

Contents

Part 1

Accessing the Model Editor	Page 2
The Setup Screen	Page 4
The Heli Screen	Page 11
The Flight Modes Screen	Page 13
The Inputs Screen	Page 16
The Mixes Screen	Page 21
The Outputs Screen	Page 25
Notes on Servos	Page 28
Inputs Diagram	Page 31
Mixes Diagram	Page 32
Outputs/Servos Diagram	Page 33

Part 2

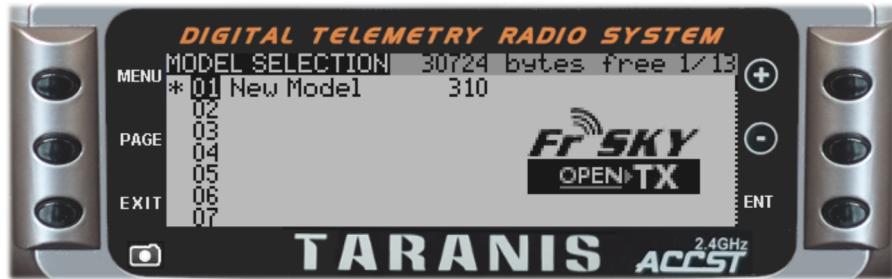
The Curves Screen	Page 34
The Logical Switches Screen	Page 39
The Special Functions Screen	Page 44
Telemetry and the Telemetry Screen	Page 49
The FrSky Smart Port Sensors	Page 51
Discovering the Smart Port Sensors	Page 57
The Telemetry Screen	Page 58
Setting up Telemetry Displays	Page 62
Data Logging	Page 63

This reference section is split into four parts for downloading convenience. It uses the [OpenTX Companion](#) as the starting basis, with references made to the Taranis transmitter screen programming as appropriate. This was done to improve clarity. Also being able to see the whole screen on the [Companion](#) makes it easier to translate to the transmitter screen knowing that a particular feature will be found within that page, but maybe not visible on the screen at the opening of the page. This reference section covers [OpenTX](#) from version 2.1 onwards. It is NOT backward compatible as there are significant differences between this version and earlier versions. In particular the telemetry programming offered in 2.1 is completely different to earlier versions. [OpenTX](#) will work on other transmitters, and some functions may vary from the Taranis 9XD used here.

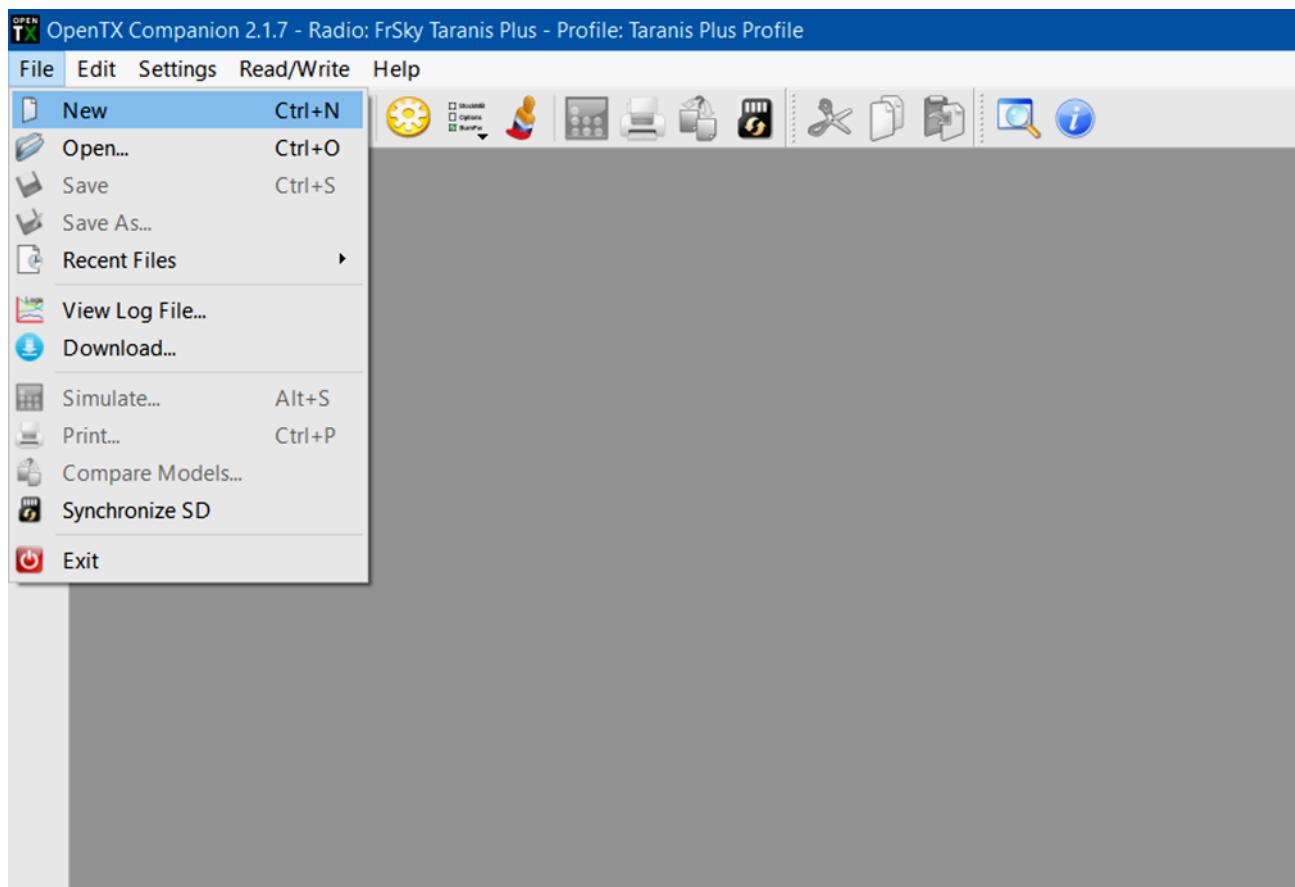
It is the sole responsibility of the user to ensure that the setting up of their transmitter functions as expected on the model.

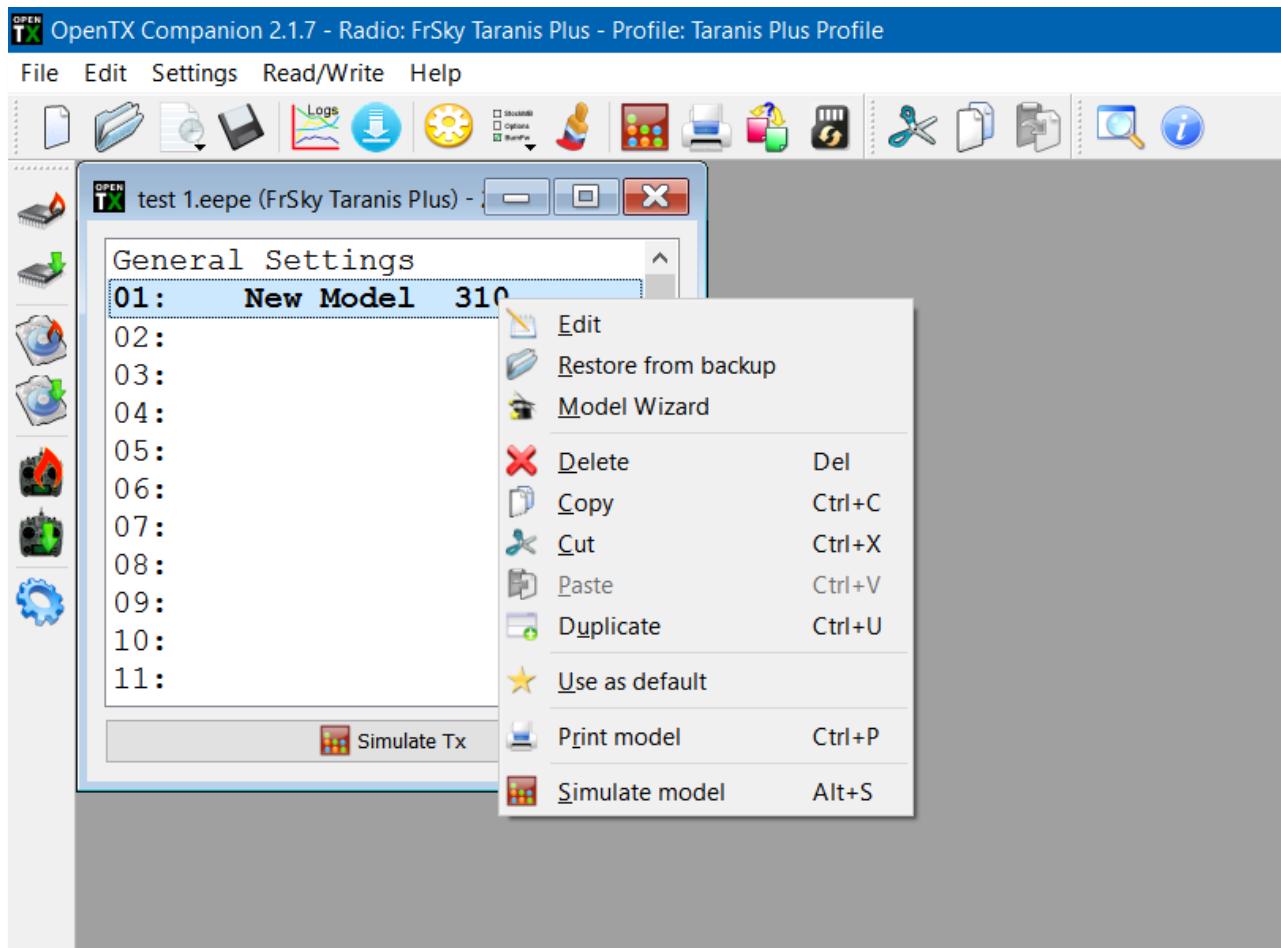
Accessing the Model Editor

The Model Editor can be selected on the transmitter by a short press of the menu button. The first screen of 13 is the **Model Select** screen. A short press of **PAGE** will move through the screens, and a long press of **PAGE** will move back.



Using the **OpenTX Companion**, click on **File** to get the following drop-down menu. Initially one has to start with **New** to create an entirely new Taranis setup. However, once one or more models have been programmed and saved, load using either **Load** or **Recent Files**.

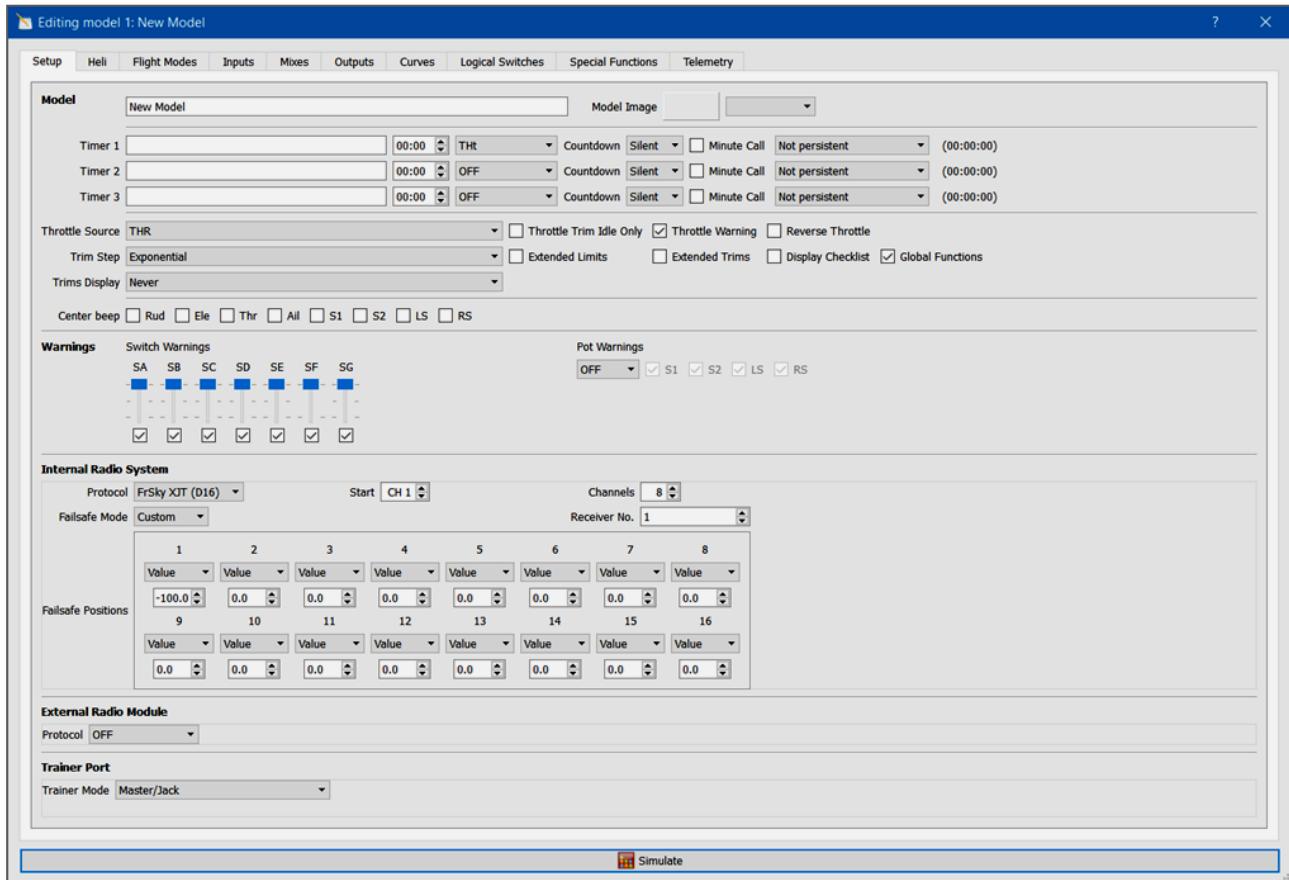




To get to the Model Editor, right click on the model to be edited, and select **Edit**. This will bring up the editing window. The **OpenTX Companion** Model Editor comprises 10 separate windows, selected by the tab at the top of the main editing window. It differs to the Taranis screen version in several minor ways:

