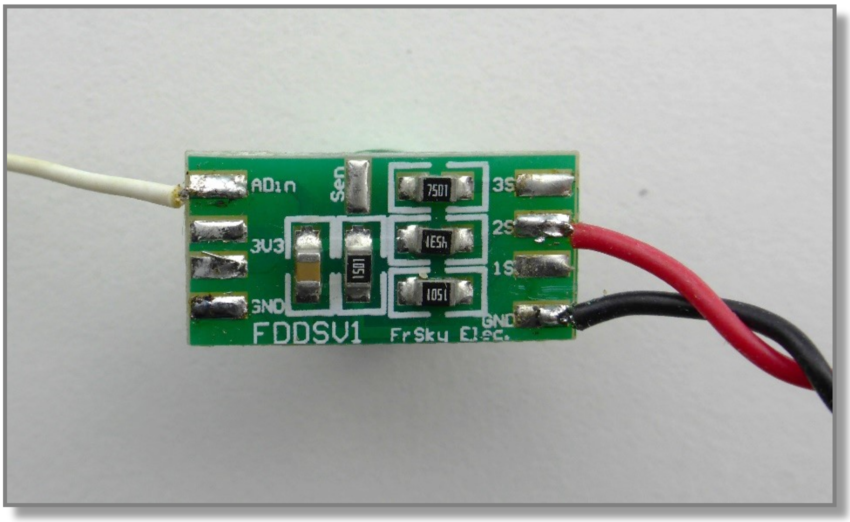
How To ... Part 2

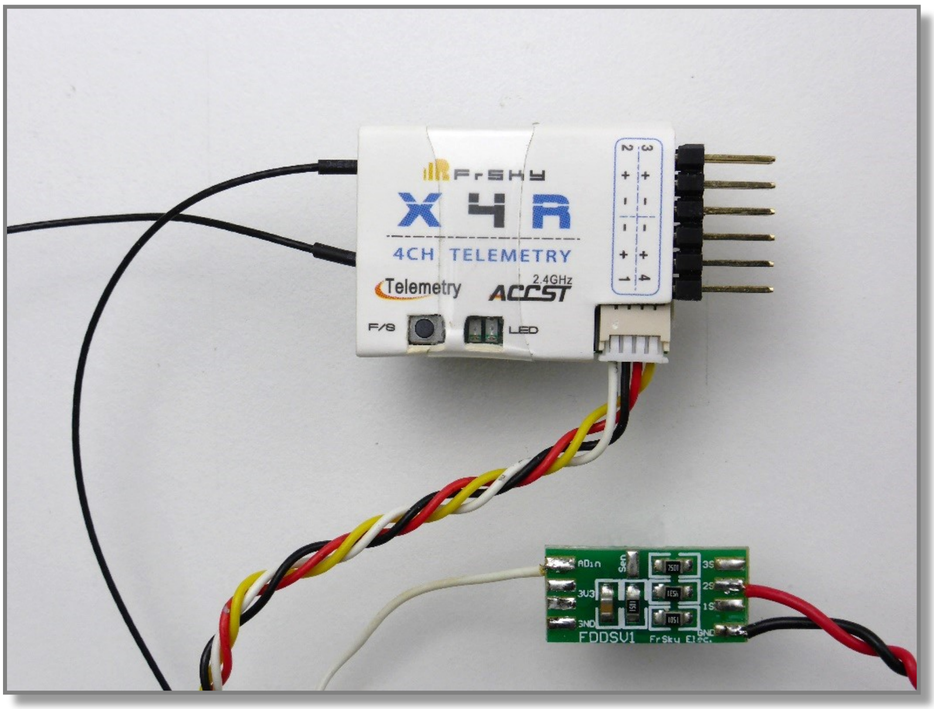
Contents

How to Add Voltage Telemetry to the X4R

The X4R is a small lightweight receiver with 3 or 4 channels depending on the model designed for small light models. Often in such planes, space is at a premium and it is difficult to fit any telemetry in. With small foamy electric models, it would be very useful to have at least voltage telemetry to give an indication of battery voltage. This could be very useful with powered gliders where there is a mixture of powered flight and gliding, and using flight times alone is not appropriate. The X4R receivers come Smart Port enabled so that the standard voltage sensor could be used. However, this is nearly the same size as the receiver itself. Another feature of these receivers is, unlike the other X series receivers, there is an analogue input, thus some of the older, pre-Smart Port, sensors can be connected. One is the **FrSky FBVS-01** battery voltage sensor. This tiny sensor is ideal for measuring flight battery voltage and is very inexpensive, although it is not always available from the main suppliers. However, a quick internet search will soon pull up a few places that sell it.



Fitting the sensor is quite straight forward. With each X4R comes a 4 way plug and lead to fit the smart port socket. Three wires are for the smart port and consist of a red positive, a black negative, a green or yellow smart port lead, and a white wire not connected at the other end. I firstly fit a normal servo connector onto the red, white and green/yellow lead to have a smart port plug just in case I want to fit other sensors or update the firmware.



First, remove the heat shrink from the sensor. Don't worry, a spare piece of heat shrink is provided. Then unsolder the red and black from the **GND** and **ADin** pads. Then solder the white lead onto the **ADin** pad at the top left-hand corner. The red and black leads soldered to **GND** and 3S can be left as they are. Only one negative lead is required as the two **GND** pads are linked on the circuit board. If using a 3S battery connect the red wire to the 2S terminal instead to get a somewhat more accurate reading. Don't worry, I cannot understand the logic either. Then the red and black leads are soldered to the power connector going to the speed controller.

Power up the transmitter and receiver and set **OpenTX** to search for new sensors. Now **A2** should come up as a new sensor and show a 13.2 volt ratio. Check the actual battery voltage and then adjust the offset until the telemetry reads the same as the actual battery voltage. Finally, once everything is working, replace the heatshrink with the spare length of clear heatshrink provided.

The X4R receiver analogue input has a maximum input of 3.3 volts. The sensor has three terminals:

1S terminal which measures voltages from 0-6.6v
2S terminal which measures voltages from 0-13.2v
3S terminal which measures voltages from 0-19.8v

