Contents

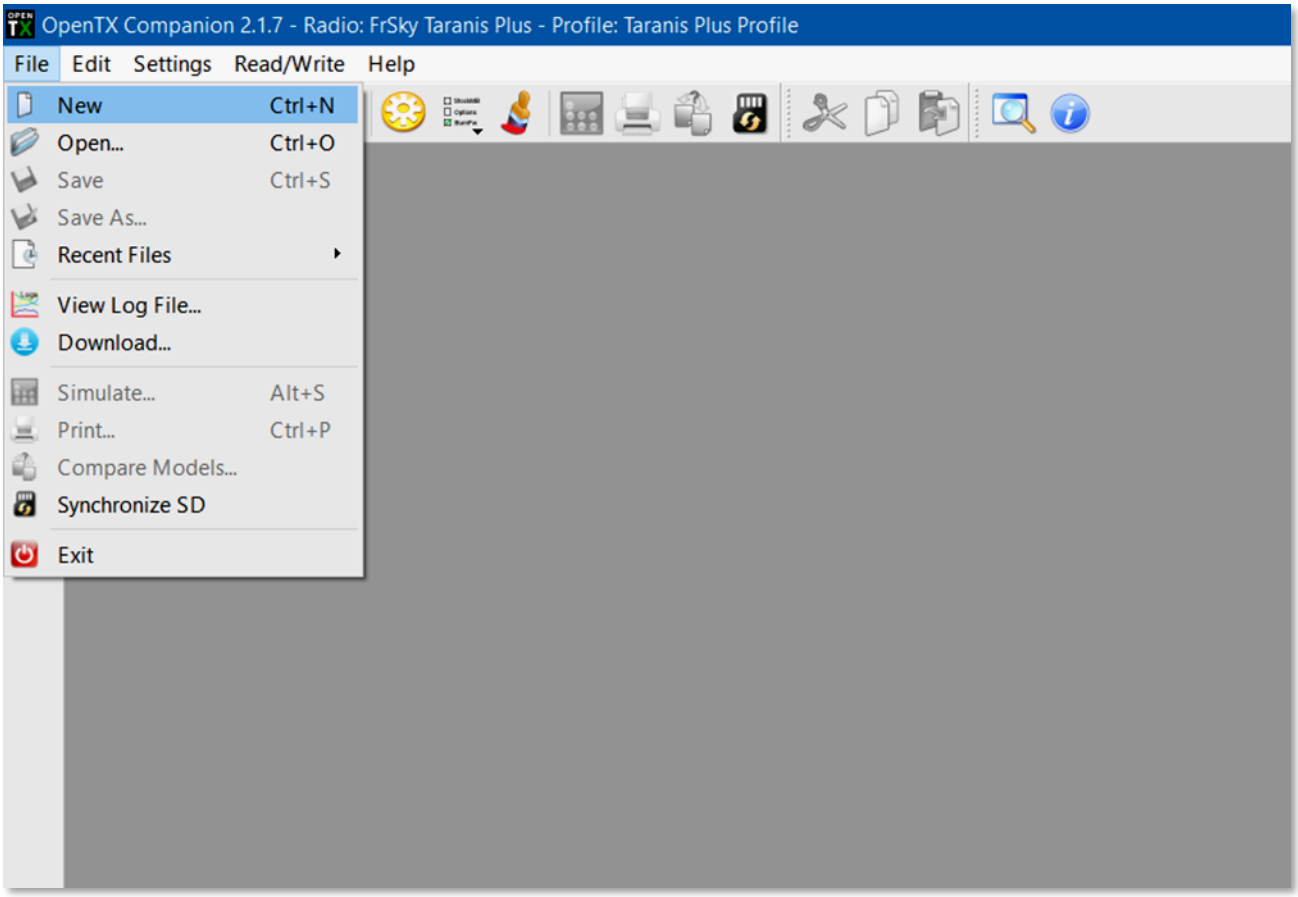
Continued overleaf ….