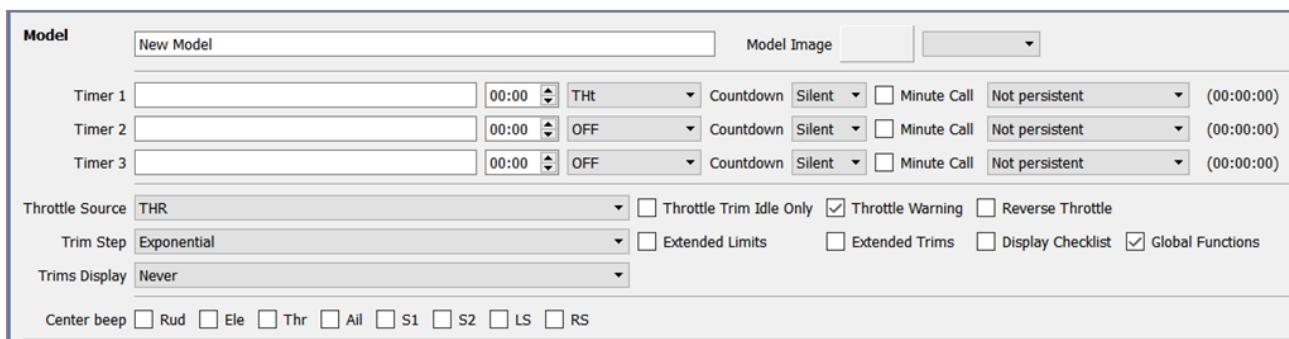
- Screen 6 on the Taranis editor is called **Mixer**, on the **Companion** it is called **Mixes**. These screens do the same job.
- Screen 9 on the Taranis editor is for **Global Variables**. On the **Companion** the **Global Variables** are included on the **Flight Modes** screens.
- Screen 12 on the Taranis editor is for **Custom Scripts**. This has never been duplicated for the **Companion**, adding custom scripts has to be done on the transmitter.

The Setup Screen



There are differences between the **Companion** window and the Taranis screen:

- Throttle Warning on the **Companion** is throttle state on the Taranis.
- Pot Warnings on the **Companion** are called pot positions on the Taranis.
- Receiver binding and range testing, and installing data logging sensors are only available on the transmitter for obvious reasons.



Open TX The Model Editor

Timer 1, 2 and 3

Timer 1, 2 and 3 each have the same features. **Timer 3** is only enabled if the **Timer 3** box in **Settings** is enabled.

- If the timer value is set to 00.00, it will count up from 0, if not, it will count down from the preset value. The timer is set in minutes and seconds with a maximum value of 59 minutes 59 seconds.
- A range of flight modes, logical switches, switches, joysticks and trims can be used to start the timer.
- The timers can be called up from within the logical switches to provide specialist functions.
- **THs** runs when the throttle is not at idle.
- **THt** starts the timer the first time the throttle is opened.
- **TH%** counts up/down as a percentage of the full stick range.
- Persistent, if ticked, means the value is stored when the radio is powered off and later switched on again.
- Minute call will beep/ say the time every full minute.
- Countdown will also give announcements several more times during the last minute.

Note: Some aspects of the timer functions on this screen date back from very early aspects of the program and lack the sophistication of most of the rest of the program. They can be somewhat crude and illogical. It is an improvement task the development team have on their list.

Source gives 4 options, silent, beeps, voice, active:

- Beeps should give a beep at 30sec, 20sec, 10sec, and then every second until zero. However, depending on the sound file used you will get a voice countdown instead of beeps with "10 seconds" repeated for the last 10 seconds.
- The same happens with haptic (if available), though it also vibrates at 15sec.
- The English Amber file gives problems, and one needs to remove the "Timer10" file from the sounds/system folder on the SD card to stop the voice repeating every second during the last 10 seconds.

The timers can be called up from **Special Functions** and timer sequences can easily be formatted from here, especially if custom sound files are created. Remember sound files can use sound effects such as beeps as well as voice commands.

Throttle Source

Throttle source can be: **Thr, S1, S2, (S3), LS, RS and CH1-32**.

Trim Step

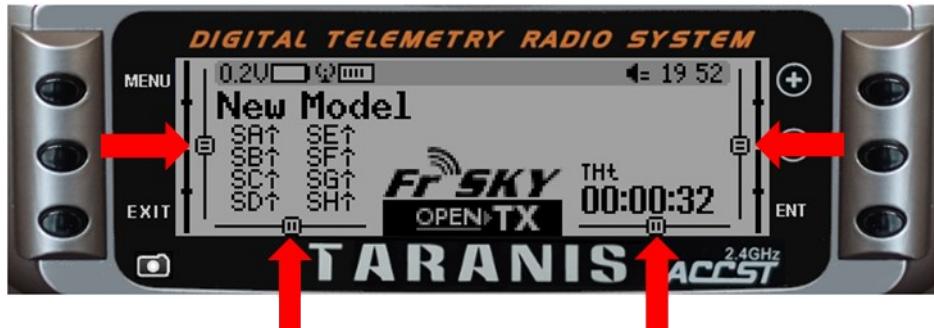
Trim step can be: **Exponential, Extra Fine, Fine, Medium, Coarse**.

For a new model, exponential is very useful. Once a model is well trimmed out, changing the setting to fine or extra fine, will allow for small changes of trim. See also the "Instant Trim" function.

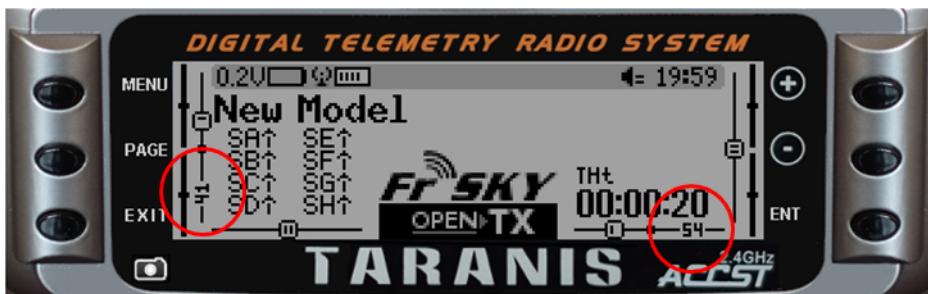
Trims Display

This has three settings: **Never**, **On Change**, **Always**.

This appears a somewhat odd option, as the trims are always displayed on the transmitter screen:



However, with the trims display set to **Always** or **On Change**, the *values* of the trim settings will be also shown.



Throttle Trim Idle Only restricts the throttle trim to the idle only portion of the stick throw unlike the usual offset which affects both ends of the throw.

- | | | | |
|--|--|--|--|
| <input type="checkbox"/> Throttle Trim Idle Only | <input checked="" type="checkbox"/> Throttle Warning | <input type="checkbox"/> Reverse Throttle | |
| <input type="checkbox"/> Extended Limits | <input type="checkbox"/> Extended Trims | <input type="checkbox"/> Display Checklist | <input checked="" type="checkbox"/> Global Functions |

Throttle Warning will display a warning when the model is first selected or the transmitter turned on if the throttle is not in the fully minimum setting. This can sometimes come on even if the throttle is fully in the minimum position. If that is the case, then the joysticks need recalibrating.

Reverse Throttle. This is used to reverse the action of the throttle, so that minimum becomes maximum. This feature was put in for owners transferring from Futaba because Futaba transmitters have their throttles reversed. However, the strong advice is NOT to use this setting when converting an electric model from Futaba, as it can result in the motor being at full throttle at switch on.

The Setup Screen

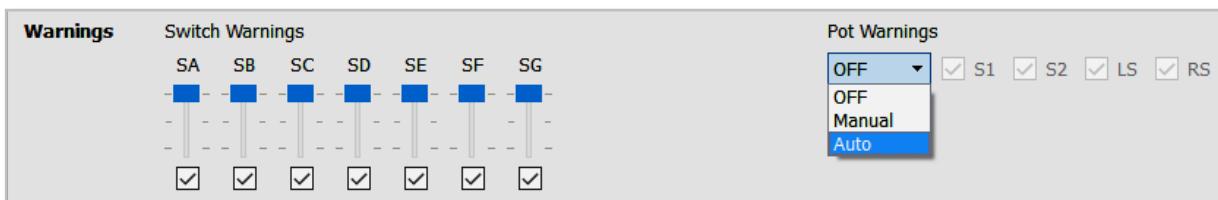
Extended Limits will extend the servo limits beyond their normal 100. With this box checked, travel limits in the **Outputs** screen can be increased to 150%, allowing PWM signals of $1500 \pm 768 \mu\text{s}$ to the servos. Maximum weights available in the **Inputs** screen and **Mixes** screen are unaffected by **Extended Limits**. The **Inputs** screen will accept weights up to 100% regardless of **Extended Limits**. The **Mixer** screen will accept weights up to 500% regardless of **Extended Limits**, but the output from **Mixer** (product of **Input** times **Mixer**) is limited to 100%. This is discussed in more detail in the section on the **Outputs** screen, however, considerable caution should be exercised before using extended limits.

Extended Trim allows the trims to cover the full stick range instead of $\pm 25\%$ of the stick range. Be careful when using this option, because holding the trim tabs for too long might trim so much as to render your model unflyable. Advice is not to use it.

Display Checklist. This is a very useful function which will display a checklist on the screen when a model is loaded or the transmitter turned on. Once one has a number of models programmed on the transmitter it is very easy to forget which switch or other control is given which function, as it is not always possible to have standard uses for each switch or control. The checklist can also provide any other details one might wish to keep recorded with the model such as battery size. A checklist is simply a text file which is shown on the screen. Scrolling up and down can be done with the **-** and **+** buttons, and pressing **EXIT** will leave the checklist. Making a checklist is covered in the **How To** section.

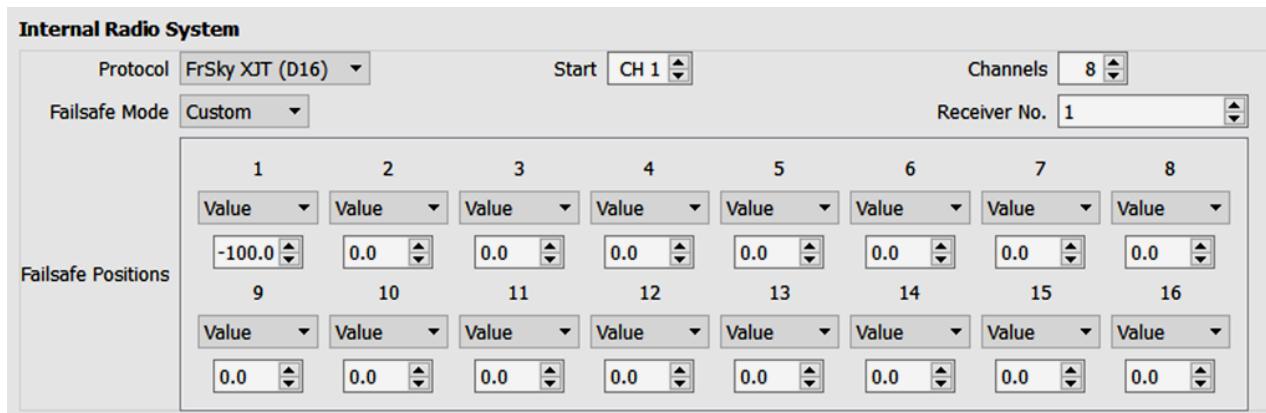
Global Functions. This needs to be checked if using global functions. Unchecked and it will simplify the transmitter screens as these will not appear.

Switch Warnings will give a screen warning when the transmitter is switched on or when a new model is selected if the switches are not in the set positions. This is a useful safety feature for, say, a throttle disable switch on an electric model, or for a gear switch. However, simply leaving them all set just creates a nuisance set of warnings when the transmitter is switched on.



Pot Warnings can be set to **Manual** or **Auto**. **Auto** automatically stores the last positions when the radio is turned off or another model is loaded, and start up warnings will check against those saved positions. **Man** stores current pot and slider settings only by a long press of **ENT** on the label of each pot before exit. Only pots that are highlighted are included in the start-up check/warning. Clearly the **Man** settings can only be carried out using the transmitter.

The Internal Radio System



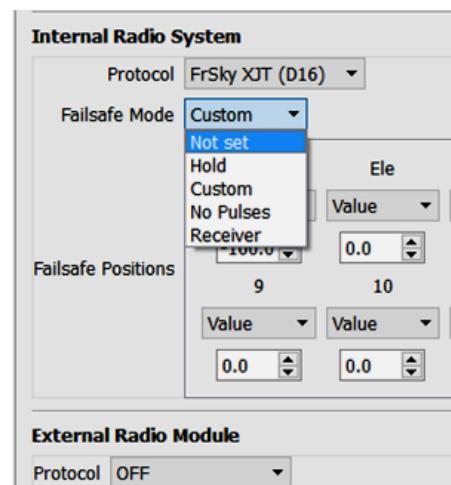
The **Internal Radio System** refers to the XJT module inside the transmitter which sends the appropriate signals to the model. On the Taranis, one can also have external modules which plug in at the back of the case to talk to other makes of receiver. The software for the XJT module is provided by FrSky, and is not part of the OpenTX software. There are 3 protocols available:

- **XJT D8** (older 8-channel two-way mode)
- **XJT D16** (current 16-channel two-way full duplex transmission, also sometimes called "X"-mode). This is the mode to use with the "X" series receivers.
- **XJT LR12** (current 12-channel one-way long range mode). This is said to have approximately three times the range of the current "X" and "D" series when used with the "LR"-series receivers.