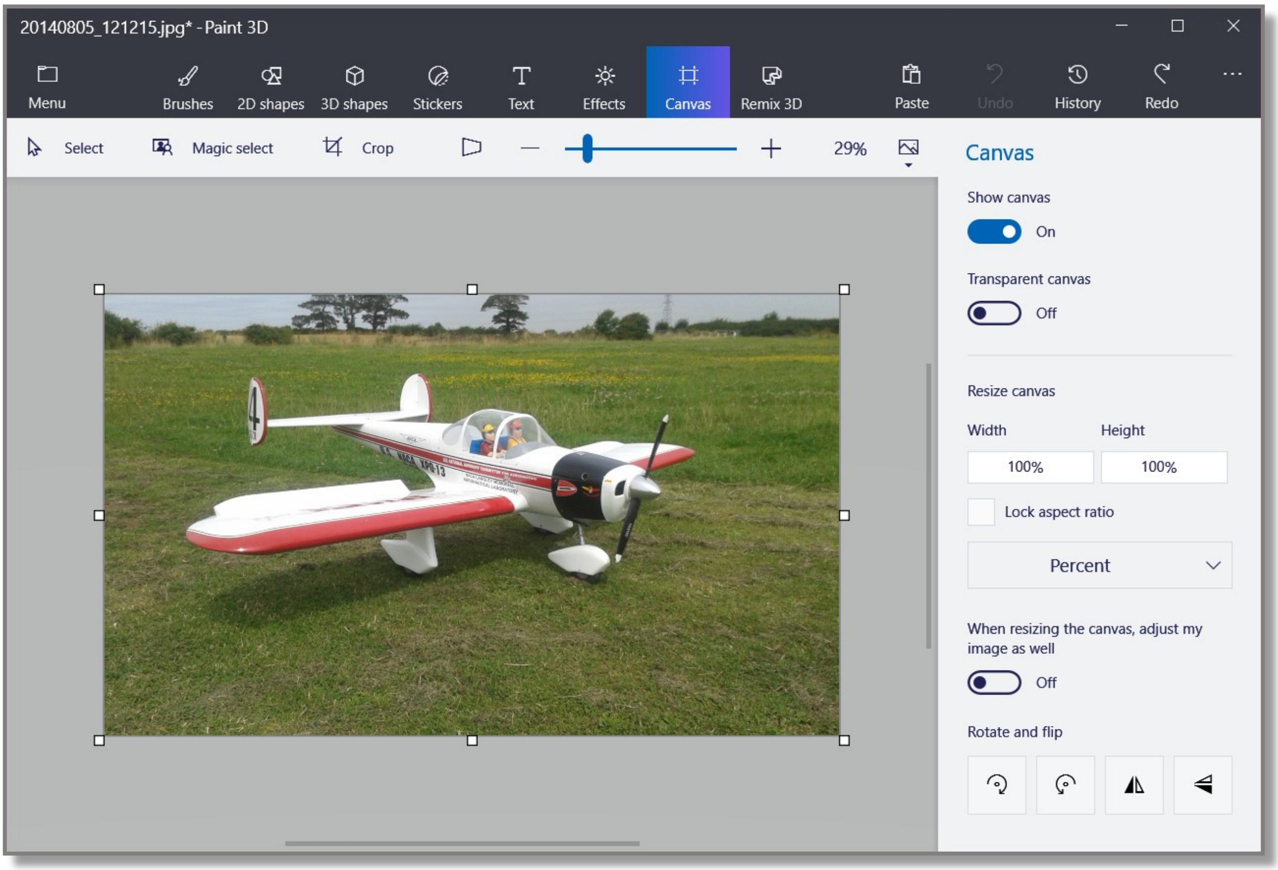
If using a different battery or using the 1S or 3S terminals, then the ratio on the **OpenTX** telemetry page will need adjusting to match the above maximum voltage.

How to Create Model Pictures for the Horus

There is a good collection of Horus model image files at this website:

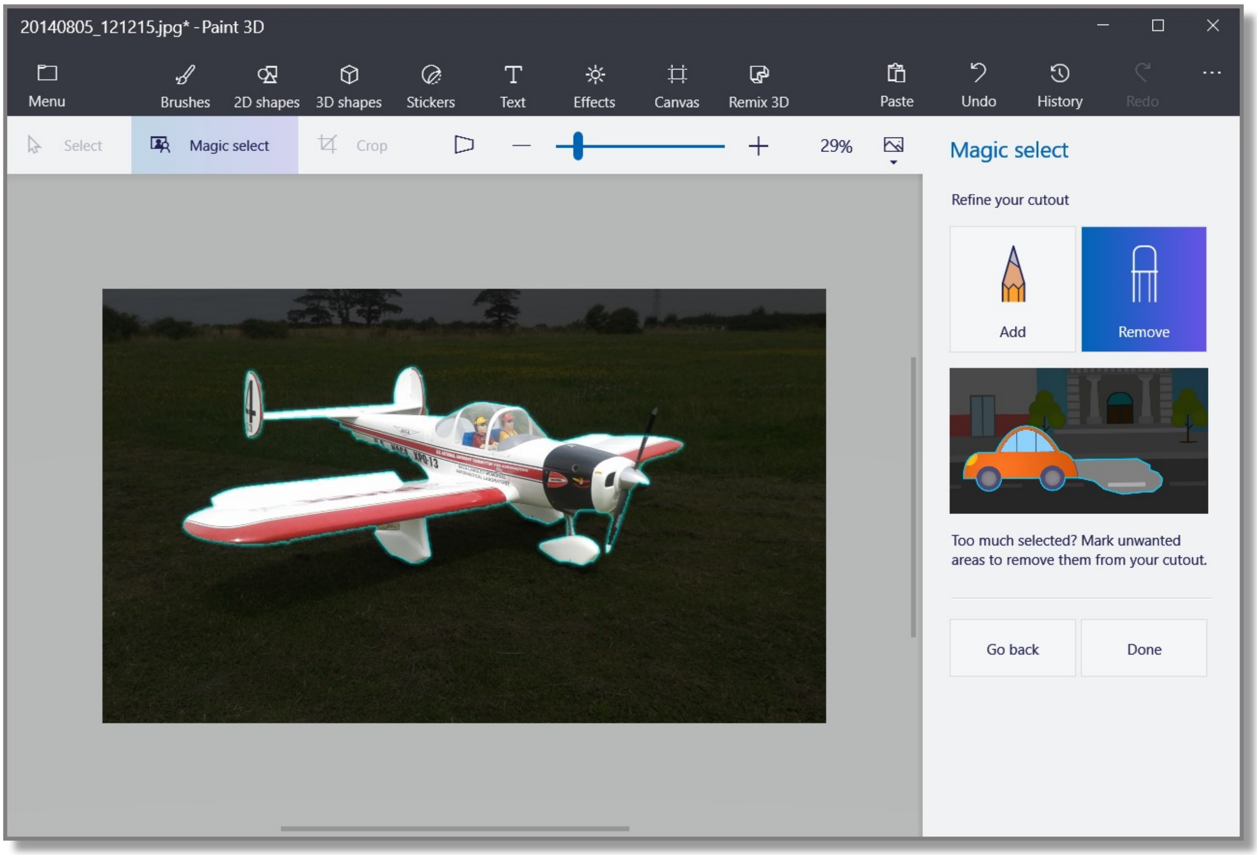
https://skyraccoon.com/icons

For one's own unique models, it is useful to create one's own images for the transmitter. A good drawing program is needed to be able to modify photographs or pictures to make them suitable for the Horus. Perhaps the best for this need is a program called **Paint 3D** from Microsoft. This free program is ideal for this sort of work, and is the program we will be using here. For a standard model, the advertising picture can be used and adapted. For one's own models, it is possible to use a photograph:

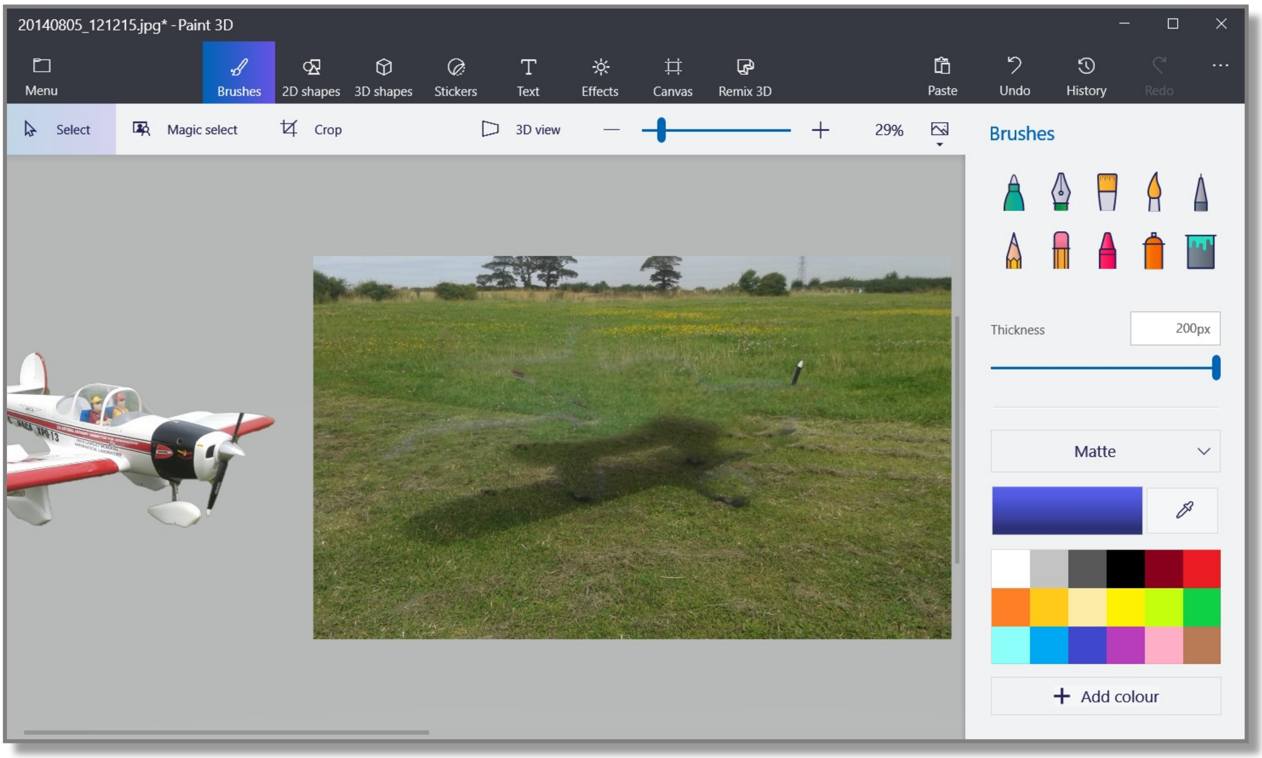


As can be seen, this is a good photograph of the model, however the background is not so good!

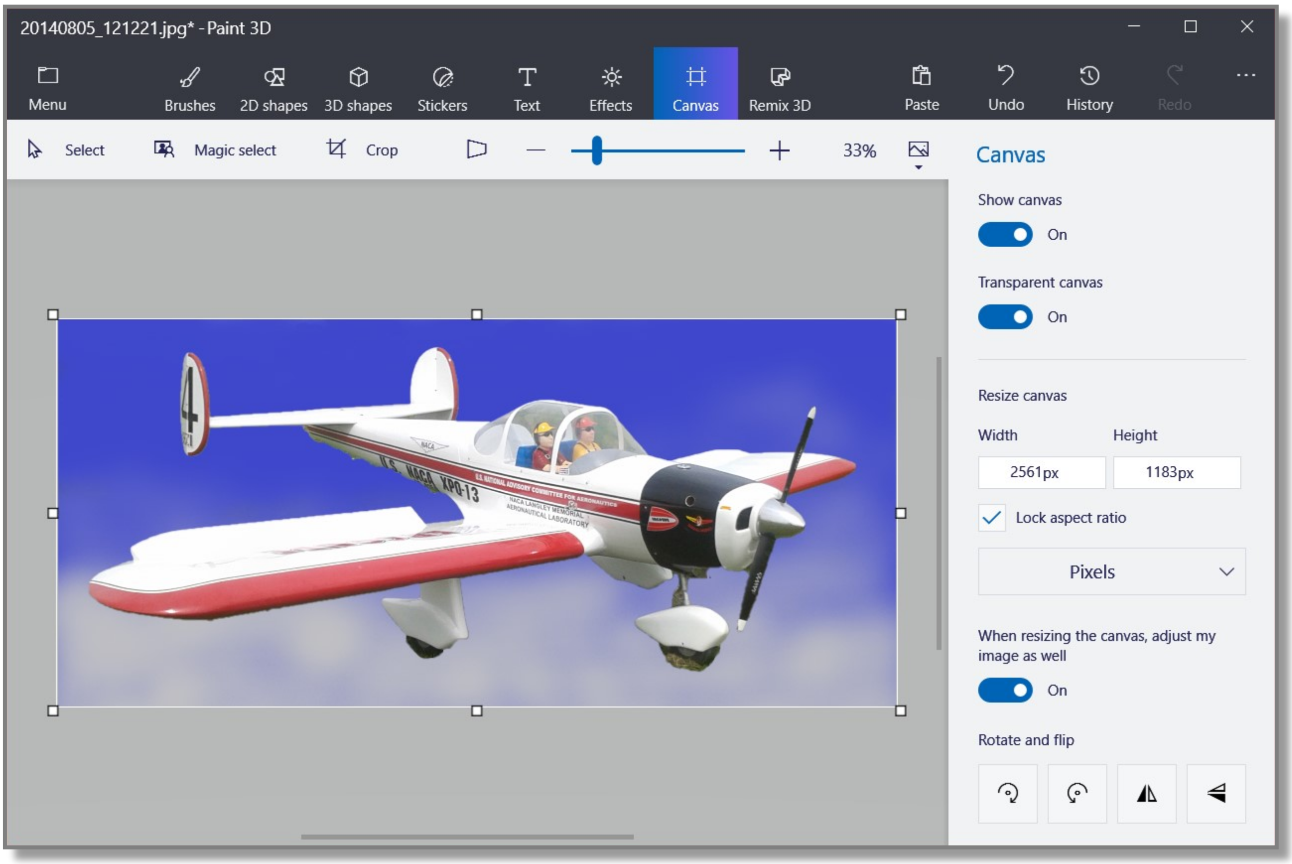
Use the **Magic Select** function show on the toolbar and use the **Add** and **Remove** options to tidy up the cutout and click **Done** when ready.



Click on the model to be cut out, and slide it out of the way.

