefore please be careful if you intend to trim your aircraft with stabilization on. If you are not conformable with, it is advisable not to use Attitude HOLD mode.

Reset Neutral Stick Position in air.

The INFLIGHT CALIBRATION option is on by default. This option allows the user to reset the neutral stick position. It may be disabled via Stick Configuration or the programming box.
INFLIGHT CALIBRATION is activated when 3 successive mode changes are detected with the time between mode changes being less that ½ second. If you start in **RATE** mode, then you must toggle **Hold**, **Rate**, **Hold** with less than ½ second between mode switches (less than 1.5 seconds for total sequence). If you start in **HOLD** mode, then you must toggle **RATE**,**HOLD**, **RATE** with less than ½ second between mode switches (less than 1.5 seconds for total sequence).

Mixer EPA Mode

There are 4 mixer EPA modes that controls the limits of all the servos.

Mixer EPA mode	Stabilizer servo output range
FULL (default)	1000-2000 us
NORMAL	1100-1900 us
TRACKING	start with 1250-1750 us, then track and never exceed RX input range
MAX	900-2100 us

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. Thus, if this feature has been enabled (default is disabled), 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 learned from the RX.

CPPM Mode

Some receivers support a mode known as Combined PPP (CPPM) which is a string of pulses lasting for the duration of one transmit frame. The time between pulses represents the stick position for a channel. This device supports CPPM mode, measuring the time for each channel by using these pulses.

CPPM reduces the number of connections to the RX and enables more channels (AUX2, THROTTLE and FLAP) on the CPPM enabled devices (for details please refer to the referenced links with the CPPM connection diagrams).

The RX3S V3 device has an integrated RX, which cannot be set to CPPM mode.

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. The throttle pass through is emitted out of RUD_IN pin.

The channel order can be changed by set using the programming box and changing the SERIALRX_ORDER parameter.

S.BUS Serial Mode, SRXL Serial Mode and Spektrum Serial Mode

These serial modes are much like CPPM in that only a single wire is required for all supported channels. The difference is that, unlike CPPM that sends time based pulses, these modes actually send bytes of data containing counts that representing channel pulse widths. Thus there is no measuring required, just decoding the counts. No chance for error, the counts used by the stabilizer are the exact counts sent from the transmitter.

Spektrum Serial and SRXL Serial sends these counts as normal negative active 115,200 Baud RS-232 data that is easily decoded by

stabilizer processor.

S.BUS, SRXL and Spektrum Serial send data too fast for most analog servos. The servo output frame rate can be set to a value between 0ms - 20m using the programming box SERVO_FRAME_RATE parameter. A value of 0 sets the output frame rate to equal the input data rate.

The channel order can be changed by set using the programming box and changing the SERIALRX_ORDER parameter.

The programming box SPEKTRUM_LEVELS parameter can be set to 1024 or 2048 to match the Transmitter resolution.

Mounting the device on your airframe

- **Normal Mounting**
  - (0 degrees) or upside down mounting (180 degrees)
  - This is the default mounting option. You don't need to change anything.
- **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.
  - Orientation can be set by stick configuration (or programming box) 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:
- Move the AIL stick **left or right** to change the config item and the ELE stick **up or down** to change the config value.
- 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).

Program Box Options and Stick Config Mode AIL/ELE toggles

The following table describes all the configurable parameters in the stabilizer. Some of them can be set up through the stick configuration mode described in the preceding section. All of them can be set up through the program box described later in the document (linked here).

CFG version 12		stick config ELE toggle	stick config AIL toggle
WING_MODE	set WING mix mode	1 ele	
DIPSW (default)	based on DIP switches		1 ail