The Failsafe Mode

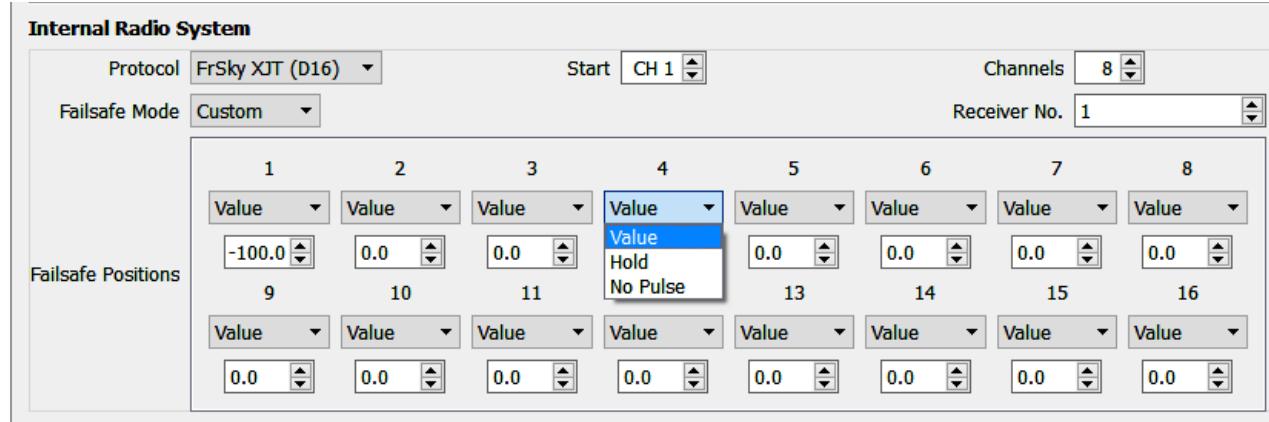
This is a most important function that should always be set. Indeed if a failsafe is not set, a warning will appear every time the model is selected on the transmitter. There are 4 options:

- **Hold** - Holds the last received positions, this could result in a "fly-away" in some circumstances so should be used with care.
- **Custom** - Allows each control channel to be set to a predetermined +/- percent of full travel in the "Set" field.
- **No Pulses** - This duplicates older style radio fail safe designs where no signal is sent to the servo. Typically, the servo would stop at its last position but would not be able to resist forces attempting to change that position.
- **Receiver** - This allows the failsafe to be set using the F/S button on the receiver. For the X series, and LR12 receivers, power up the Tx and Rx, put the sticks and other inputs in the desired position upon loss of signal, press the F/S button on the X series/LR receiver until the green LED blinks.



The Setup Screen

Once the failsafe has been set by [OpenTX Companion](#) or the transmitter, these channel positions will be sent to the receiver each time the radio is started. Whichever method is used to set failsafe, the model should be tested after setting the failsafe. Do this by either removing the propeller, or secure the model using a restraint, and switch the transmitter off. The motor/flying surfaces should go to their set positions.



On the transmitter screen there are two extra functions, **Bind** and **Range** which obviously cannot be undertaken using the [Companion](#).

Bind

Binding is the process of linking a receiver to a particular transmitter model. This prevents accidentally flying using the wrong model program. To bind a receiver to the transmitter, first switch on the receiver with the failsafe button pressed and the red led will glow. On the transmitter, scroll down to **Bind** and then press **Ent**. **[Bind]** will flash, and there will be a series of beeps from the transmitter. The led on the receiver should now flash and the receiver should be powered down. Come out of bind mode on the transmitter and the receiver should now be bound. If this process does not work, ensure the transmitter and receiver are at least a metre apart and rebind.

Range

This does a range test. It reduces the range by a factor of 30, the power is reduced by 900. On a range test, in ideal conditions with the model unobstructed and at least a metre off the ground the range to the critical alarm warning should be about 100m. In real flight conditions the operating range of the X8R is about 1.5 - 2km. The Taranis display will show the RSSI, received

The Setup Screen

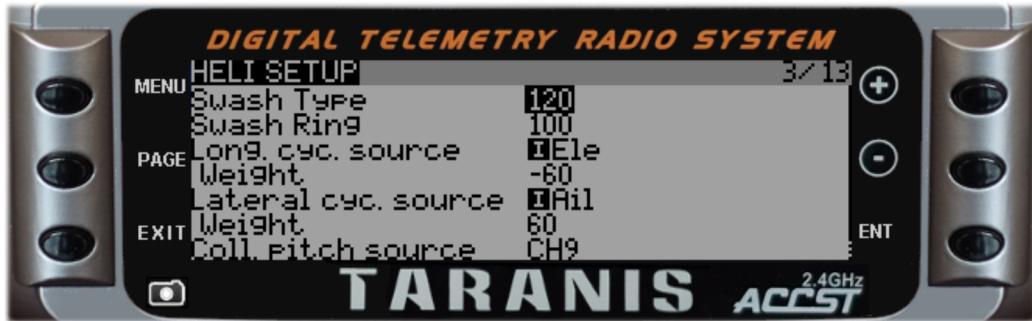
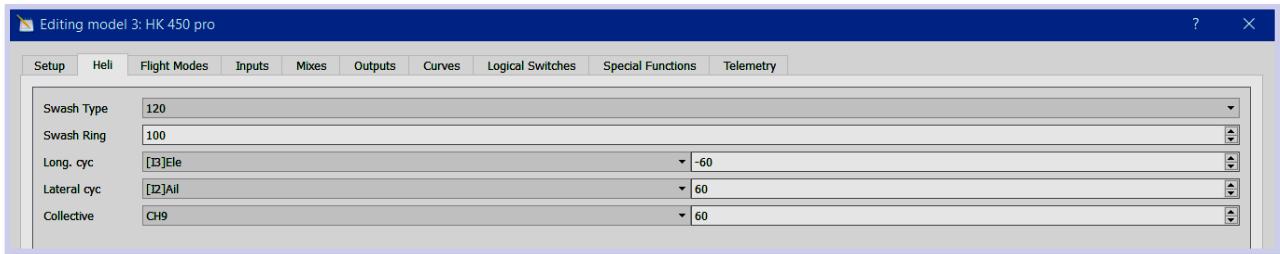
signal strength indicator, which is measured in decibels. The low RSSI audible alarm is pre-set at 45, and critical RSSI alarm is set at 42, though these can be changed. These pre-defined alarms still give a considerable leeway before the signal is lost. In normal conditions even if the critical alarm is heard the receiver would still be far from losing control; the receiver would be at least 1000m away, with another 500m to spare.

The received signal strength (**RSSI**) is actually not an absolute value like voltage or temperature but a number that indicates the ratio of the signal to some initial "good" value. It is in dB and is the same measuring system used for audio levels. For a fixed single system like the FrSky system we can treat the **RSSI** as an absolute value of strength since the "good" value is already chosen and fixed. The only unusual thing is that dB is a logarithmic measure, not linear. What this means is that any increase of 6 means the signal has doubled in strength. So a change of +12 in the RSSI means the signal has increased in power by 4 times, by +18 means the signal has increased in power by 8 times, by +24 means the signal has increased in power by 16 times. Conversely if the **RSSI** goes down by -18 the signal power is 1/8 what it was.

For the FrSky system the **RSSI** should read about 100 at 1m and every time you double the distance between the transmitter and receiver, the **RSSI** level should drop by 6. That's why it goes down quickly at first then less and less as the model flies away since you have to double the existing distance each time to get 6 lower.

Open TX The Model Editor

The Heli Screen



Note: The **Noheli** option in the radio profile must not be ticked, otherwise this heli screen will not appear.

The inputs of this mixer are the aileron and elevator joysticks, plus the virtual channel selected in "Collective Pitch Source". This channel would see entries added on the **MIXER** page for one or more pitch curves to adjust the weight of the collective. The outputs of the CCPM mixer are **CYC1**, **CYC2** and **CYC3**, which need to be assigned on the **MIXER** page to the channels you will connect your servos to. Note that the settings made here have no effect unless you are using those **CYC1**, **CYC2** and **CYC3** sources. A multirotor or flybarless helicopter which uses onboard computers/mixers will NOT use them.

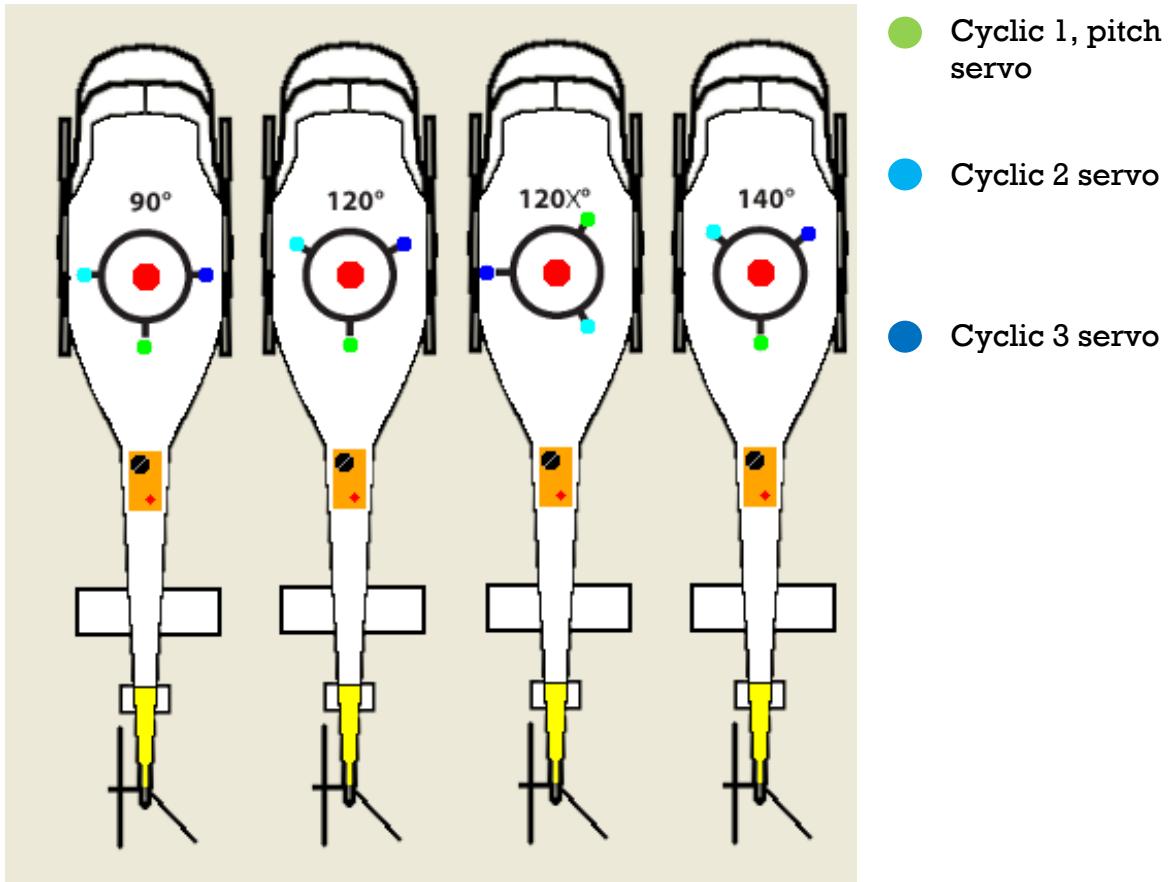
The **OpenTX Companion** and transmitter screens above show a setup for a helicopter with a flybar. Here we have the aileron, elevator, and collective (assigned through a virtual channel, in this case channel 9) which are combined to give the helicopter swash plate servo commands. This virtual channel assigned in the **Mixes** screen, is actually a channel beyond the receiver channel range. Channel 9 was used because the X8R receiver only support channels 1 to 8.



Swash Type

This defines what kind of Swash plate you have on your helicopter. Swash type offers several different choices depending on the servo arrangement on the swashplate:

- **Off** Control is handled by other onboard electronic systems
- **120** The standard 120° swash plate. The pitch servo is towards the front/back.
- **120X** Same 120° swash plate but turned 90° so the pitch servo is on one side.
- **140** 140° swash plate, again, the pitch servo is towards the front/back.
- **90** 90°, basically a simple 90° setup where you have a single servo operating the pitch and two operating the roll.



Swash Ring

Swash Ring actually crops the "square" nature of the aileron/elevator stick to a circle. If the joystick is moved fully left from the centre, this gives a movement with an amplitude of 100%. It is the same if the joystick is centred and moved fully down. But, as the joystick can do a square, if the joystick is moved to the lower left corner, the total amplitude movement is actually of 140%. This means that depending on the different adjustments and limits, even if all the cyclic angles are at $\pm 100\%$ of every single control, there could be a servo that hits its limit earlier than expected if the joystick is moved to the corner, and due to the cyclic mixing if one servo hits a limit it could affect all the 3 functions with unpredictable flight behaviour.