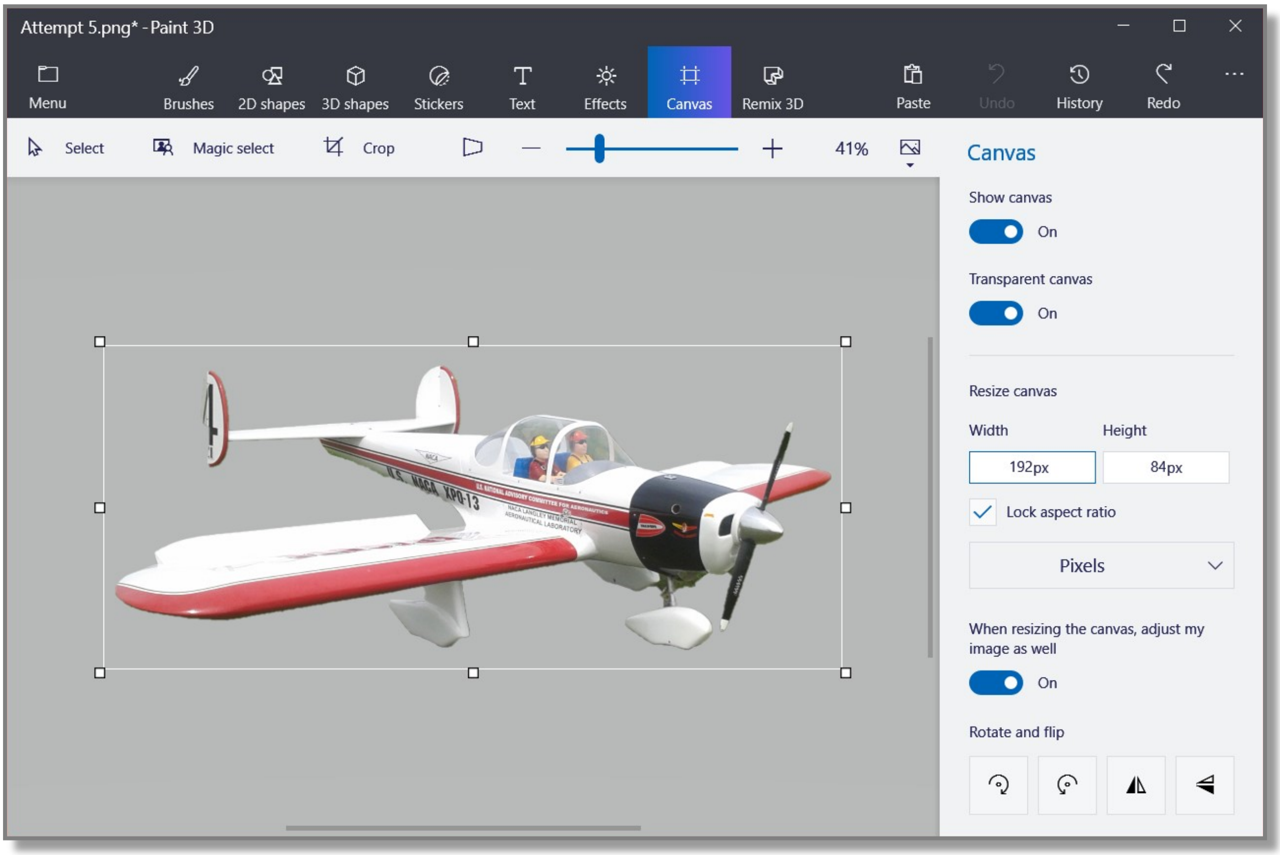


Now recolour the background to suit the model. This can be done with any of the brush tools. As this example model is white, a blue background has been created, and then a light grey colour has been sprayed over part to give a sky effect.



Then slide the cutout back over the background. Finally export the picture as a **png** or **jpg** file. Now reload the saved picture and resize for the Horus screen. This should be a maximum of 192 pixels by 116 pixels. For some reason this only works if the resizing is done after being reloaded. First lock the aspect ratio, then adjust either the width or the height to get the largest picture. Finally resave, remembering to make the filename no more than 6 characters long.

If the above screenshot is compared to the one on the previous page, it can be seen that this image has a transparent background. This is quite easy to achieve with **Paint 3D**. Once the **Magic Select** has been used to cut out the model, slide it out of the way and use the rubber to erase the background. Then restore the model back onto the background and **Export**. Next reload it, crop to size and save as a reduced size file ensuring the **Transparent Canvas** is enabled:



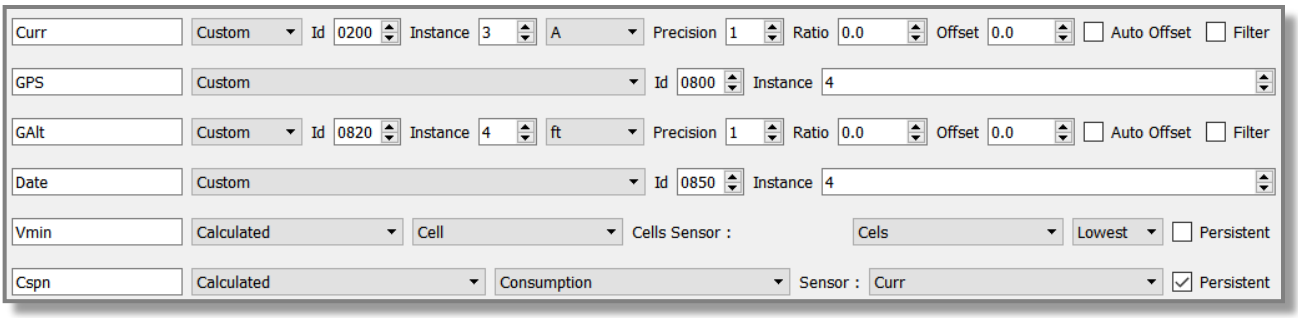
How to Create a LiPo Fuel Gauge

This is a useful procedure to display a telemetry setting as a percentage. It was devised by Mike Naylor (Miami Mike) and is reproduced here with his permission.

This is a "fuel gauge" setup for **OpenTX** to display the flight battery's state of charge as a percentage, beginning at 100% for a fully-charged pack and decreasing as the battery consumes power. It requires a current sensor, such as a **FrSky SP-40A - Smart Port 40 Amp Sensor** or **FrSky SP-150A - Smart Port 150 Amp Sensor**.

It allows alarms and voice announcements to be added that trigger at specific charge states, or at regular time intervals, or each time the charge state decreases by a certain amount so that there will be announcements at, for example, 100%, 90%, 80%, etc. Announcements can also be brought up at a flip of a switch. The battery state will be retained between flights even if the radio is turned off, so a switch is provided to reset the charge state to 100% when the battery is replaced or recharged.

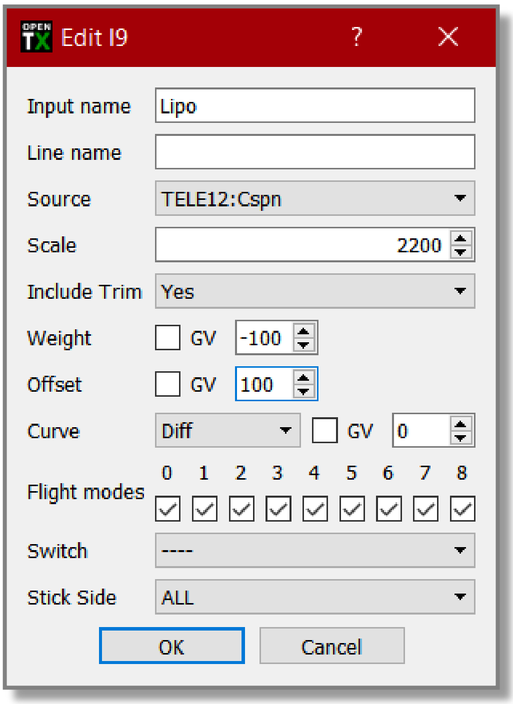
This idea could also be adapted to give percentage readings of other telemetry values.



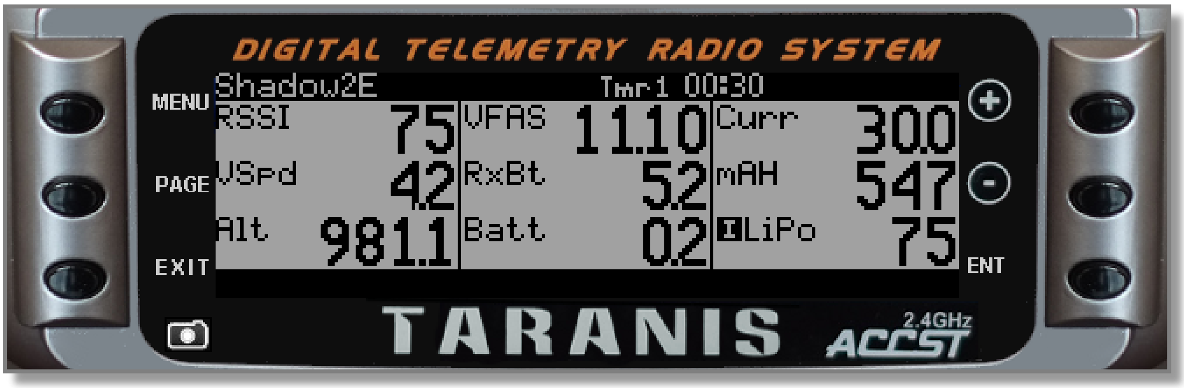
First the current sensor needs to be discovered, and then a new sensor added to calculate the consumption. Here it is called **Cspn**. Notice the **Persistent** box is checked to retain the value if the transmitter is switched off.