RUDELE 2-AIL	override to RUD+ELE and 2-AIL		5 ail
DELTA 2-AIL	override to DELTA and 2-AIL		6 ail
VTAIL 2-AIL	override to VTAIL and 2-AIL		7 ail
DUCKERON	override to DUCKERON		8 ail
RUD 2-ELE 2-AIL	override to RUD+ 2-ELE and 2-AIL		9 ail
MIXER_EPA MODE	limit servo output range	2 ele	
FULL (default)	1000-2000 us		1 ail
NORM	1100-1900 us		2 ail
TRACK	start with 1250-1750 us and never exceed RX input		3 ail
MAX	900-2100		4 ail
SERVO_FRAME_RATE	maximum allowable servo frame rate (in 1/ms)	NA	NA
min = 0 max = 20	limit to 1/0ms (∞ Hz -- no max limit). use for digital servos limit to 1/20ms (50Hz). use for analog servos		
SERIALRX_ORDER	channel order with SERIALRX CPPM/Spektrum/SBUS	NA	NA
per-channel configurable	each channel in ch[12345678] refers to one of [RETA1a2F] eg. ch[RETA1a2F] for Frsky, ch[TAER1a2F] for Spektrum		
SPEKTRUM_LEVELS	spektrum serial 1024 or 2048 levels	NA	NA
1024	1024 levels (10 bits)		
2048 (default)	2048 levels (11 bits)		
MOUNT_ORIENT	mount device sideways on flat fuselage	3 ele	
NORMAL (default)	mount device flat “normally”		1 ail
ROLL_90_LEFT	roll device 90 deg left and mount on left side		2 ail
ROLL_90_RIGHT	roll device 90 deg right and mount on right side		3 ail
ROTATE_90_LEFT	rotate normally mounted device left 90 degrees		4 ail
ROTATE_90_RIGHT	rotate normally mounted device right 90 degrees		5 ail
STICK_GAIN THROW	stick-position gain blends from max (1.0) to zero, over	4 ele	
FULL (default)	full stick range		1 ail
HALF	half stick range (ie. no correction from ½ stick onwards)		2 ail
QUARTER	quarter stick range		3 ail
MAX STICK_ROTATE	set stick-controlled rotation rate at full stick	5 ele	
VERY_LOW	0.25x		1 ail
LOW	0.5x		2 ail
MEDIUM (default)	1.0x		3 ail
HIGH	2.0x		4 ail
RATE STICK_ROTATE	allow stick-controlled rotation rate in RATE mode	6 ele	
DISABLE (default)	disabled		1 ail
ENABLE	enabled		2 ail
INFLIGHT CALIBRATE	toggle RATE/HOLD 3x within 0.5s to calibrate	7 ele	
DISABLE	disable inflight RX calibration		1 ail
ENABLE (default)	enable inflight RX calibration		2 ail
VR_GAIN		NA	NA
AIL/ELE/RUD	“POT” = use device POT setting (default) -127 to 127 = override POT setting		
RATE_PID	per-axis P, I, D parameters	NA	NA
Ail P/I/D	ROLL axis P/I/D in RATE mode (500/0/500 default)		
Ele P/I/D	PITCH axis P/I/D in RATE mode		

Ail P/I/D	ROLL axis P/I/D in HOLD mode (500/500/500 default)		
Ele P/I/D	PITCH axis P/I/D in HOLD mode (500/500/500 default)		
Rud P/I/D	YAW axis P/I/D in HOLD mode (500/500/500 default)		
LOW_PASS_FILTER	set gyro Digital Low Pass Filter Values	NA	NA
Highest	256Hz		
	188Hz		
	98Hz		
	42Hz		
	20Hz		
	10Hz		
Lowest (default)	5Hz		
EEPROM	EEPROM action	8 ele	
Update Cfg	Write Config to EEPROM		2 ail
Erase Cfg	Erase Config in EEPROM to Default		NA
Erase Stats	Erase 1/2/R Stats in EEPROM to 0/0/0		NA

- NA in stick config AIL toggle column means accessible by program box only

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 & RX3SM	AILL_OUT and ELE_OUT
NanoWii, MINI MWC & Flip MWC 1.5	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.