Setting **Swash Ring** to 100 will "round off" the corners, and never allow an input of more than 100% total deflection with the joystick. Setting the **Swash Ring** to less than 100 will reduce the overall servo movement further, however it is probably better to adjust the servo linkages to reduce travel, yet still have full servo movement than use a reduced **Swash Ring** setting.



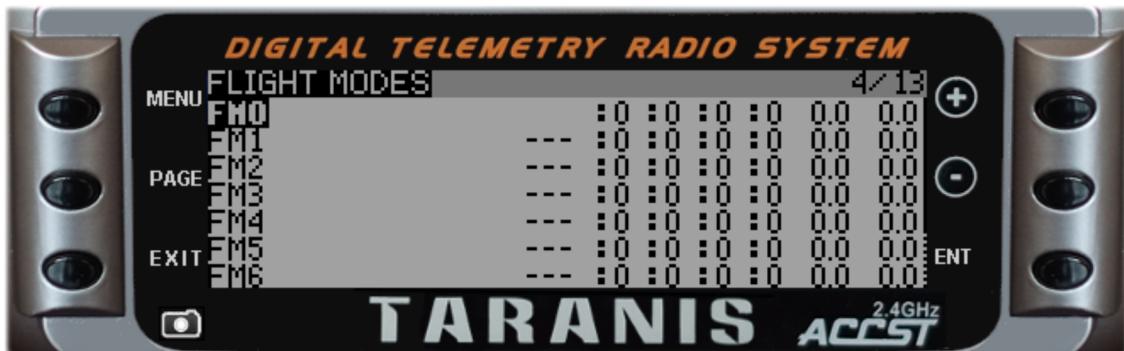
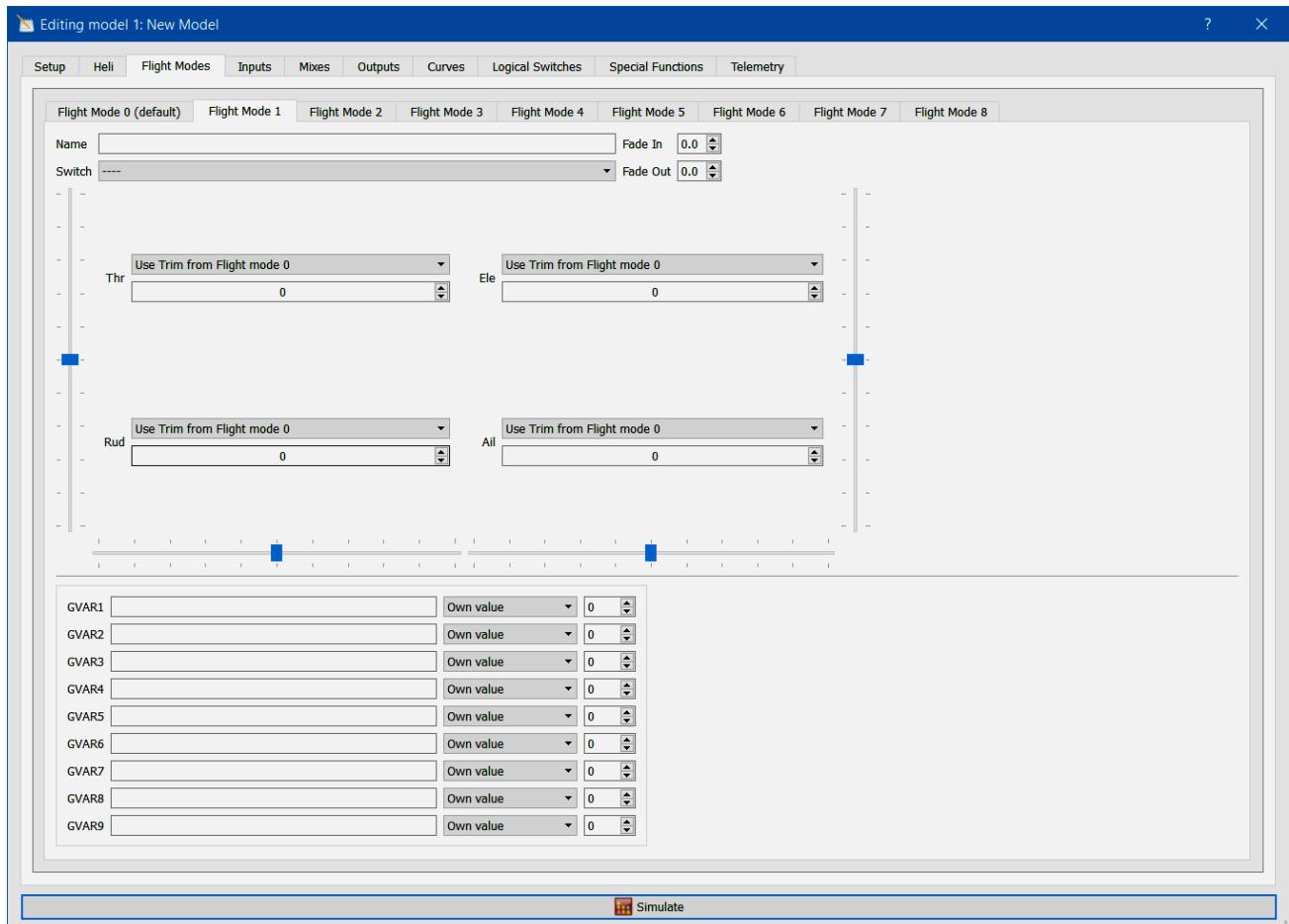
Long Cyclic, Lateral Cyclic and Collective

Here you can assign a range of inputs, and add weightings to these inputs.

Open TX The Model Editor

The Flight Modes Screen

The Flight Modes Screen



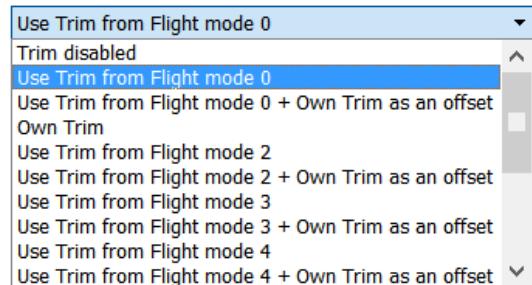
Nine flight modes including the default one are available for use. Each of them has the following:

- They can be named
- There is a selectable activation switch, either physical or logical, (apart from the default mode)

Open TX The Model Editor

The Flight Modes Screen

- A trim selection array (R, E, T, A when shown) mean the mode has its own trim setting for that control, but each can be changed to a number from 0 to 9 and thus use the same value as the specified mode).
- Slow up/down parameters for smooth transitions between modes.



The priority of the flight modes is such as the first FM of 1-8 that has its switch ON is the active one. When none has its switch ON, the default FM0 is active.

Global Variables

Normally the layout of **OpenTX Companion** and the transmitter screen are as close as practicality allows. However, when using **Global Variables**, on the **Companion** they are included on the **Flight Modes** screen. On the transmitter they appear as screen 9 between the **Curves** screen and the **Logical Switches** screen.

Global Variables are values that can be substituted to the usual number on every Weight, Offset, Differential or Expo setting. Their main use is to group the adjustment of several parameters that should have the same value. For example, consider an aileron differential on a glider with 4 surfaces responding to the aileron function. When trying to find the sweet spot for the differential value, instead of having to repeatedly edit the differential value in 4 mixers, all 4 can be set to use a global variable e.g. **GV1**. Then adjusting **GV1** on this page is all it takes for all differentials to be updated. Global variables are also flight mode specific, so instead of having to create separate mixer lines with different values depending on the flight mode one can simply use a global variable with different values for each flight mode. This can significantly help simplifying the mixer screen by avoiding many duplicate entries. Global variables can also be adjusted in flight thanks to the Adjust GV in the **Special Functions** screen. Thus adjusting those parameters that are easier to set up in flight like D/R ratios, expos or again differential becomes straightforward. The **Global Variables** screen allows setting a name for each of the 9 available variables for convenience, and seeing/setting the value each of them will have in each of the 9 flight modes.



Open TX The Model Editor

The Flight Modes Screen

The following screenshot is using flight mode 0.

GVAR1	<input type="text" value="25"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	<input type="checkbox"/> Popup enabled
GVAR2	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	<input type="checkbox"/> Popup enabled
GVAR3	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	<input type="checkbox"/> Popup enabled
GVAR4	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	<input type="checkbox"/> Popup enabled
GVAR5	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	<input type="checkbox"/> Popup enabled
GVAR6	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	<input type="checkbox"/> Popup enabled
GVAR7	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	<input type="checkbox"/> Popup enabled
GVAR8	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	<input type="checkbox"/> Popup enabled
GVAR9	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	<input type="checkbox"/> Popup enabled

Flight modes 1 to 8 have an extra drop down box, **Own value** available. This box enables the global variable to take the value from any of the other flight modes.

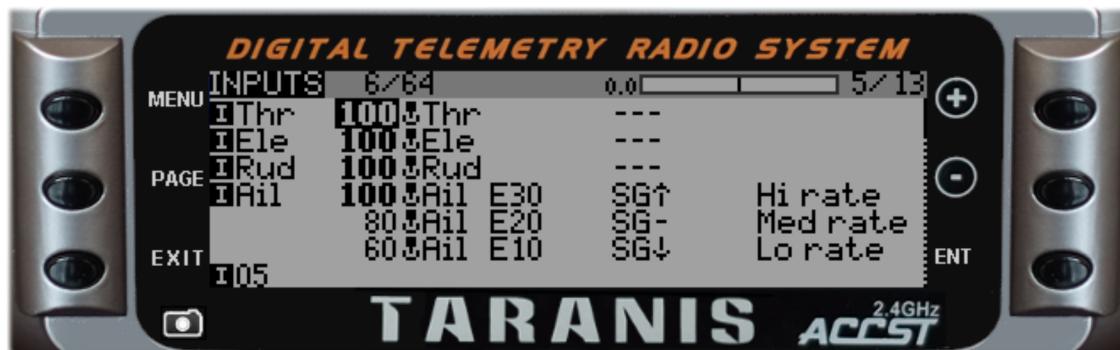
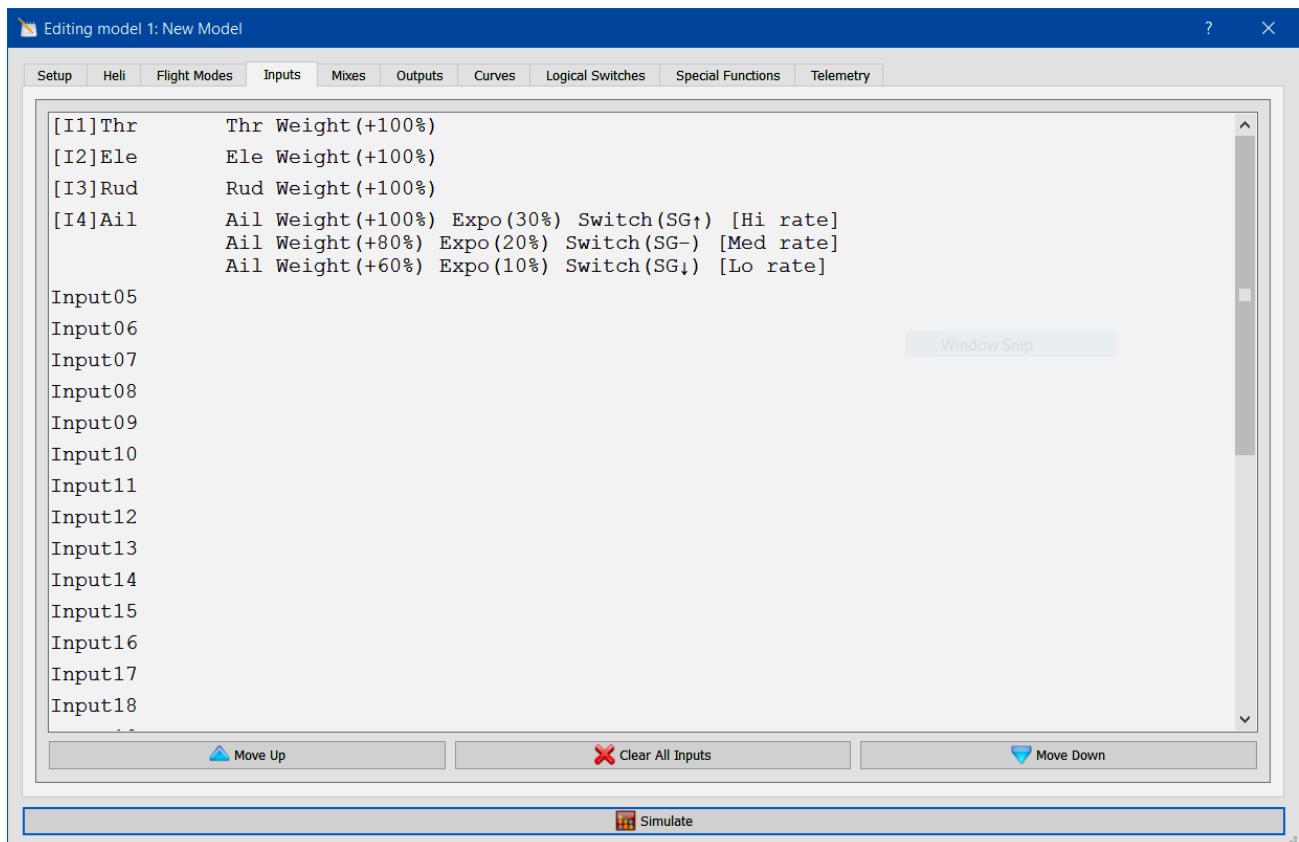
GVAR1	<input type="text" value="20"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>
GVAR2	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>
GVAR3	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>
GVAR4	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>
GVAR5	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>
GVAR6	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>
GVAR7	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>
GVAR8	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>
GVAR9	<input type="text" value="0"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>

Own value dropdown menu:

- Own value
- Flight mode 0 value
- Flight mode 2 value
- Flight mode 3 value
- Flight mode 4 value
- Flight mode 5 value
- Flight mode 6 value
- Flight mode 7 value
- Flight mode 8 value

Open TX The Model Editor

The Inputs Screen



The **Inputs** screen is where the transmitter controls are linked to the inputs that will be manipulated by **OpenTX**. The left hand column shows the inputs defined, highlighted with a capital **I**. The next column shows the actual transmitter controls that are linked to these inputs; in this case the transmitter joysticks. The order of these was defined in the **Setup** screen.