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

Contents, Continued

Accessing the Model Editor

Using the **OpenTX Companion**, a model file has to be loaded first, using either **New**, **Open** or **Recent files**:

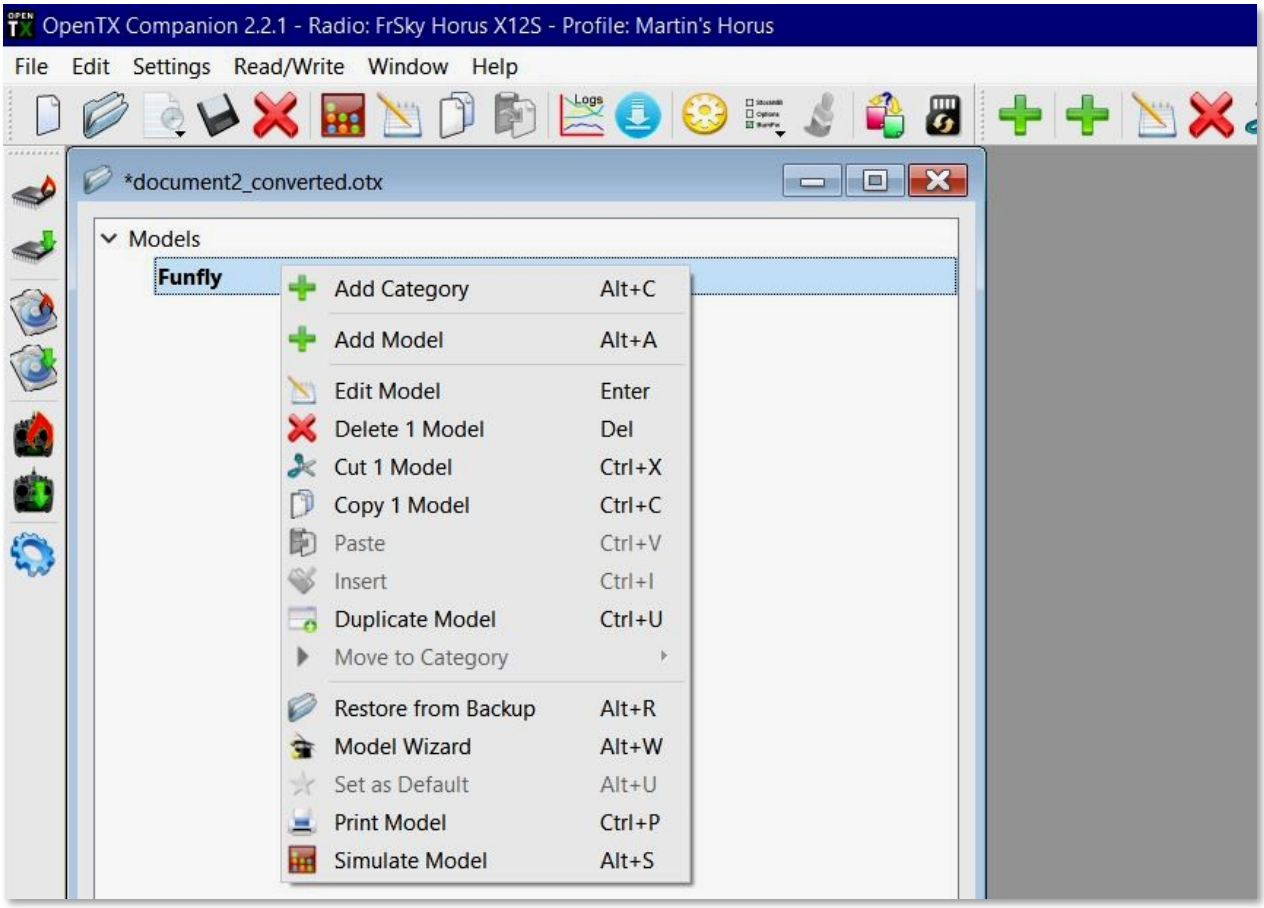


Taranis:

The **Model Editor** can be selected on the Taranis radios 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.

Horus:

The **Model Editor** can be selected on the Horus radios by pressing the **MDL**. **PgUp** and **PgDn** will move through the screens.