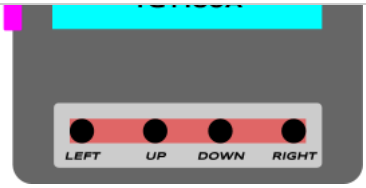
Normally, the plane will wiggle the surfaces 3 times to indicate that calibration has completed. If the EEPROM was reset to factory default (either on first boot or on jumper reset), the plane will wiggle the surfaces 9 times instead. **This can be used to indicate a problem with EEPROM corruption when you do not expect the EEPROM to be reset.**

Programming Box Support - 16K (Recommended)

(link)
FlightStab also supports configuration through the TGY160A programming box. Using the programming box is more convenient and will allow more options than both the 8K Programming box or 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_TGY160A.hex** bootloader
- Configure the programmer for 5V operation
- Use the AVRootloader to update the main programming box firmware:
  - **201xxxxx_TGY160A.hex** for the TGY160A box
- Connect the programming box to the FlightStab device through the “one-wire” servo connector and power them up.



Flashing the TGY160A with AVRrootloader

On the TGY160A, 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.

- The TGY160A processor runs at 5V so the USBasp should be set for 5V operation and it is safe to power the programming box from the USBasp.
- Select the target chip type as **ATMega168PA** chip 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	**TGY160A**
LOW	0xE6 or 0xEE
HIGH	0xDD
EXT	0xF8
LOCK	**0xCC**

DO NOT PROCEED if the values are unexpected.

- Set the Programmer to 5V
- Erase the chip. This should also reset the LOCK fuse for to 0xFF.
- Set the Fuse Bits as follows:
  - Low = 0xF7
  - High = 0xDF
  - Ext = 0xF9 without AVRootloader
  - If you **do not plan** to install the AVRootloader in the programming box then install the proper firmware, close up the box and you are done.. The TGY160A firmware is at **.\Builds\201xxxxx_TGY160A.hex**.

- To install the AVRootloader continue with the steps that follow

- Change the EXT fuse from 0xF9 to **0xFC**. This will enable the chip to boot a 512-byte (256-word) bootloader.

- Flash the AVRootloader bootloader. Use the USBASP ISP programmer to flash the **AVRootloader_TGY160A.hex** bootloader

- Use the AVRootloader instructions to flash the hex file at **.\Builds\201xxxxx_TGY160A.hex**

- Close up the box.

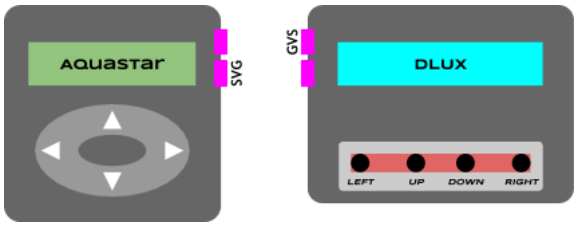
Programming Box Support - 8K (16K Preferred)

(Aquastar link)
(dlux link)
FlightStab supports configuration through the Aquastar or DLUX programming boxes. Using the programming box is more convenient and will allow more options that 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

● Connect the programming box to the FlightStab device through the “one-wire” servo connector and power them up.



Flashing the Aquastar/DLUX with AVRrootloader

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.

● The Aquastar/DLUX processor runs at 5V so the USBasp should be set for 5V operation and it is safe to power the programming box from the USBasp.

● 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
EXT	not used by ATMega8A	not used by ATMega8A
LOCK	**0xFF**	**0xFC or 0xC0**

On the Aquastar, the LOCK fuses are may not be set (0xFF). If this is the case 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 to 0xFF.
  ○ If you **do not plan** to install the AVRootloader in the programming box then install the proper firmware, close up the box and you are done.. The Aquastar firmware is at **.\Builds\201xxxxx_AQUASTAR.hex** and the DLUX firmware is at **.\Builds\201xxxxx_DLUX.hex**.
  ○ To install the AVRootloader continue with the steps that follow

● 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 AVRootloader bootloader. The Aquastar bootloader hex file is at **.\AVRootloader\AVR\default\AVRootloader_AQUASTAR.hex** and the DLUX bootloader hex file at **.\AVRootloader\AVR\default\AVRootloader_DLUX.hex.**
NOTE - Both files are actually the same, just with proper device names to avoid confusion.

● Close up the box.

Using the Aquastar/DLUX/TGY160A Programming Box

Connect the programming box to the FlightStab device. The RX3S devices uses AILR_OUT, the RX3SM device uses RUD_OUT and the MINI MWC and Flip 1.5 devices use A0 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.