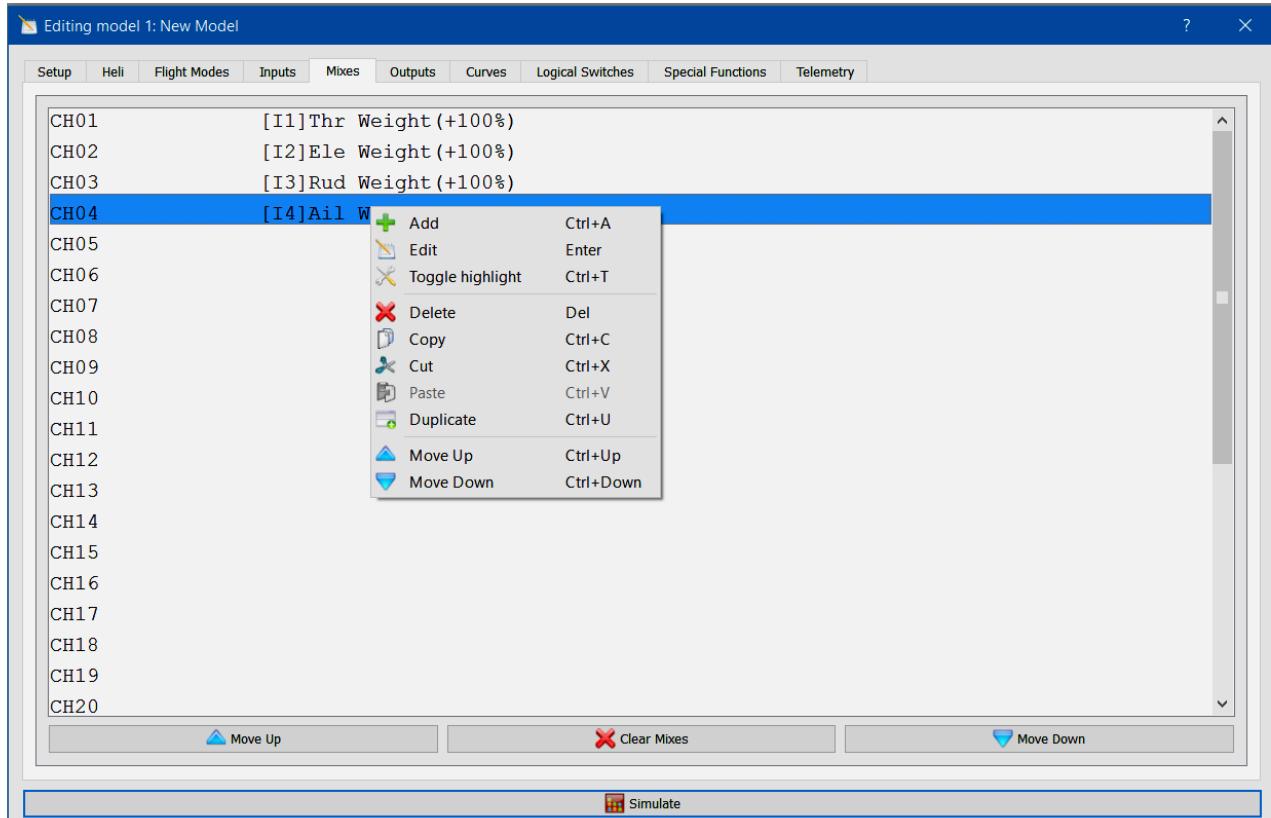
Notice there are slight differences between the **OpenTX Companion** and the transmitter screen. The input **I** on the **Companion** is followed by a number. On the transmitter the actual input types are shown after the weight.

- Sticks
- Pots
- MAX
- Cyclic
- Trims
- Switches
- Trainer
- Channels

The Inputs Screen

Care must be taken whilst programming not to mix up the actual control with the defined input. For instance on an input window, it offers both the raw function, i.e. **Rud**, and the rudder input control **[I3]Rud**.

Clicking on any line on the **Inputs** screen gives a sub-menu:



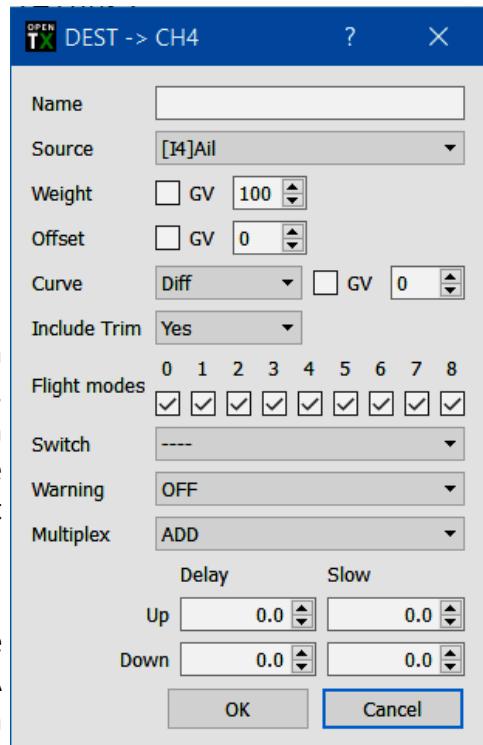
These are largely self-explanatory, and some lead to further sub-menus. Programming on the transmitter also gives these menus, using a combination of long and short presses will allow the various functions to be accessed.

Toggle highlight will highlight the particular channel, and all instances where this channel is shown. Useful with complex model setups.

Move Up and **Move Down** are used to shift the line up and down. The order in which the lines are stored is the order in which **OpenTX** will apply the lines. Thus if the first line is true due to a condition, **OpenTX** will act upon that line, if not it will move down to the next line.

The Inputs Screen

- **Input Name** (4 characters)
- **Line Name** (8 characters):
There can be many lines configuring a channel. Naming each line helps keep up with the purpose of the line and avoid confusion, especially when returning to alter the programming some time later. E.g., naming high rates etc.
- **Source:**
There are over 100 possible input sources available in the choice box, including all sticks, switches, sliders, trims, telemetry, and other channels. This provides an opportunity to use an otherwise idle channel to create another combination that can become a switched input source for an active channel.
- **Weight:**
This is similar to “Rate” in other radios. 100% is the maximum weight available to us in the Inputs screen. A negative value entered here will reverse the direction of servo rotation, but it is strongly recommend not to use negative weights in **Input** to reverse servo direction.
- **Offset:**
Defined as a fixed amount added to an input value. When used on flight controls, it can be thought of as a trim value that is not adjustable in flight. When used with calibrations, it allows displayed values to be corrected to actual measured values. When used with curves, it can act as a constant for adjusting output characteristics.
- **Curve** (2 fields):
There are 4 options, **Diff**, **Expo**, **Funct** and **Curve**. **Diff** (or differential) can accept a number or a **Global Variable**. The **DIFF** weight entered applies to one side of the action and 100% minus the DIFF weight on the other. For example DIFF = 100% allows full movement to one side and 0% to the other side.
OpenTX uses positive values to soften the servo response near stick centre. This allows the pilot to have large total deflection of a control surface at 100% stick movement while retaining fine control near stick centre.
- **The Inputs Curve Functions**
 - ✗ **(X>0)** only operates on the positive side of the stick and **(x<0)** works on the minus side. You could use these for setting differential or complicated ailerons.
 - ✗ **([x])** moving the stick from centre either way results in a movement of ±100.
 - ✗ **(F>0)** results in full travel to the positive side as soon as the stick is moved to the plus side and has no output on the negative side.
 - ✗ **(F<0)** it's the opposite of **(f>0)**.
 - ✗ **([f])** as soon as a stick is moved the output goes full travel to the side the stick is



Open TX The Model Editor

moved towards.

Note:

Many of these are "legacy" functions from er9X or even earlier. They can be used to generate things like differential, which was added later. They would still be used to generate things like knife edge mixes where you want the correction to be the same for both stick directions

→ **Flight Modes:**

By default, each channel configuration line is available in all flight modes. This field allows you to specify which flight modes may and may not include a given configured input line. To exclude a line from a flight mode, edit the list of flight modes to blank the number for the flight mode to be excluded.

→ **Switch:**

This identifies which of the available switches will be used with the input. Logical switches can also be used.

→ **Side:**(3 choices). The default ‘—’ allows both sides of the stick to be configured by this line. The other two settings, **x>0** and **x<0** are used to limit the configuration settings in this line control action to one side of the stick or the other. The single-sided options allow each stick side to be treated as an individual input if that would be useful. For example, each side could use different weights and expo values. Later in the **Mixer**, each side could be mixed with different combinations of variables.

→ **Trim:**

(Six choices). **On, Off, Ail, Ele, Thr, Rud.** Selection determines which trimmer value, if any, is added to the input source. Can be used to set up “cross trims” that allow a free finger or thumb to adjust in-flight trim for a difficult initial flight, such as using a left thumb to trim elevator and aileron via left stick trimmers while the pilot controls elevator and aileron with his right hand. The **Off** choice adds no stick trim to that stick if that is useful for other set up combinations.

Note: The choice to include **Trim** appears in both the **Inputs** screen and the **Mixer** screen. In order for **Trim** to be passed to the servos, it must be enabled in both the **Inputs** screen and in the **Mixer** screen.

Important Warning

When assigning switches to either **Inputs** or **Mixes**, do ensure that no switch position is unassigned otherwise this could lead to unanticipated behaviour. E.g. using dual rates on the

```
Ail Weight (+100%) Expo (30%) Switch(SG↑) [Hi rate]
Ail Weight (+80%) Expo (20%) Switch(SG-) [Med rate]
```

Inputs screen:

Here switch SG↓ is unassigned, and should this switch be accidentally moved to the fully down position, then the ailerons will not move at all and will remain at weight 0%. There are two ways of ensuring this does not happen, both are equally valid.

```
Ail Weight (+100%) Expo (30%) Switch(SG↑) [Hi rate]
Ail Weight (+80%) Expo (20%) Switch(SG-) [Med rate]
Ail Weight (+80%) Expo (20%) Switch(SG↓) [Med rate]
```

Method 1

Always assign every switch position every time:

Now SG↓ simply replicates SG-. If one is going to do this, one might as well make use of the 3 position switch and enable triple rates thus:

```
Ail Weight (+100%) Expo (30%) Switch(SG↑) [Hi rate]
Ail Weight (+80%) Expo (20%) Switch(SG-) [Med rate]
Ail Weight (+60%) Expo (10%) Switch(SG↓) [Lo rate]
```

Method 2

Use a “catchall” line:

```
Ail Weight (+100%) Expo (30%) Switch(SG↑) [Hi rate]
Ail Weight (+80%) Expo (20%) Switch(SG-) [Med rate]
Ail Weight (+100%) [Catchall]
```

Basically a catchall line will assign a particular operation to a sequence of operations when an expected condition is not met, in this example when SG is down.