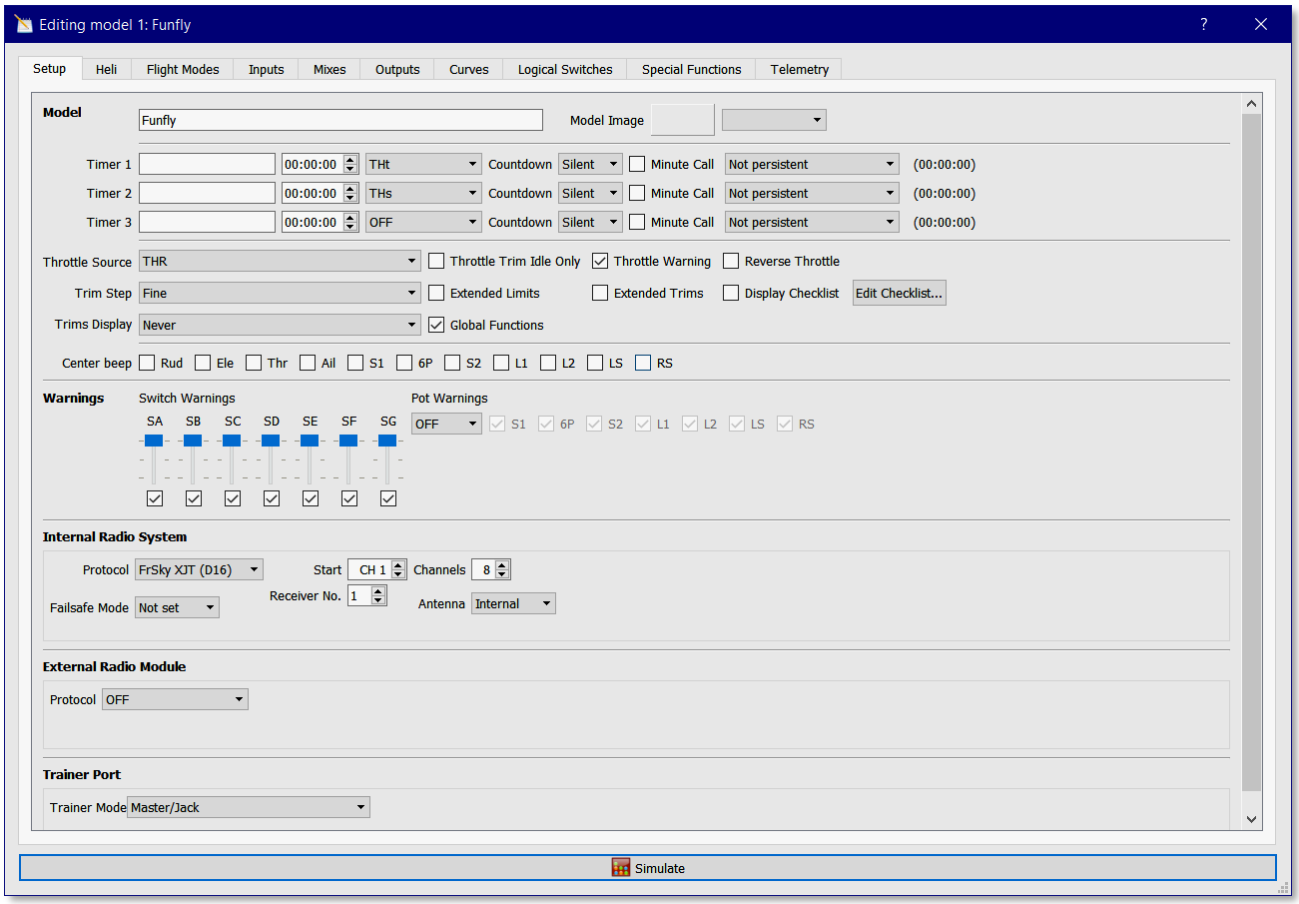


To get to the **Model Editor**, right click on the model to be edited, and select **Edit Model**. This will bring up the editing window. (Note this is a screenshot from the Horus Companion, not all options are available for all radios.) A simpler way is to double click on the model to be edited. The **OpenTX Companion Model Editor** comprises 10 separate windows, selected by the tab at the top of the main editing window. It differs to the transmitter screen version in several minor ways:

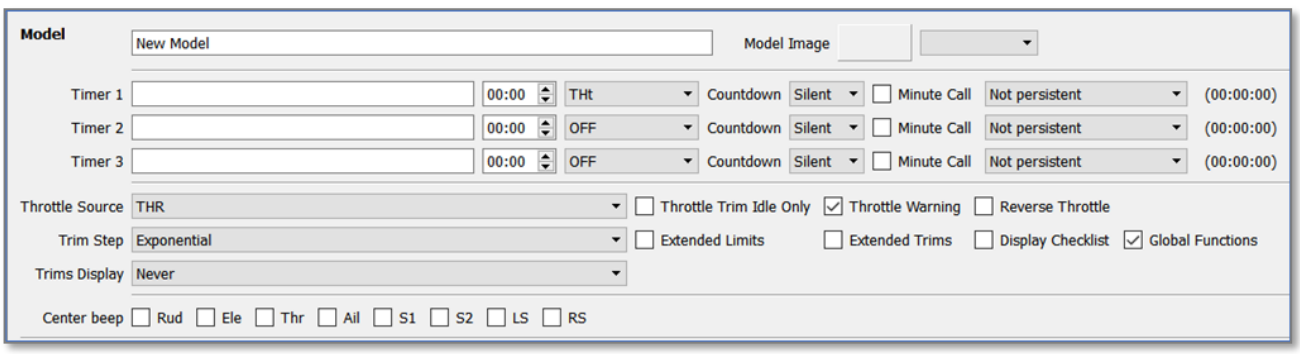
- Screen 6 on the Taranis, or screen 5 on the Horus, on the radio editor is called **Mixer**, on the **Companion** it is called **Mixes**. These screens do the same job.
- Screen 9 on the Taranis, or screen 8 on the Horus, on the radio editor is for **Global Variables**. On the **Companion** the **Global Variables** are included on the **Flight Modes** screens.
- Screen 12 on the Taranis editor or screen 11 on the Horus 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 radio screen. The functions that appear on the **Companion** screen also appear on the **Radio menu** screen, although some are in different places and under different headings.



Model Images are covered in more detail in the **How-To Part 1** section. Do note that a copy of the model image needs to be on the SD card, and on the mirror image on the computer for the simulator to work correctly. The drop down box on the **Companion** is case sensitive. Using the **radio menu** it is not.



Timer 1, 2 and 3

Timer 1, 2 and 3 each have the same features.

- 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 source** is not at -100. This value does not seem to be critical with the timer starting when the value increases to -75.
- **THt** starts the timer the first time the **throttle source** is opened.
- **TH%** counts up/down as a percentage of the full **throttle source** 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 give an announcement (a beep, haptic, or say the time) at 30 seconds then 20 seconds and count down in seconds from 10 seconds. It can be used in conjunction with the minute call.

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

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.** This setting dictates where the timers will get their start information. It also dictates the source for the throttle warning.

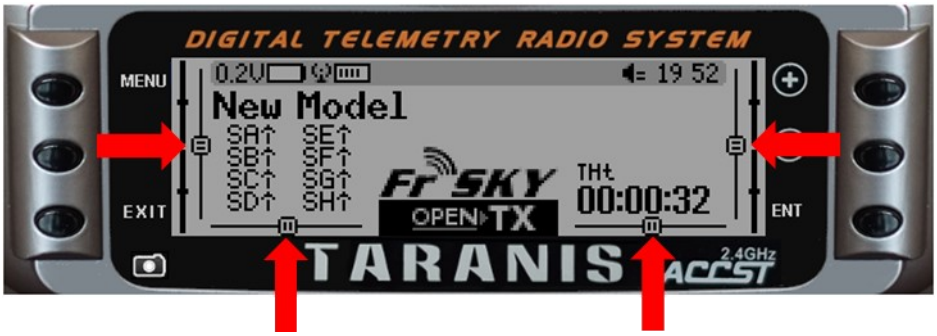
Trim Step

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

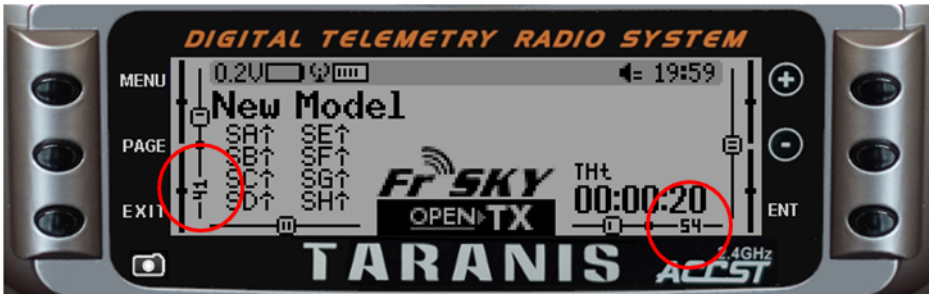
For a new model, exponential is very useful. Once a model is well trimmed out, changing the setting to fine or extra fine, will allow for small changes of trim. See also the “Instant Trim” function.