Tips

General guidelines for adjusting the PID gains

With Open flight stab it is suggested to use knobs to control for overall Master Gain, per-axis POT Gains and individual P, I, D parameters. For most cases, adjusting the master gain and the per-axis POT are usually sufficient for the majority of airframes tested. Master Gain is for the overall flight condition (when speed goes up or prop wash increases over control surfaces). Per-axis POT lets you control the relative correction for the pitch, roll and yaw axes (usually roll and pitch have to be turned down, and yaw turned up).

General guideline is:

- increase Master/Pot Gain if stabilization feels sluggish

- reduce Master/Pot Gain if it feels too sensitive and the aircraft start to oscillate

The suggested method of adjusting Pot Gains is to set them at a low setting and gradually increase Master Gain until the latter reaches maximum. If the correction does not cause undulations increase Pot Gain and start again until the correction do cause undulations. Then slightly reduce Pot Gain on the offending axis to allow a safety margin. The aim is to use 100% Master Gain with Pot Gains on all axis adjusted to provide correction to your airframe with no undulations.

However sometimes one might want to adjust individual P, I, D parameters because the Pot Gain adjustment is too sensitive for a particular axis (for example roll axis on 3D airframes with large control surfaces).

Individual P, I, D parameters are roughly as follows:

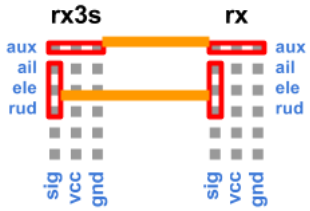
P = Proportional is the major component and is similar to the master gain and POT gain in its effect. Higher P means more aggressive stabilization.

I = Integral and is used only in HOLD to correct any attitude drifts. This parameter accumulates errors in the heading and applies the opposite action, thus achieving "heading hold". Higher I mean stronger tendency to restore the heading.

effect but too high is bad.

Wiring Connections

Receivers are usually built so that the “vcc” and “gnd” line are common on all channels. Some receiver manufacturers utilise an edging on their products (like futaba, spektrum etc) that prevent you from inserting the servo plug the wrong way round. Depending if your receiver will allow a plug to be positioned along the signal line (for example FrSky D8R-XP) the proposed wiring configuration below will save you some wiring and is much neater. You can rearrange the male to male wires if the RX3S channel order does not match your receiver.



Adjusting the PID gains to have two RATE modes

One can use the configuration programming box to change PID values on the gyro. A very simple representation of Flightstab code implements what is an effectively a PID controller that works individually on all axes. Generally the P gain affects stabilization correction and the I gain affects the Attitude HOLD. Therefore if one changes the I gain in Attitude HOLD one can effectively have two rate modes. However the two modes will be different; this is because of the RATE STICK_ROTATE parameter. This parameter is always enabled (and cannot be disabled) in Attitude HOLD but it is configurable and can be enabled or disabled in RATE mode. By default it is disabled in RATE mode

The effect of RATE STICK_ROTATE is that the gyro will take your stick demand from the tx and will stabilize based on that position. Obviously if you use expo then the stick position is not a linear function anymore. For example if you don’t have expo (so stick position and servo output from your tx is linear) and you have up elevator half way, the gyro will stabilize the elevator servo on that position. The control of the aircraft is then effectively "fly by wire" because your tx demands are fed into a yellow box and the output is purely function of the PID gains, The effect of this while flying can be described as if you use expo.

What happens if there is a Power cycle in the air?

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 stable 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.

Programming your transmitter for adjusting the Master gain

Spektrum DX8

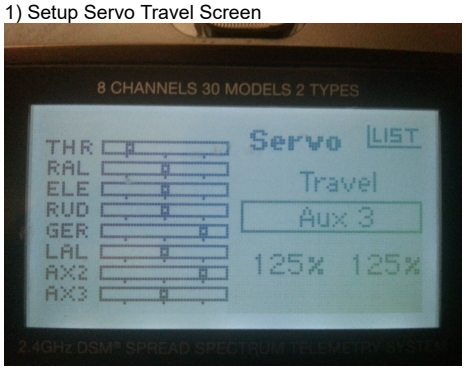
The following example describes how to retain AUX1 for Flaps and use the Flight Mode switch and Knob for gain control on channel 8 (AUX3). Channel 8 from the RX must be connected to AUX1 on the RX3S.