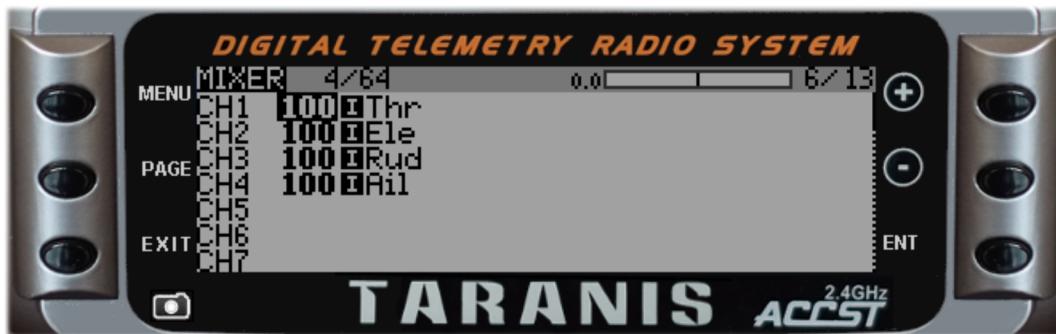
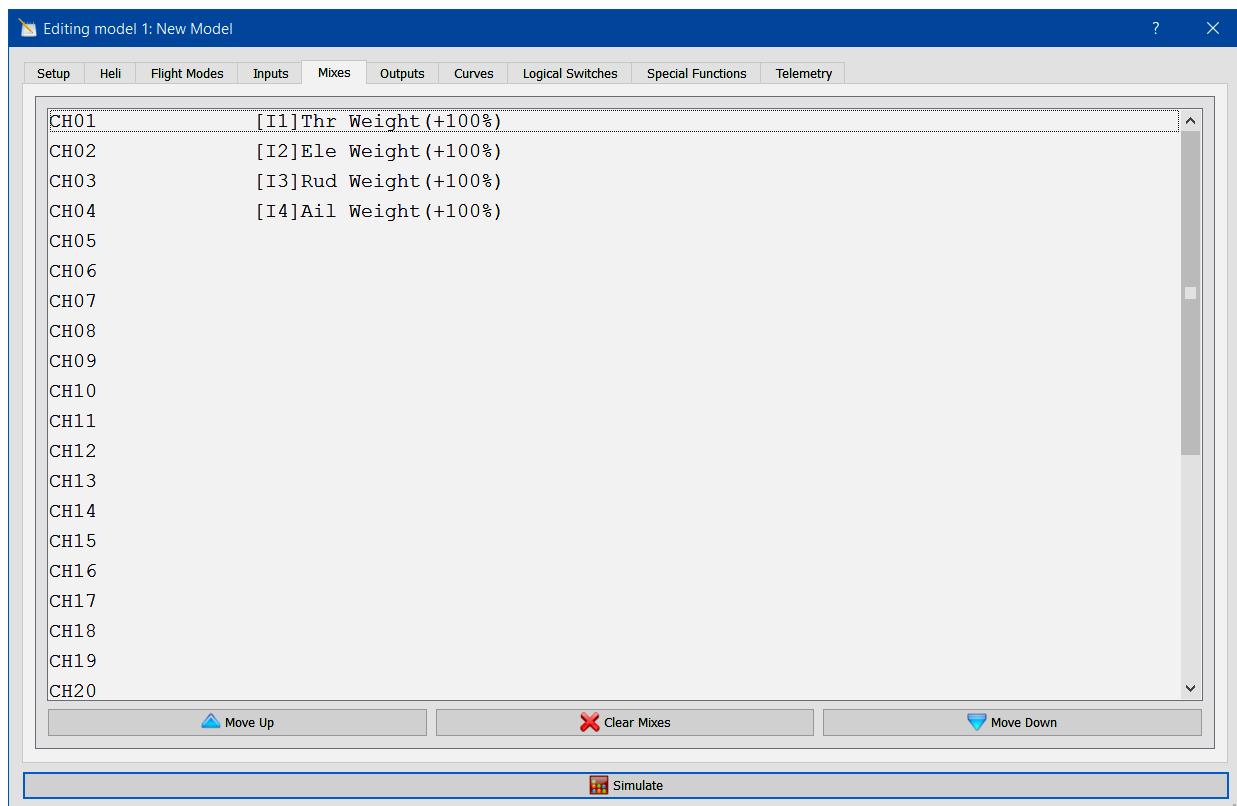
The example above illustrates how important it is to always check out the functionality of model settings before flying with a new configuration or after changing the configuration. The **Companion** simulator is an excellent method for doing this as each combination of switch and controls can quickly be checked. This check is, of course, in addition to the normal field checks made before each flight.

Remember it is the pilot's responsibility to ensure that a model is correctly configured and operating properly before every flight.

Open TX The Model Editor

The Mixes Screen

The **Mixes** screen is where the actions on the controls will be mixed and mapped to outputs and thence on to servos. **OpenTX** does not have any predefined mixing functions that relate only to a particular model type or situation, it rather gives you a blank canvas you can build upon. The key to configuring a model on **OpenTX** is not to think about "activating the delta mix" like on certain radios, but rather to think about what you want your control on the model to do in response to an input on the radio's controls. The mixer is where all that "logic" gets entered. The various channels are outputs, for example **CH1** being channel 1 on the receiver (with the default protocol settings). A channel without a mixer line will just centre a servo that would be connected to it.



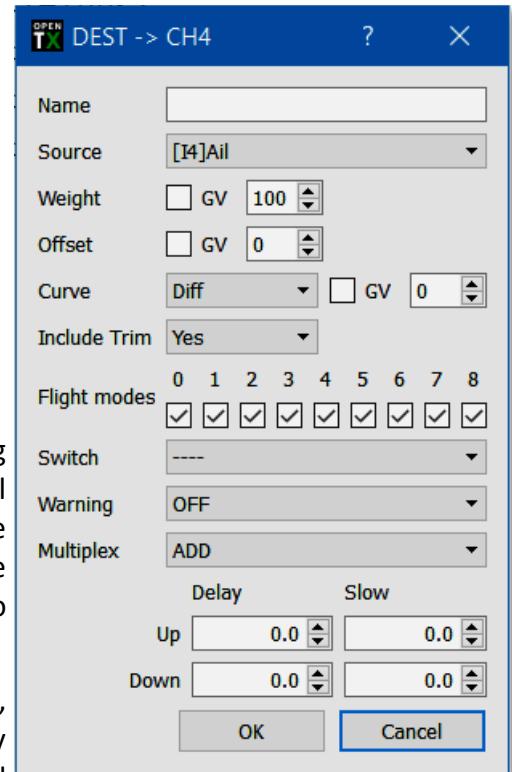
Open TX The Model Editor

The Mixes Screen

Each mixer line connects one input to the channel it's on. Inputs can be:

- The 4 joysticks
- The pots and sliders
- The heli mixer outputs **CYC1-3**
- A fixed value **MAX** (consider this as simply +100)
- The physical switches
- The 32 custom (logical) switches
- The trainer port input channels (PPM1-8)
- A LUA script
- Each of the radio's 32 channels, which allows using channels as a virtual functions for clarity (mix several inputs into one re-useable function that can then be assigned to one or more channels). Note that the settings of the **Outputs** screen are NOT taken into account there.

All inputs work on a -100% to +100% basis. Sticks, pots, channels, CYC sources, trainer inputs will vary proportionally within this range. 3-position switches will return -100%, 0% or +100%. 2-position switches (and logic ones) will return -100% or +100%. MAX is always +100%.



If you want the servo connected to the number 2 channel of your receiver to be controlled by the elevator stick, you will simply create a mixer entry on **CH2** with **Ele** as source.

There can be as many lines as needed on each channel, and the operation between each line can be selected. By default all the lines on a same channel are added together, but a line can also multiply those before it, or replace them.

For clarity on the transmitter display, each line that is currently active and contributing to the channel's output will have its source displayed in **bold**. This can be very handy when many lines are present and to check switch functions.

For each mixer line, several parameters are available:

- **Name**
A name can be entered for convenience. Up to 8 characters are allowed
- **Weight**
The weight (in %) of the input can be set. This sets how much of the input control has to be mixed in. A negative value inverts the response.
- **Offset**
An offset on the input value can be added.
- **Curve**
Either a differential value can be set (reduces response by the specified percentage on one

side of the throw) or a curve (built-in or custom) can be assigned. When a custom curve is selected, a press of the **MENU** key will bring you to the curve editor on the transmitter screen.

→ The Mixes Curve Functions

- ✗ **(X>0)** only operates on the positive side of the stick and **(x<0)** works on the minus side. You could use these for setting differential or complicated ailerons.
- ✗ **([x])** moving the stick from centre either way results in an of 0/+100.
- ✗ **(F>0)** results in full travel to the positive side as soon as the stick is moved to the plus side and has no output on the negative side.
- ✗ **(F<0)** it's the opposite of **(f>0)**.
- ✗ **([f])** as soon as a stick is moved the output goes full travel to the side the stick is moved towards.
- ✗ Note:
Many of these are "legacy" functions from er9X or even earlier. They can be used to generate things like differential, which was added later. They would still be used to generate things like knife edge mixes where you want the correction to be the same for both stick directions.

→ Include Trim

A trim can be used, for sticks this is by default the trim associated to the stick, but can be chosen to be one of the other trims (for cross-trimming for example) or disabled altogether. For other inputs the trim defaults to **OFF**, but can of course be set to one if required.

→ Flight Modes

By default, each channel configuration line is available in all flight modes. This field allows you to specify which flight modes may and may not include a given configured input line. To exclude a line from a flight mode, edit the list of flight modes to blank the number for the flight mode to be excluded.

→ Switch

A switch (physical or logical) can be used to activate the mixer line.

→ Warnings

A sound warning (1, 2 or 3 beeps) can be set to play whenever the line is active.

→ Multiplex

The Multiplex setting defines how the current mixer line interacts with the others on the same channel. **Add** will simply add its output to them, **Multiply** will multiply the result of the lines above it, and **Replace** will replace anything that was done before it with its output. The combination of these operations allows creating complex mathematical operations.

→ Delay

Response of the output can be delayed and/or slowed down with regard to the input change. Slow could for example be used to slow retracts that are actuated by a normal proportional servo. The time is how many seconds the output will take to cover the -100 to +100% range.

The Mixes Screen

As an example, if you wanted to add some compensation on the elevator channel when you increase throttle, you would go through a simple path:

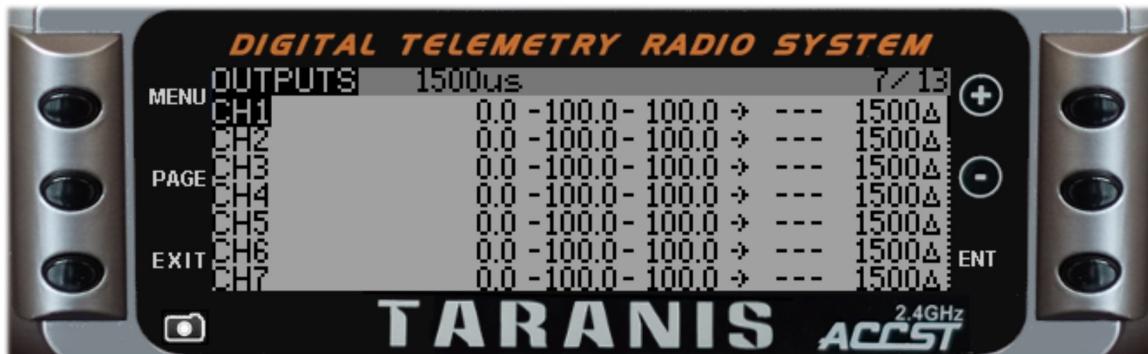
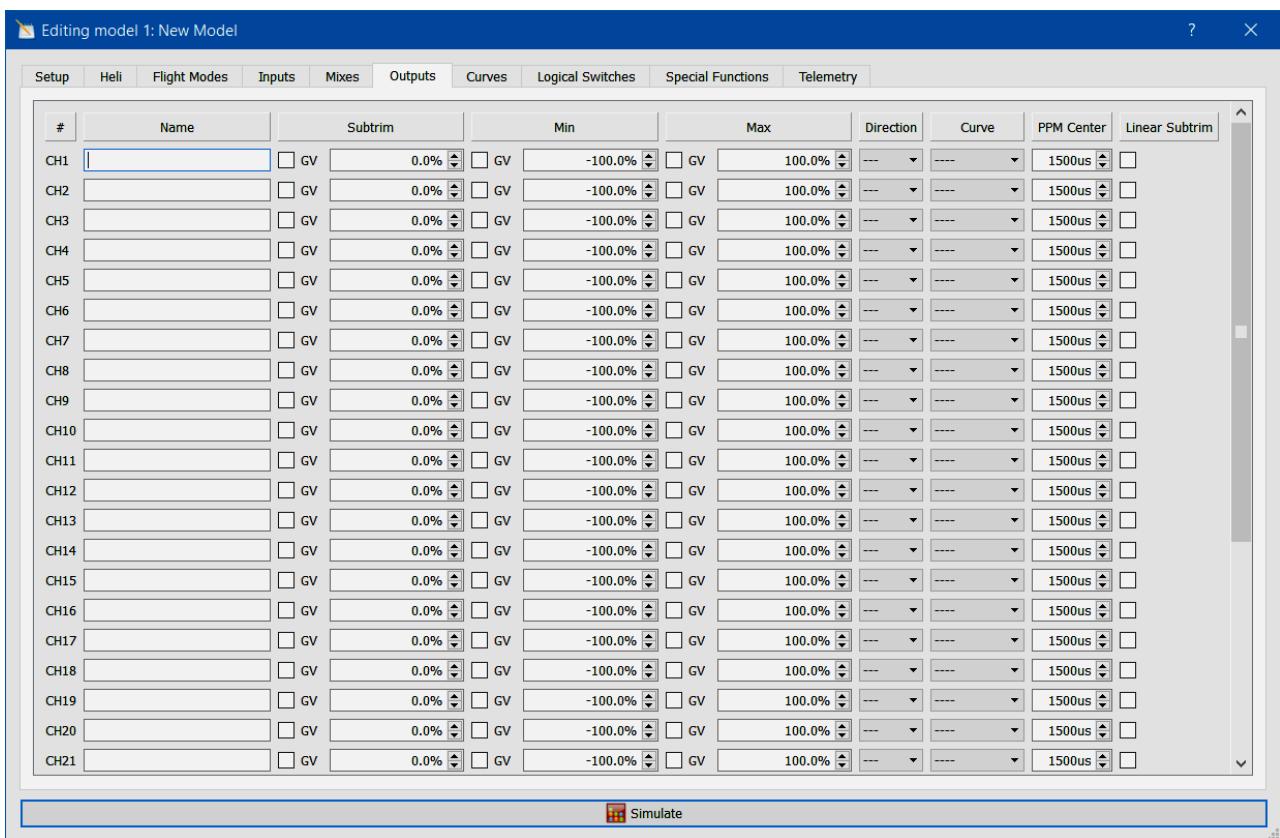
- What's the control surface I want this to act on? *Elevator, which is connected to CH2.*
- When do I want it to move? *When I move the throttle stick, in addition to whatever would already be present (usually the elevator stick).*

So you would simply go on **CH2**, and insert a new line with **Thr** as source. Type would be **Add** as the compensation needs to be added to the "normal" elevator response. As the required compensation is likely small, you will dial in a small weight, maybe 5%. On the ground with motor disconnected, you will check the elevator compensates in the correct direction. If not, you'll invert the weight to -5%.