Now for the clever bit. A new input is created on the **Inputs** page. Use a channel above the normal channels used for a model. In the example below, channel 9 is used.

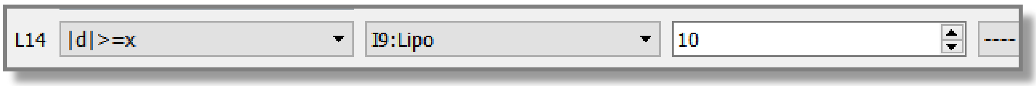
The **Weight** is set to -100 and the **Offset** is set to 100 to give a range of 0% to 100%. The Scale is set here to 2200, which is the capacity of the battery. In the scale setting, the capacity of the battery is entered to give us an accurate percentage for the battery being used. One could decide that 0% is when the battery is fully discharged and then only fly whilst the battery is above 20% say. On the other hand one could set this figure to 1760 so that 0% would now represent the minimum flying time, and still leave the battery 20% charged.



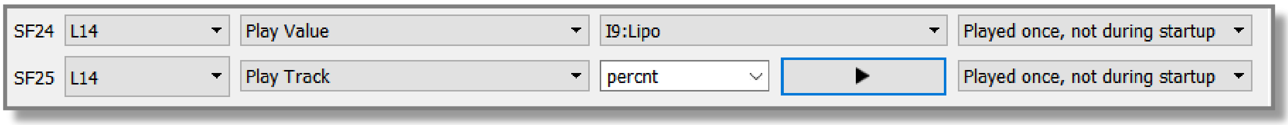
On the telemetry page of the transmitter, an input can be selected as well as the more normal telemetry figures. Thus the percentage remaining of the battery charge can be displayed. Here the value of 75 is displayed in the bottom right hand corner of the screen.



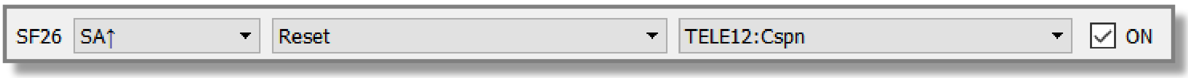
To add an automatic call down every 10%, first create a **Logical Switch**:



Then add two **Special Functions** to enable the announcement.



There is no percent sound in the user sound files, however, a percent is available in the **SYSTEM** sub folder of the **SOUNDS** folder on the SD Card. It is called **Percent0**. This needs copying across to the main **SOUNDS** folder and the name needs shortening to just 6 characters. While generally the **Special Functions** can be entered in any order, in this example, as the routine uses the same **Logical Switch**, the one that appears first on the list will be executed first. A switch could also be set up to reset the **Cspn** value:



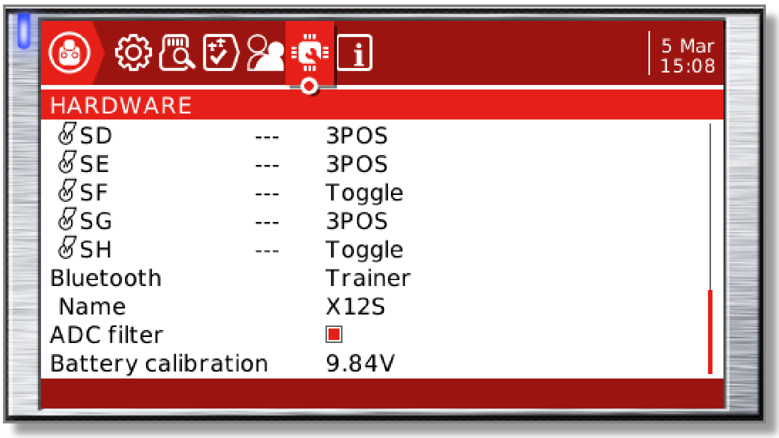
If the battery is normally changed each flight, then this switch would not be needed, and the **Persistent** box should be unchecked on the **Telemetry** page.

No doubt readers will be able to think up all sorts of uses for this routine. The clever bit is using an **Input** to store a particular value in the **Scale** parameter and use the **Weight** and **Offset** to give a range of 100.

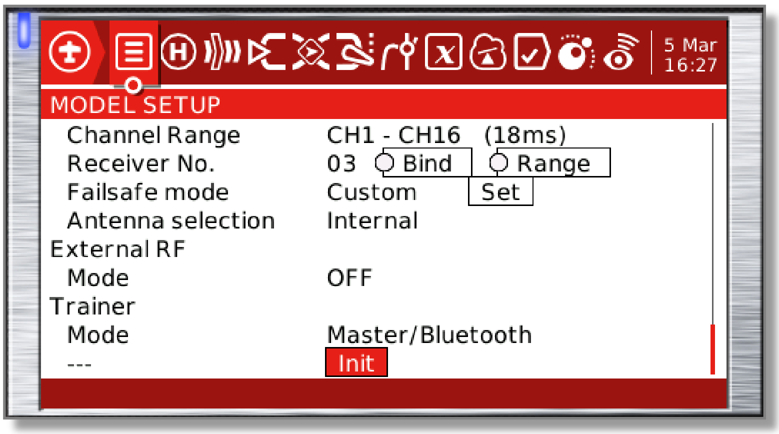
How to Use the Bluetooth Trainer Function

The Horus X12S and the Horus X10 and X10S transmitters all have Bluetooth facilities which can be used for telemetry and trainer functions. This **How To** shows how to set up the Bluetooth to work with the trainer function.

Setting up the Bluetooth trainer must be done on the transmitters, NOT the **Companion**.



1. Go to the **Hardware** screen of the **System Menu**. Scroll down to the Bluetooth settings, and enable **Trainer** mode and then give the device a name. It would appear this is not used for the trainer function.
2. Do the same for the second transmitter being used.
3. Set the **Slave** transmitter trainer function to **Slave/Bluetooth** in the **Model Setup** menu.
4. Set the **Master** transmitter trainer function to **Master/Bluetooth** in the **Model Setup** menu.
5. Now on the **Master** transmitter, the box below, the **Init** box should change to **Discover**. Click on the **Discover** button, and both transmitters should pair.



Setting up the S6R and S8R Stabilised Receivers

Introduction

The S6R and S8R receivers are essentially the same, and are programmed in exactly the same way. The S6R has 6 channels and does not have the RSSI or SBUS connectors. Each receiver has a built in three-axis accelerometer and three-axis gyroscope, and can be configured for a conventional model layout, a delta wing or a V-tail configuration. In the conventional model layout, there are four flight modes, stabilisation, automatic level, hover and knife-edge. These modes are all switchable from the transmitter, and all can be switched off so the receiver operates as a standard receiver. The delta wing and V-tail configurations only allow stabilisation and automatic level. In each mode, gain is controlled by a knob or slider on the transmitter. The quick mode only allows stabilisation and automatic level.

At this point, some might be wondering how all this works given the receivers are only 6 or 8 channels. The reality is that all the X and S series receivers are actually 16 channels. The control functions for stabilisation use four of these spare channels that are normally only accessed by the S-Bus. Both receivers are telemetry equipped and will automatically send receiver battery voltage and signal strength back to the transmitter together with any sensor data from sensors plugged into the Smart Port. An 8 page set of A5 instruction comes with each of these receivers. However, these instructions can be a little hard to follow.