Note - this example is for use with standalone flight controllers. To use this example with an RX3S V3 (integrated RX) flight controller which require the gain control to be on channel 6 (AUX1) replace all references to **AUX3** with **AUX1**.

Using the FM three point switch and the Knob on DX8 for RX3S Rate/Off/Hold Control
The following describes how to configure the FM switch and the knob

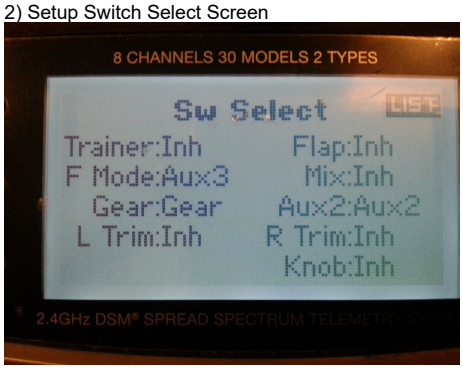
changed using the knob and the position of the knob sets the same value for master gain for both **Rate** and **Hold** mode. However at the moment the knob is operating -100% to 0%. The knob at position -100% sets the Master gain value to 0% whereas the knob at position 0% sets the Master gain value to 125%. The knob range from 0 to 100% sets the Master gain value to 125%.

1) Setup Servo Travel Screen



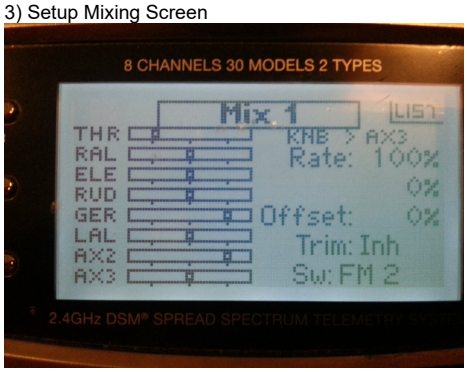
(Travel for Aux 3 has to be symmetrically extended to 125% on both servo travel ends)

2) Setup Switch Select Screen

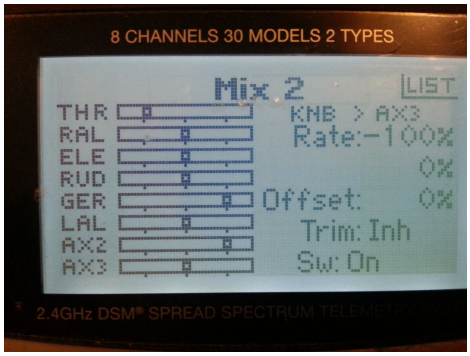


(set F Mode : Aux3, the other switches depend on your model setup)

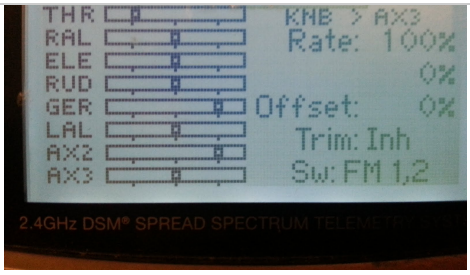
3) Setup Mixing Screen



(setup Mix 1)



(setup Mix 2)

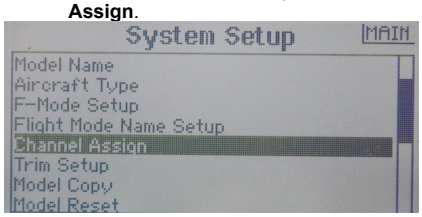


(setup Mix 3)