You could then assign a switch, in order to be able to activate/deactivate it in flight to see if the amount of compensation is actually appropriate. If the correction is more complicated, you might want to assign and create a curve that matches what's required.

The Outputs Screen

The Outputs Screen



For each channel, we can define:

→ **Name**

A name, that will be shown on the mixer screen when the cursor is on a line belonging to that channel, on the channel monitor and on the failsafe settings page.

→ **Subtrim**

Trim and sub-trim have identical outcomes to the servo, but subtrim is set on the **Outputs** screen with the plane on the bench, while trim is adjusted from the transmitter trim buttons with the airplane in flight. Sub-Trim is used to achieve the best mechanical geometry of the servo/arm/linkage/horn physical set up to get control surfaces at their balanced neutral condition when the transmitter joysticks and trims are centred. Trims are in-flight adjustments to achieve “hands-off” straight/level flight of the aircraft.

Open TX The Model Editor

→ Min/Max

Low and high limits. These are "hard" limits, i.e. they will never be overridden. They also serve as gain or "end point settings", so reducing limit will reduce throw rather than induce clipping.

→ Direction

Reverses the servo travel.

→ Curve

Allows the inclusion of a curve to define the servo travel.

→ PPM Centre

PPM Centre adjustment. This is similar to the symmetrical subtrim, with the difference that an adjustment done here will shift the entire servo throw (**including hard limits**), and won't be visible on the channel monitor.

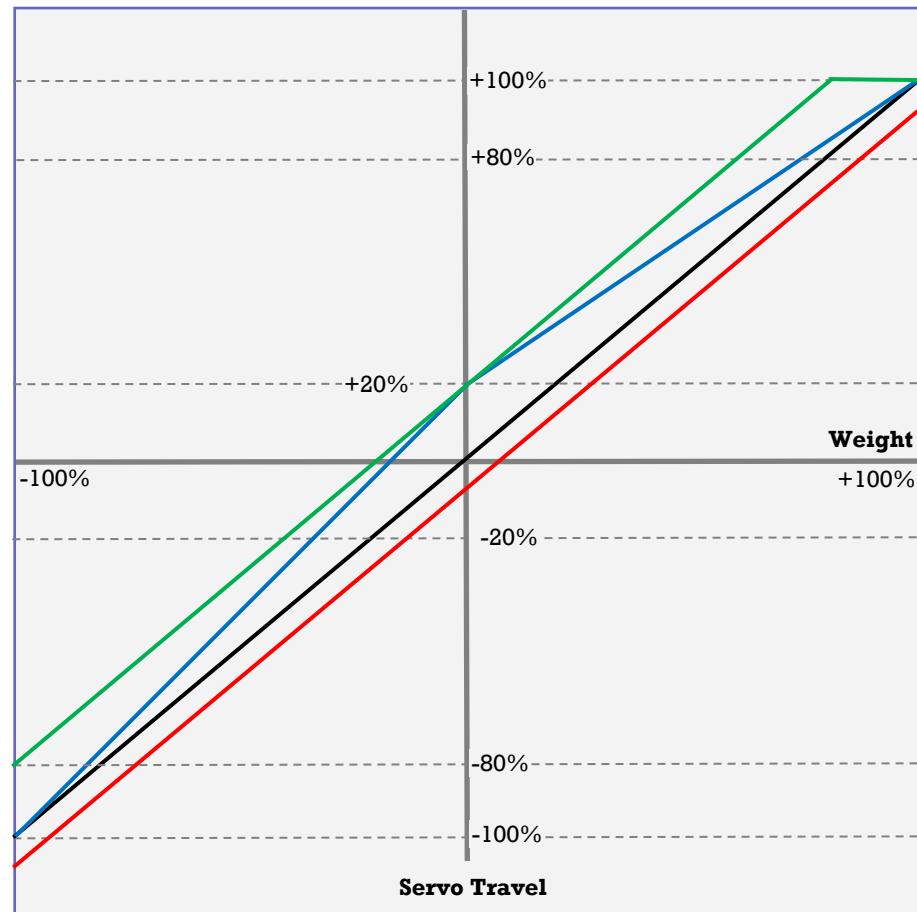
→ Linear Subtrim

If the subtrim behaviour is left in a linear mode then the subtrim acts just like holding the joystick that amount to one side of centre. But since the joystick still begins from its physical centre, if you continue to move the stick to its limit on that side, then the signal to the servo will max out before the stick does. If you move the stick to the opposite side, the stick stops before the servo signal reaches its maximum. To the trimmed side, you still have the full 100% signal available but it arrives earlier. To the side away from the trim, you have less than 100% signal available to the servo by the amount of the trim given to the other side. Thus the addition of subtrim in a linear manner causes you to lose a little something on both sides of the stick.

The final field to the far right of each line is called sub-trim mode and shows a default “ Δ ” symbol on the transmitter. This mode selection “ramps down” the stick-to-servo sensitivity and extends servo travel as necessary to cause both the stick and servo to max out together at the limiting values on both sides of the adjusted starting point. This default works well for simple set ups, i.e., multiple servos driven by a Y-cable or no mixes of other channels with this control channel. The variable stick-to-servo sensitivity can cause unexpected effects where mixes and curves are employed in the channel, so the “=” alternative choice is available to retain the linear relationship at the expense of the trade-offs described above for trim and sub-trim.

The Outputs Screen

The following diagram illustrates the respective behaviour of both subtrim modes and how centre adjustment compares to them:

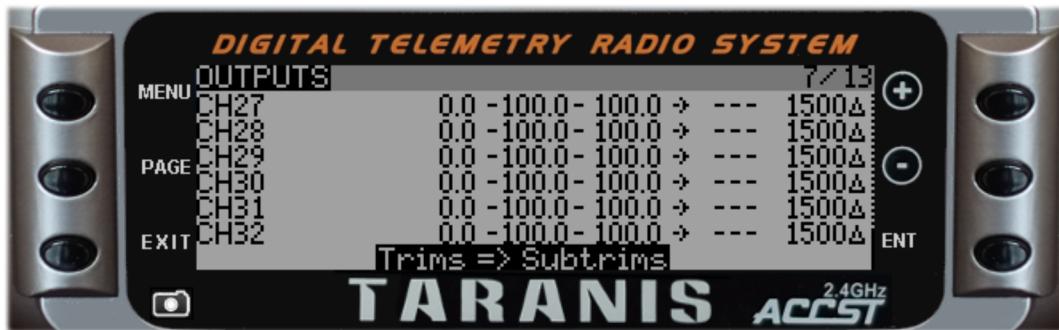


—	0% subtrim
—	PPM centre shift negative
—	20% non-linear (or Δ) subtrim
—	20% linear (or $=$) subtrim

Open TX The Model Editor

Trims to Subtrims

A final feature of the **Outputs** screen is the trims to subtrims option, only available via the transmitter screen not the **Companion**. Scrolling down the screen (or more often it is quicker to scroll up), there appears the **Trims => Subtrims** feature. A long press of the **ENT** on the **Trims => Subtrims** line will add all flight trims to their sub-trim values and reset the flight trims to zero. This can be handy at the field following a trim-and-balance flight. This feature fills the need for those who prefer to fly with the Trims zero'd out. Once subtrims have been set, if **OpenTX Companion** is used, and the models and settings are read from the radio, these subtrims will also be copied



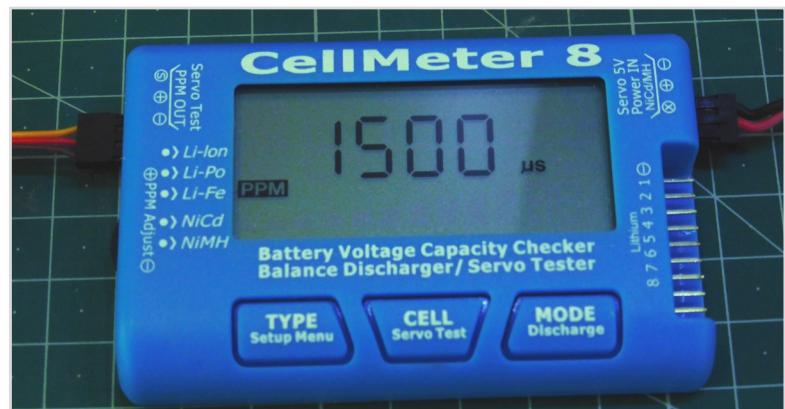
Notes on Servos with OpenTX

It soon becomes apparent that by using **OpenTX** one becomes far more familiar with the mechanics of how the transmitter, receiver and servo combination works in a model. It is, therefore, worth outlining some of the mechanics of servos here to ensure they are used to best effect.

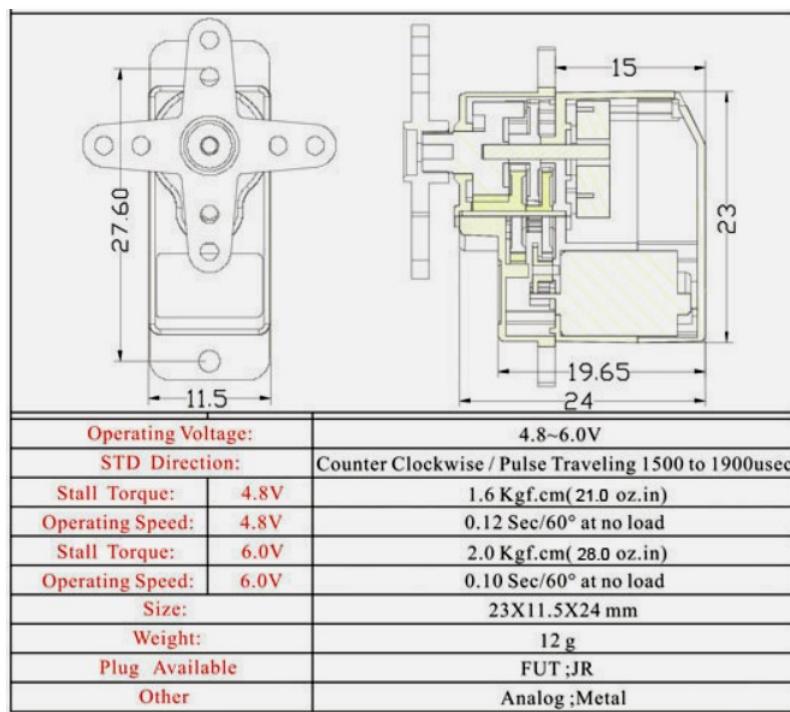
- **OpenTX** uses normal servo limits of -100% to $+100\%$. Not all radio systems do that. Spektrum, for instance, uses -80% to $+80\%$, and the extended limits on a Spektrum extend this to -100% and $+100\%$.
- **OpenTX** uses a PPM (Pulse Position Modulation) of $1500\mu s$ as the centre. The default 100% travel for **OpenTX** on the Taranis is $1500 \pm 512\mu s$, or $988\mu s$ to one side and $2012\mu s$ to the other side. Some radio systems use a slightly different centre point.
- If converting a model from another radio system, servo centres and travel limits will need to be checked carefully.
- Take great care if using extended limits with **OpenTX**. Extended limits can alter the range to $\pm 150\%$. Not all servos can cope with this. You might exceed the available travel in which case the servo will stall, and draw a much higher current, some servos will even rotate through 360° if their normal travel limit is exceeded! Also one should be aware that as the limits are extended, the servo travel becomes less and less linear due to the rotary motion of the servo arm. Usually, normal servo travel is $\pm 30^\circ$, however this varies between different makes and models of servo.
- If much trim or subtrim is required to obtain stable flight, then it is better to adjust the servo linkages rather than continue with the trim settings.

It is probably true that whilst learning to use **OpenTX**, one also learns a great deal more about the whole radio control process. The **Outputs Screen** is no exception, and a better understanding of how the servo works certainly helps here. **OpenTX** freely talks about pulse width and pulse centre. Instead of the normal little servo tester, a digital servo tester compliments **OpenTX** very well.

One like this will give the centre position precisely, as shown, or will rotate between fixed points, either 1000 μ s to 2000 μ s, or 500 μ s to 2500 μ s depending on how the device is set up using the menu system. Clearly the 1000 μ s to 2000 μ s setting is perfect for **OpenTX**. There is also a small knob on the side to manually rotate the servo. This one can set up the flight control centres and travel as one is building the model and connecting up each servo before a receiver is ever connected. Testing different servos using a device like this shows just how different many servos are. Using the wider 500 μ s to 2500 μ s setting one gets some very odd results. Older servos tend to move through 180°, newer servos can do some very strange things. Some start jittering badly when they get to 500 μ s, others will continue rotating through a full 360° when set to 2500 μ s.

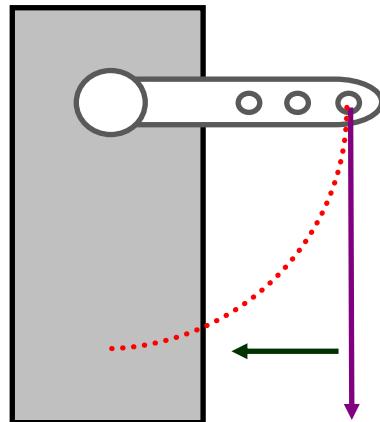


Wherever possible, one should look up the manufacturer's specifications. Here one can see the operating pulse width extends from 1100 μ s to 1900 μ s. This illustrates just how inadvisable it is to use extended limits with some servos and **OpenTX**. One can also see the servo arm travel is $\pm 30^\circ$. This is fairly normal and there is a good reason for this.