There are 2 distinct ways of setting up the stabilised receivers. The first, and more difficult is to use is the computer and program the receiver using an inexpensive tool called the Smart Port Tool Kit or STK for short, and a program downloaded from the FrSky website. This program is not the easiest to use and sometimes the STK can be difficult to get working on the computer. To overcome this, the thoughtful developers at **OpenTX** have produced two very clear, and frankly much simpler, LUA scripts that allows the user to fully program either stabilised receiver from the transmitter. These LUA scripts can be run from any FrSky transmitters. This **How To** will be using **OpenTX** LUA script. During the setting up process it if often necessary to go back to the programming to say, reverse a servo. It is so much simpler to do this wirelessly from the transmitter. Several of the steps should be completed before installing in the model.

Step 1. Update the Receiver Firmware.

The first task is to check that the receiver has the latest version of the firmware loaded. In 2017 alone, there were 5 releases of firmware. Sadly there is no easy way to tell which version is loaded on the receiver; even a brand new receiver may have been in stock for many months so could have an older version installed. If in doubt install the latest version, just to be sure. From the FrSky website load the latest firmware for the receiver. Once downloaded, there should be two versions available for each receiver:

s8rfcc_20171213.frk
s8rlbt_20171213.frk

S6RFCC_171204.frk
S6RLBT_171204.frk

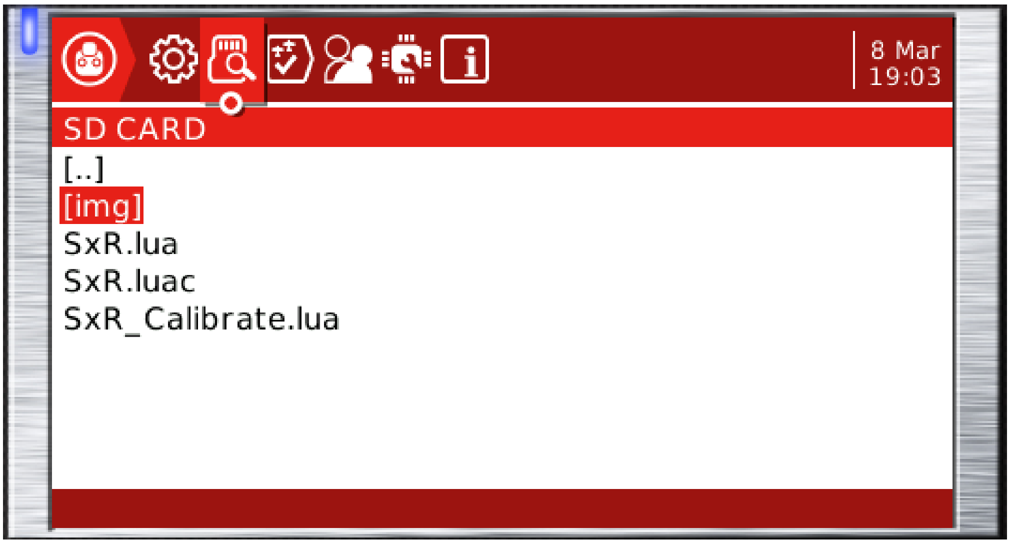
The S6RLBT/s8rlbt is the EU version, the S8RFCC/s8rfcc is the rest of the world version. Do ensure you use the right version for your transmitter setup. Now copy this file onto the SD card. Probably the easiest way is to remove the card and pop it in the computer. Copy the file to the folder called **FIRMWARE** and replace the SD card back in the transmitter. Next go to Section 9, How To … Part 1 and follow the instructions there to update the firmware on the receiver. Please note, the versions shown here may not be the latest versions.

Step 2. Bind the Receiver

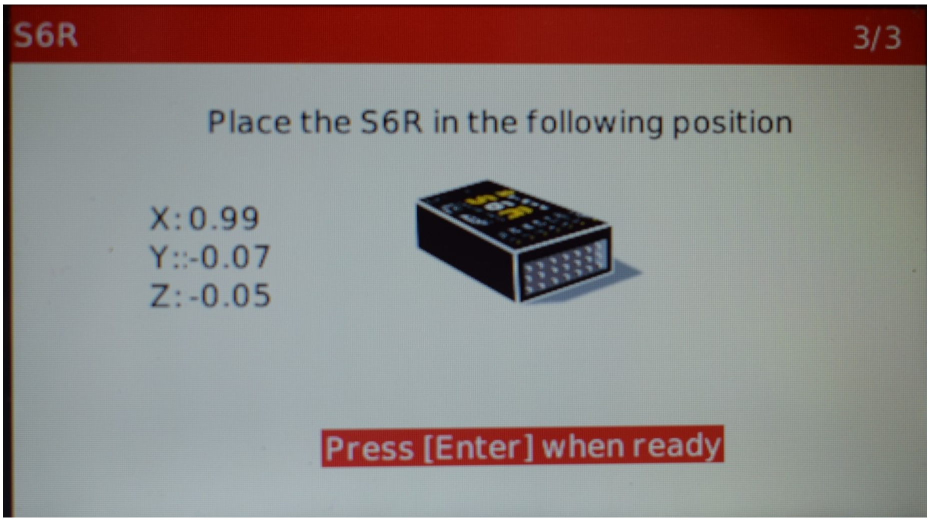
Select the correct model on the transmitter and put the transmitter in bind mode. Next power up the receiver with the F/S button pressed. For the programming following later in this **How To**, when selecting bind mode, select channels 1 to 8, not 9-16. An easy way to power up the receiver is to use a receiver battery with a switch in line. Much easier to then hold down the F/S button with one hand and switch on with the other. When binding, the green light will be on, and the red light will flash. Come out of bind mode on the transmitter and switch the receiver off then on again, and if all is well, the green light should come on.

Step 3. Calibrating the receiver.

The receiver does not need to be physically connected to the transmitter for the next step, all is done wirelessly. With the receiver still on the bench and powered up, select the **SYSTEM** menu on the transmitter and go to the SD card page. Then scroll down and select the SxR folder and you should see the following LUA files:

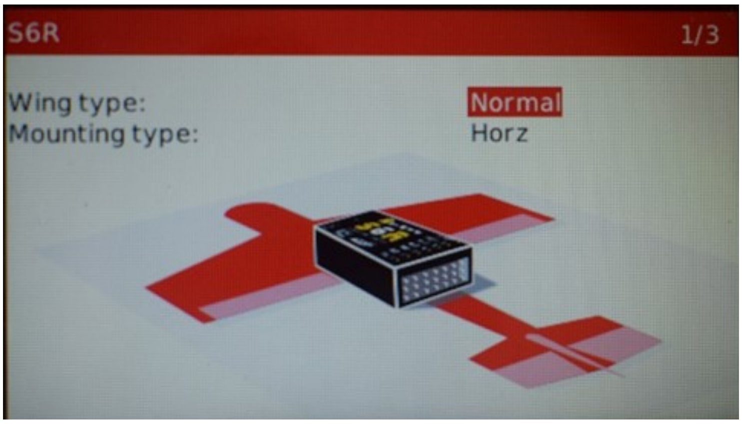


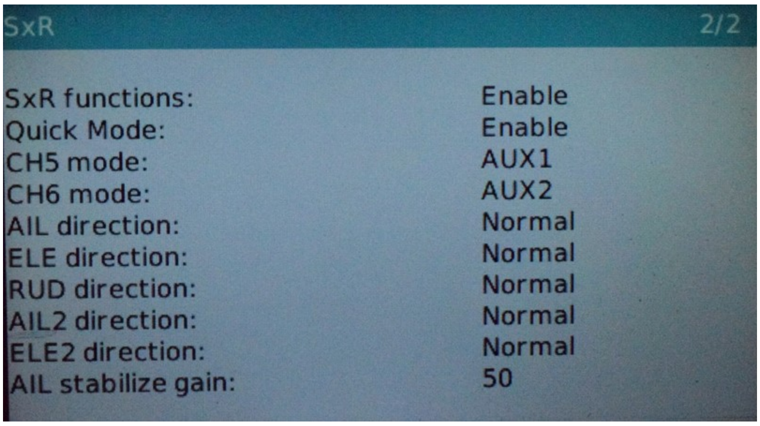
Select **SxR_Calibrate** and execute it. The yellow light should flash whilst each part of the calibrating process is talking place.