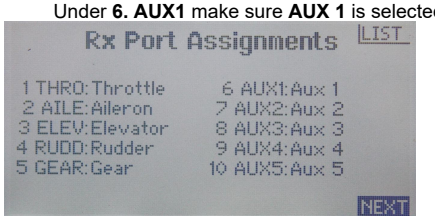
Spektrum DX18

Using Left Paddle on DX18 for RX3S Rate/Off/Hold Control
The following describes how to configure the Left Paddle on a DX18 to be used for RX3S Gain Control. The instructions provided will place the RX3S in the Off position when the paddle is centered. Full paddle down is 100% **Rate** mode and Full paddle up is 100% **Hold** mode. The gain for both **Rate** and **Hold** mode is variable from 0-100% with paddle center being 0%

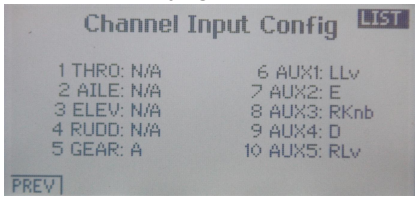
1. Place Transmitter in System Mode and select **Channel Assign**.



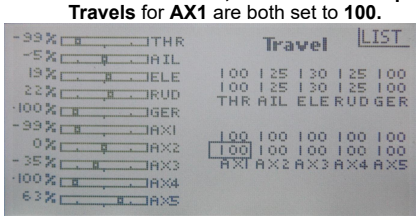
Under **6. AUX1** make sure **AUX 1** is selected7



2. Then select **NEXT** to view **Channel Input Config** and make sure **6 AUX1:** is set to **LLv**



3. Save and exit **System Mode.**
4. In **Model Mode,** select **Servo Setup** and verify that the **Travels** for **AX1** are both set to **100.**



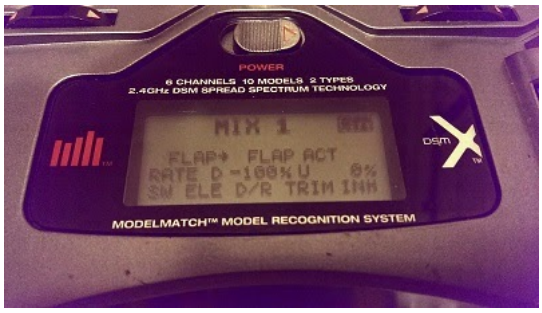
Spektrum *DX6i or OrangeRX T-Six transmitter (using pseudo' 3-Position Switch)*

Using FLAP/GYRO switch as 'master' on/off switch; when ON it allows the ELEV D/R switch to select the stabilisation mode for RX3S Rate/Hold Control. You can't implement “in flight variable Master Gain Control” because there is no rotary knob.
The DX6i sadly lacks a 3-Position switch and a rotary knob, hence it is difficult to fully utilise RX3S Gain control and access all three modes - Heading Hold, Off and Rate mode currently available in Open Flight Stabilizer. However one can accomplish this using two switches and a bit of mixing creating a 'pseudo' 3-Position Switch. Note this guide also applies to the OrangeRX T-Six transmitter.

Setup settings as follows:

(NOTE: You may need to experiment with the NORM< v7% figure and adjust until there is no servo corrective movement when the switch is in the OFF position. This may be different on individual transmitters.)

In the MIX 1 menu, activate the mix and set:
FLAP> FLAP ACT
RATE D -100% U 0%
SW ELE D/R TRIM INH



In the MIX 2 menu, activate the mix and set:
FLAP> FLAP ACT
RATE D -100% U 0%
SW ELE D/R TRIM INH

The operation of the switches and modes activated are as follows:

FLAP/GYRO switch OFF = Stabilisation is off.

FLAP/GYRO switch ON, ELEV D/R switch OFF = Heading Hold mode.

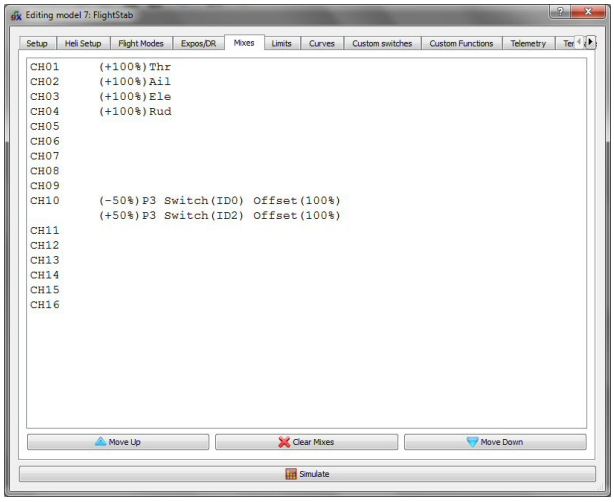
FLAP/GYRO switch ON, ELEV D/R switch ON = Rate mode.