Open TX The Model Editor

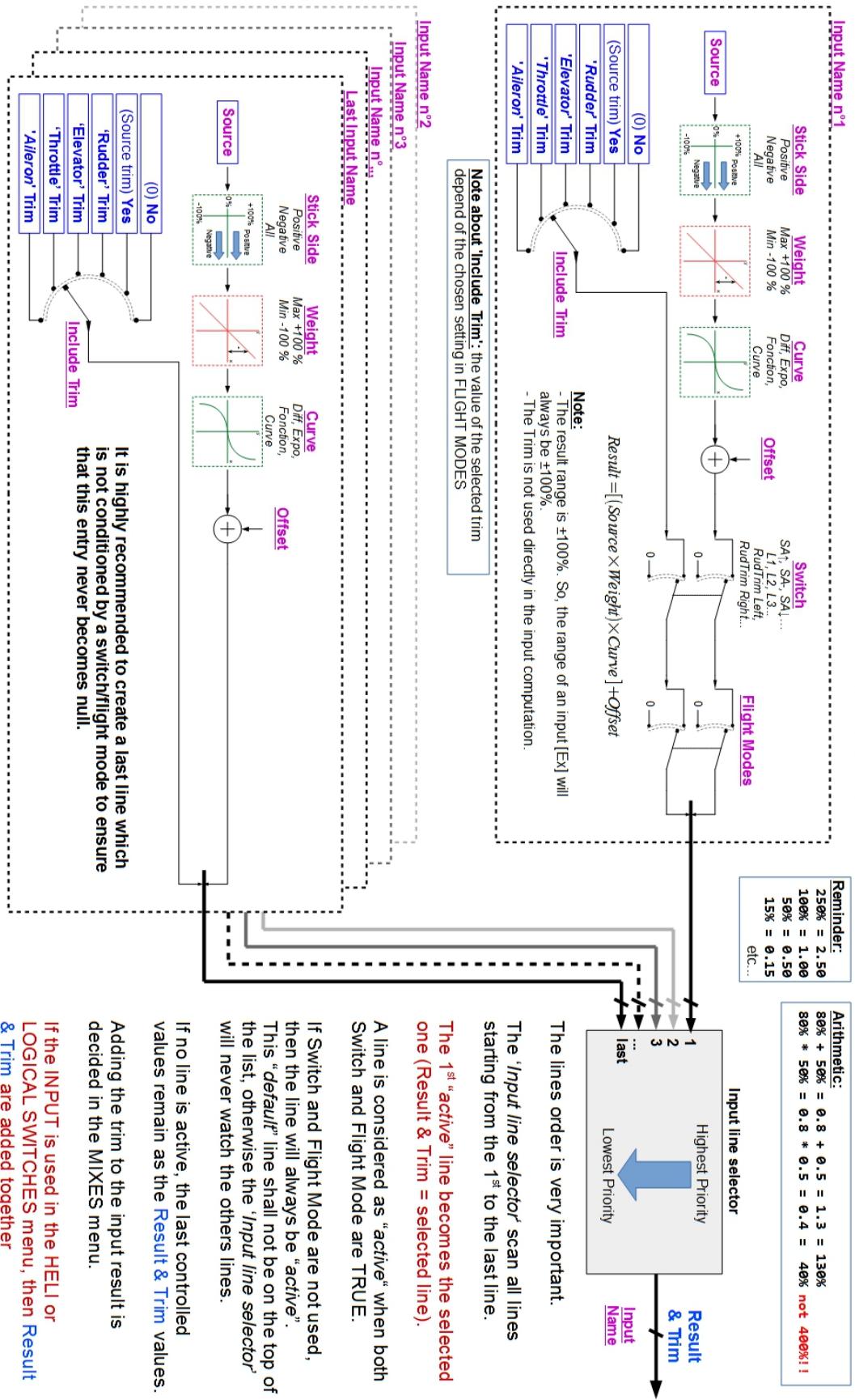
Consider the arm of a servo. As it rotates clockwise the arm moves down, as shown by the purple line, but at the same time it moves left as shown by the green line. Thus the rotary motion of the servo arm does not give linear motion of the servo linkage. The graph below shows the rotation angle to the distance the servo linkage moves (purple line). As can be seen, it is fairly linear up to about 30°, but after that the distance moved becomes less proportional to the rotation angle. This chart also ignores the movement in towards the centre of the servo, (i.e. the green line) which can be ignored on long servo linkages, but will have an adverse effect on short linkages or when using snakes.



The distance moved is given in terms of the distance from the centre of the servo arm to the linkage hole being used. Thus rotating through $\pm 30^\circ$, or a total of 60° will give a good approximation for linear travel, and the servo linkage travel will be half the distance from the servo arm centre to the hole the servo linkage is connected to. Also the sideways movement of the linkage will be minimised.

Inputs Diagram

INPUTS Diagram



This diagram reproduced by kind permission of Sébastien Charpentier

Open TX The Model Editor

MIXES Diagram

Direction
of
calculation

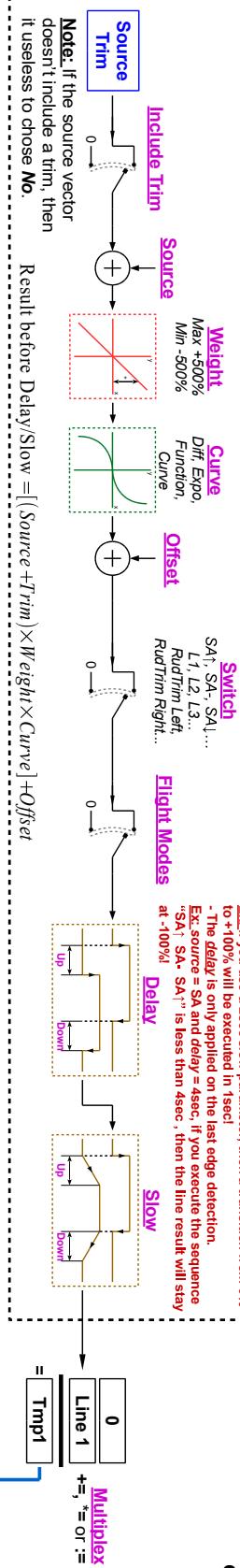
Name n°1

Notes about 'Delay' and 'Slow':
 - The specified time for the **SLOW** function is corresponding to a transition from -100% to +100%.
 Ex: If you use a 2sec slow parameter, then a transition from 0% to +100% will be executed in 1sec!
 - The delay is only applied on the last edge detection.
 Ex: source = SA and delay = 4sec, if you execute the sequence "SA↑ SA, SA↓" is less than 4sec, then the line result will stay at -100%!

MIXES Diagram

Direction
of
calculation

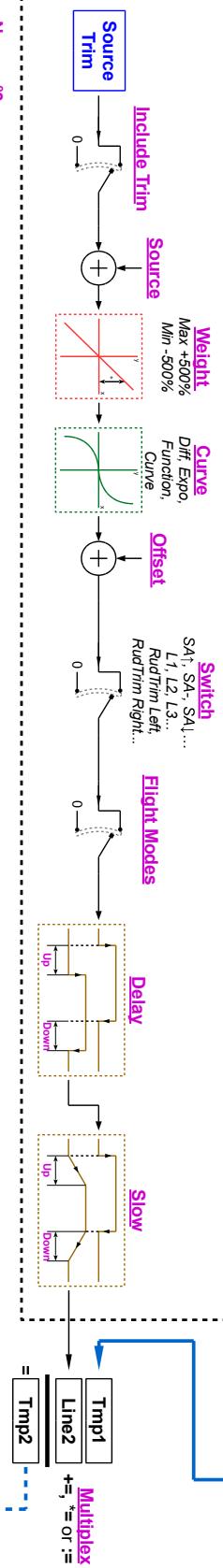
Name n°2



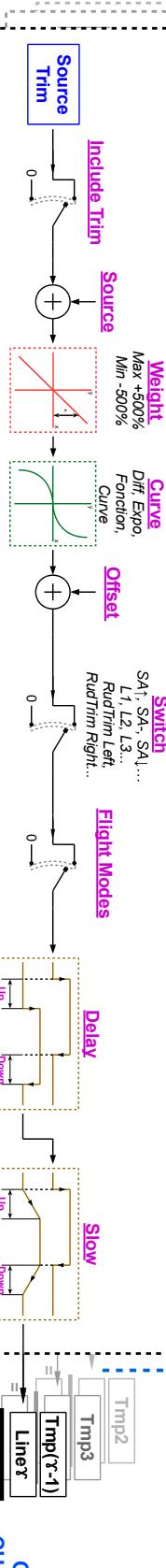
Result before Delay/Slow = [(Source+Trim)×Weight×Curve]+Offset

Note: If the source vector doesn't include a trim, then it is useless to choose No.

Name n°3



Name n°4



The order of the lines is very important. The computation achieved at each line is done according to the temporary result of previous lines.

3 kind of operands is available : **'ADD' +**, **'MULTIPLY' *** or **'REPLACE' :**

When using 'REPLACE', the previous temporary result is replaced with the current line. This multiplex is very useful to create a "throttle cut" feature.

This diagram reproduced by kind permission of Sébastien Charpentier

Reminder:	
250%	= 2.50
100%	= 1.00
50%	= 0.50
15%	= 0.15
etc...	

Remarks: The final/temporary result range is ±500%. Clipping is done if the result exceed 500%.

However the range used by the OUTPUTS menu is ±100%, even if the 'Extended Limits' option is ticked!

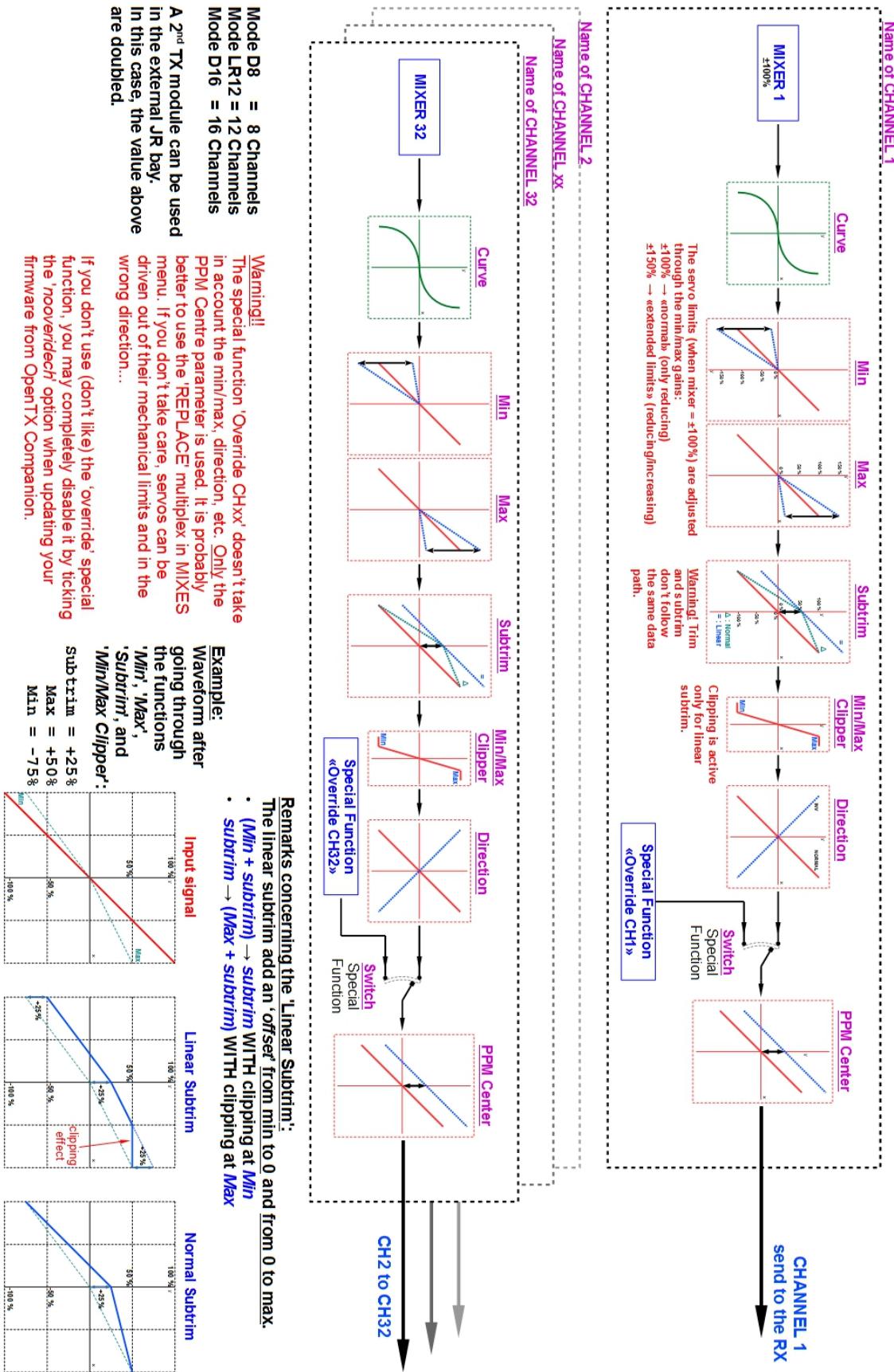
Arithmetic:

$$80\% + 50\% = 0.8 + 0.5 = 1.3 = 130\%$$

$$80\% * 50\% = 0.8 * 0.5 = 0.4 = 40\% \text{ not } 400\%!!$$

Outputs/Servos Diagram

OUTPUTS/SERVOS Diagram



This diagram reproduced by kind permission of Sébastien Charpentier

Open TX The Model Editor