Step 4. Setting up the receiver for the model.

This time load the SxR Lua script. The first page selects the wing type and the mounting. Use Page Down to go to the next page.





Finally, the **OpenTX** software needs to be configured to control the stabilisation features. On the setup screen, normally just 8 channels are enabled. This must be changed to 16 channels before anything will work. The instruction sheet is less than helpful in setting up each mode. For stabilisation, the instructions say that channels 10 and 11 should be set to:

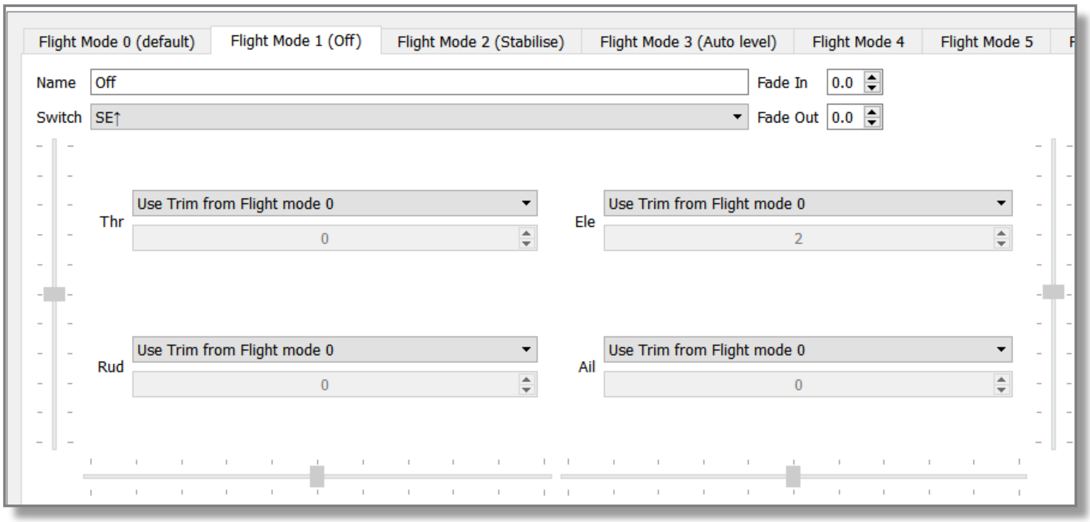
Channel 10	**CH10>M+H**
Channel 11	**M-H<CH11<M+H**

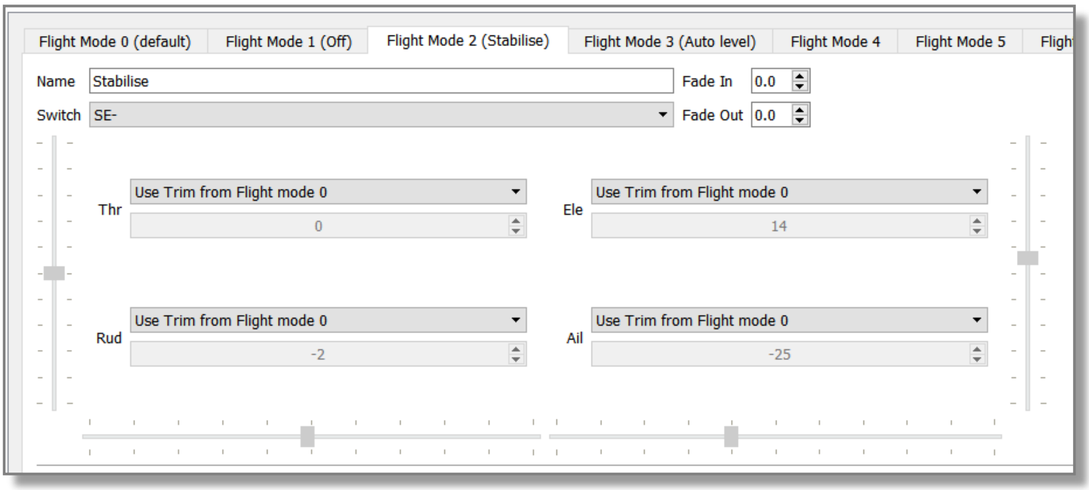
Where M=1500µs represents the neutral signal, and H=50µs represents the required signal change to activate the change.

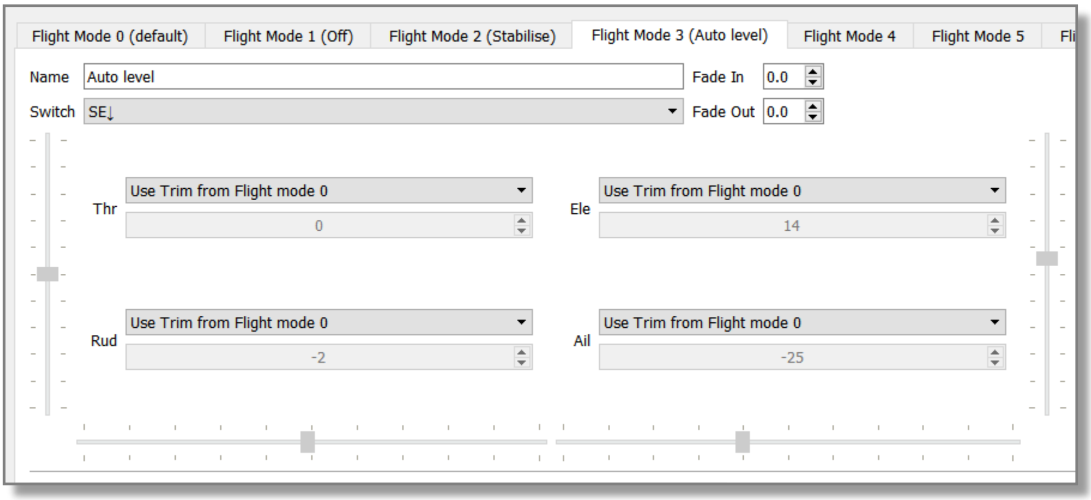
To switch stabilisation off, channel 10 must have a weight of 0

OpenTX uses a pulse width of 1500µs ±512µ. Thus M has a weight of **0.** For **H** a weight of 75 seems a reasonable figure at the time to be on the safe side, and certainly works It is not necessary to program all four modes available into the transmitter.

Thus, a simple program to enable just stabilisation and automatic level mode would look like this. First set up three flight modes as shown.