Summarizing in this setup we use the FLAP/GYRO switch as 'master' on/off switch; when ON it allows the ELEV D/R switch to select the stabilisation mode for RX3S Rate/Hold Control. When the FLAP/GYRO switch is OFF, the ELEV D/R switch has no effect on the stabiliser.

Note, The suggested method of adjusting Pot Gains is to set them at a low setting and gradually increase Master Gain until the latter reaches maximum. If the correction does not cause undulations increase Pot Gain and start again until the correction do cause undulations. Then slightly reduce Pot Gain on the offending axis to allow a safety margin. The aim is to use 100% Master Gain with Pot Gains on all axis adjusted to provide correction to your airframe with no undulations.
However using a DX6i you can't implement “in flight variable Master Gain Control” because there is no rotary knob on the DX6i. If the Pot Gain adjustment process becomes difficult one can quickly adjust Master Gain by adjusting the RATE % settings in the two MIX menus. To simulate a gradual increase of Master Gain one can set the RATE % settings low to begin with, and with trial and error gradually increase until the corrections cause undulations. Then reduce pot gain on the offending axis and start again.

The following describes how to configure the ID switch and P3 knob on a Turnigy 9X/9XR/9XR Pro transmitter to be used for RX3S Gain Control using OpenTx. The mix for a FrSky Taranis is very similar, all you have to do is change the switch and knob source according to your transmitter). This example is a screenshot using companion9x and implements Master Gain Control on channel 10, however you may use any channel of your choice (normally on a PPM mode receiver like a FrSky D8R-XP Receiver this would be channels 1:8).

The instructions provided will place the RX3S in the **off** position when the ID switch is centered (ID1 region) or when the knob is set to 0% (1500 us). ID0 range is operating within **Rate** mode and ID2 range is operating within **Hold** mode.
The Master gain for both **Rate** and **Hold** mode is variable and can be changed using the knob and the position of the knob sets the same value for master gain for both **Rate** and **Hold** mode. OpenTX/Er9X channel/servo range limits is 100%, the knob is operating within -100% to 100%.



FrSky Taranis

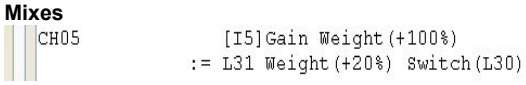
Using Taranis Special Functions and Logical Switches to generate InFlight Calibration sequences

RCGroups user **LEXM** has developed some InFlight Calibration sequences for the Taranis.
He has published these in a document that can be found here.

I like to use the left slider for gain control on channel 5 with center being 0 gain (effectively Off), below center increasing Rate mode gain and above center increasing Hold mode gain. I have used the information provided by LEXM to construct a Taranis InFlight Calibration sequence using the spring loaded Taranis switch (**SH**) for activation.

The setup I use is below:

Mixes



Special Functions

Logical Switches

The above sequence will generate 4 mode changes within about a 600ms period when switch **SH** is activated.

Warning - Make sure that the stabilized controls are all at their center position before activation. If the controls are not centered when activated the firmware will used the current position as center, not

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 cmake (recommended)

- Install CMake 2.8 (http://www.cmake.org/cmake/resources/software.html) in: C:\Program Files (x86)\CMake 2.8
- The installer should automatically add the following path in your environment setting but do ensure PATH string includes C:\Program Files (x86)\CMake 2.8\bin
- You can use arduino 1.0.5 or 1.0.3. If you chose 1.0.5 that all references to 1.0.3 should be changed to 1.0.5 in the following section of the document.
- install/extract arduino 1.0.3 (http://arduino.cc/en/main/software) in: "C:\Program Files\arduino-1.0.3" (**this exact path is important!)**
- Ensure PATH now includes C:\Program Files\arduino-1.0.3\hardware\tools\avr\utils\bin the path string should look include paths for both cmake 2.8 and arduino 1.0.3:
- C:\Program Files\arduino-1.0.3\hardware\tools\avr\utils\bin;C:\Program Files (x86)\CMake 2.8\bin
- **NOTE there are no spaces between the semicolon character**
- now go to
- .\FlightStab
- and run the following commands (or compile them into a batch file). The text below in red are comments and should not be included in your batch file::

```
cd ./Builds
rd /s .
(this step wipes out all files within the current
directory. The Builds directory should be empty
for the following commands to work)
cmake [-DTODAY=yyyymmdd] -G"MSYS Makefiles" ..
(-DTODAY is optional to override the system date.
Do not enter the [ ] for example use either
cmake -G"MSYS Makefiles" ..
cmake -DTODAY=20130101 -G"MSYS Makefiles" ..
make [VERBOSE=1]
again do not use [ ] use either
make
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.

Building the firmware via arduino IDE

NOTE that this method cannot be used with the current source because the generated image won't fit on the CPU. This method is kept for reference.

- 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

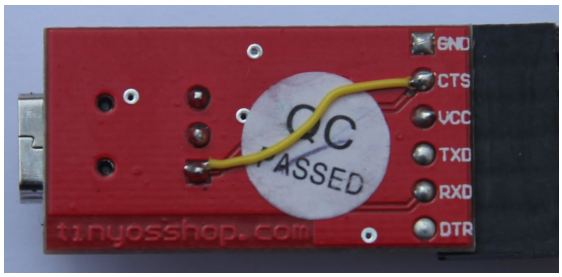
```
C:\users\<user>\My
Documents\Arduino\Libraries\_Stub
C:\users\<user>\My
Documents\Arduino\Libraries\I2Cdev
C:\users\<user>\My
Documents\Arduino\Libraries\ITG3200
...
```

- run Arduino.exe and select Board as `Arduino Pro or Pro Mini (5V, 16MHz) w/ ATmega168`
- 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

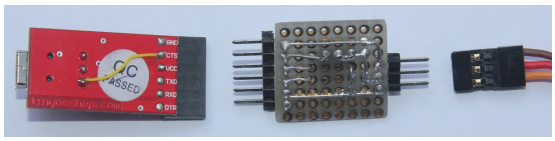
TinyOS FTDI Programmer Modifications

Modifications to supply 5V for the RX3S and 3.3V for the FTDI chip and RX3S signal pin.

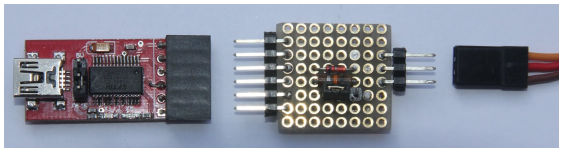
Modification to the programmer is not difficult. It only requires 1 trace on the programmer to be cut and one wire added. This change places 5V from the USB port to the pad that was previously used for CTS (CTS is not needed in this application). Photos below show where to cut the trace and add the wire.



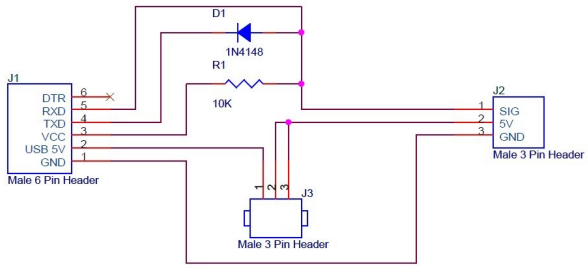
I also constructed an adapter to connect the programmer to an RX3S. Photos of the adapter as well as a schematic are also attached.



Bottom View



Top View



Schematic

The jumper on the TinyOS board should be connected in the 3.3V position.

The adapter board also has a jumper to enable 5V power for the RX3S when a jumper is installed between pins 1 & 2. When a powered RX3S is being programmed, place this jumper on pins 2 & 3.

FlightStab Guide V3.1

Интервал автоматического обновления в минутах: 5