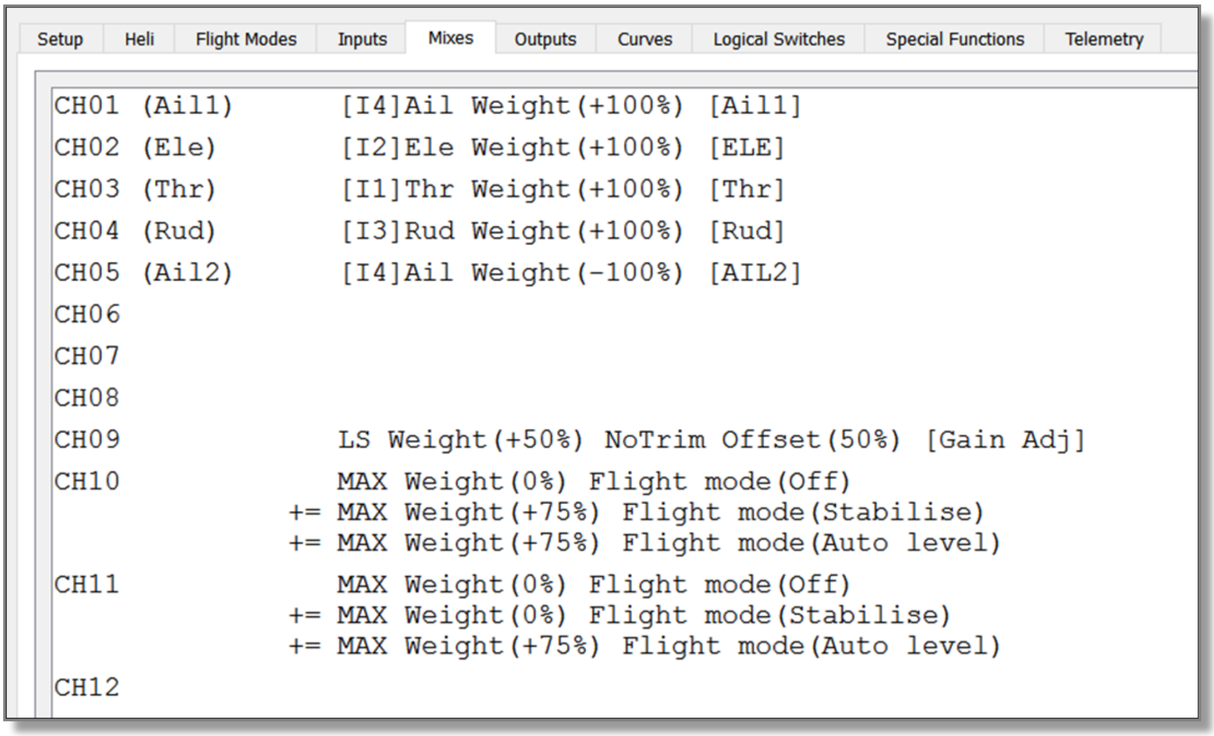






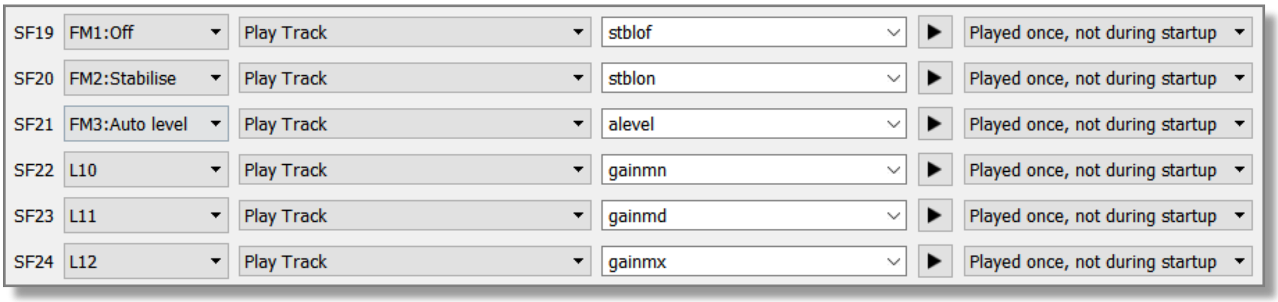
Then the **Mixes** screen needs to set weightings for these three flight modes for channels 10 and 11.

This screenshot shows the **OpenTX Mixes** screen on the **Companion**. Channels 1 to 5 are the



channels used for the plane control surfaces and throttle. Notice here that the inputs for the 5 channels are mixed up, so the throttle on channel 3 takes its input from Input 1. This is because I used an existing model with the channel order set to my own convention, but with the **SxR** receivers I had to use the channels designated on the receiver. Channel 9 uses the left slider on a Taranis/Horus to alter the gain which must have a weight between 0 and 100. Channels 10 and 11 control the two functions.

Sounds can be added in **Special Functions** for the verbal warnings:



Many of these sounds you may have to create yourself. **Gainmn** is gain minimum etc.

Step 5. Final setting up

When installing the receiver in the model, make sure the orientation is the same as set earlier when the receiver was configured. It is also advisable to always try and have the bind button accessible. Once everything is installed in the model and the model is complete with the throws set as desired, there are two further steps that need to be taken:

Check the deflection of each control surface.

Power up the model (without the propeller connected if electric) and switch to auto level mode. In stabilisation mode, the control surfaces should twitch and then return to their normal positions when the plane is moved. In auto level mode, when the model is moved and held in that position the control surfaces will stay in a different position. Thus it is far easier to check their correct movement in auto level mode.

Tilt the plane down, the elevator should move up to correct the dive. Roll the plane to one side and the ailerons should move to correct the roll. Similarly with the rudder. If any deflection is incorrect, use the LUA script to reverse the movement, and then check again.

Perform the self-check.

This check is necessary to attain the auto level angle and gimbal neutral position. Once the self-check is initiated, the model must not be moved until complete. There are two ways to undertake the self-check.

Self-check method 1. The bind button

This method is used if the bind button is accessible with the model level and the right way up.

- Ensure the ailerons, rudder and elevator controls are in the neutral position.
- Ensure the model is on a level surface, and press the bind button and release.
- The blue LED will turn on and the self-check process will start.
- Once the self check has finished, the control surfaces will deflect then return to their normal position and the blue LED will turn off.
- After completion, move the sticks from channel 1 to 6 apart from the throttle to their channel limits to complete the set-up.

The receiver will store all the settings after being powered down. It is not necessary to repeat this every flight.

Self-check method 2. Using the transmitter

This method is used if the bind button is not accessible with the model level and the right way up.

- For this method, channel 12 needs programming by adding the following line in the **Mixes Screen**.

```
CH12          MAX Weight(+100%) Switch(!SC-) [Initia]
```

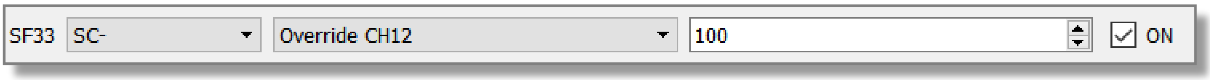
  Any 3 position switch can be used, in this example switch C is used. When switch C is in the mid position, it will have a weight of 0, otherwise it will have a weight of 100.

- Ensure the ailerons, rudder and elevator controls are in the neutral position.
- Ensure the model is on a level surface, and move switch C through the mid point 3 times and the self-check will start.
- Once the self-check has finished, the control surfaces will deflect then return to their normal position.
- After completion, move the sticks from channel 1 to 6 apart from the throttle to their channel limits to complete the set-up.

The receiver will store all the settings after being powered down. It is not necessary to repeat this every flight.

A word of caution: It is very important that this switch is not accidentally operated either before a flight or during a flight. Thus it should be disabled after the self-check. There are several ways this can be achieved.

- Delete the channel 12 instruction once the self-check is completed.
- Move the channel 12 line to a channel higher than 16.
- Add a **Special Function** to disable the mixer line:



  Do remember to ensure the **ON** box is ticked.

Either of the latter two methods are preferred, as sometimes the self-check does need repeating. Months later when it is needed again it is often difficult to remember what was actually programmed in.

The S6R/S8R LED Status:

Green LED	Red LED	Status
ON	Flashing	Binding
Flashing	OFF	Normal
OFF	Flashing	Signal lost

Yellow LED	Status
ON	Accelerometer module value out of [0.9 G , 1.1 G] / Calibration
OFF	Accelerometer module value in [0.9 G , 1.1 G]
Flashing	Complete accelerometer calibration

Blue LED	Status
ON	Self-checking
OFF	Complete self-checking