Trims Display

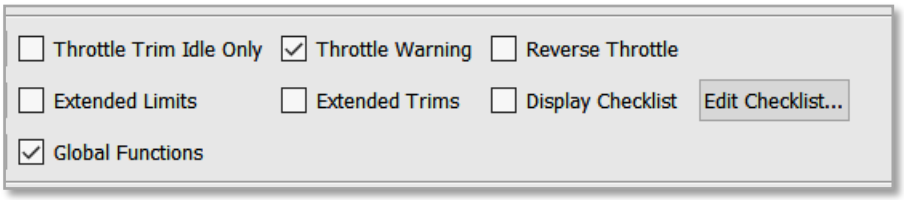
This has three settings: **Never, On Change, Always**. This appears a somewhat odd option, as the trims are always displayed on the transmitter screen:



However, with the trims display set to **Always** or **On Change**, the *values* of the trim settings will be also shown. Note, these are not weights, but are the actual value in the range ±512. Thus a trim value of +100 approximates to a weight of +20%.



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



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.

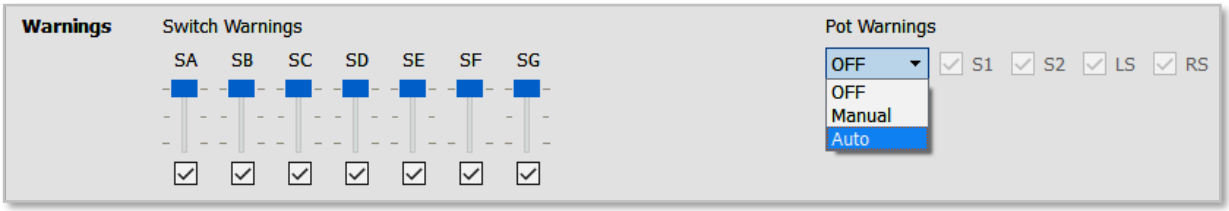
Extended Limits will extend the servo limits beyond their normal 100. With this box checked, travel limits in the **Outputs** screen can be increased to 150%, allowing PWM signals of 1500 ±768 µS to the servos. Maximum weights available in the **Inputs** screen and **Mixes** screen are unaffected by **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 ±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 on the Taranis. The Horus uses **PgUp**, **PgDn** and **RTN**. Making a checklist is covered in the **How-To Part 1** 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.

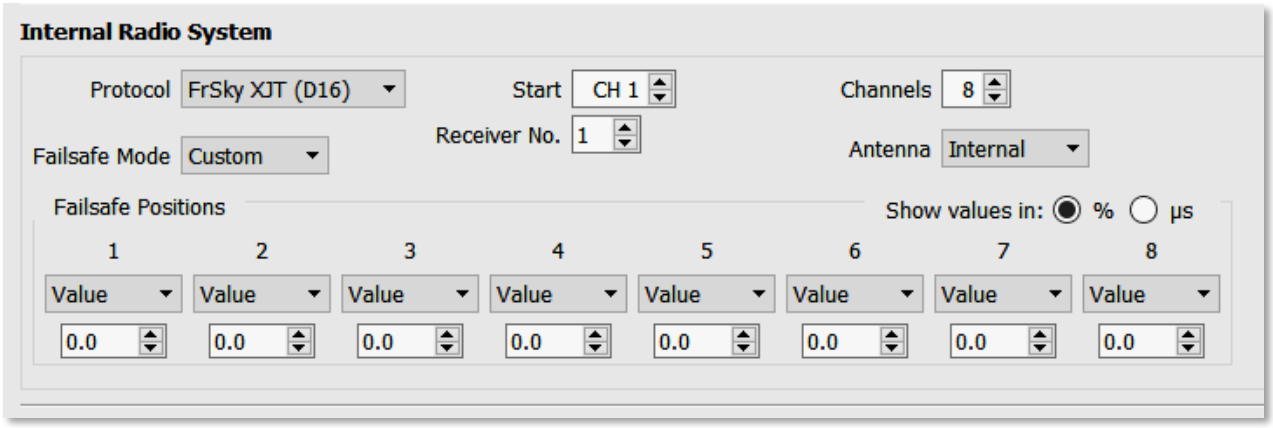
Switch Warnings will give a screen warning when the transmitter is switched on or when a new model is selected if the switches are not in the set positions. This is a useful safety feature for, say, a throttle disable switch on an electric model, or for a gear switch. However, simply leaving them



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

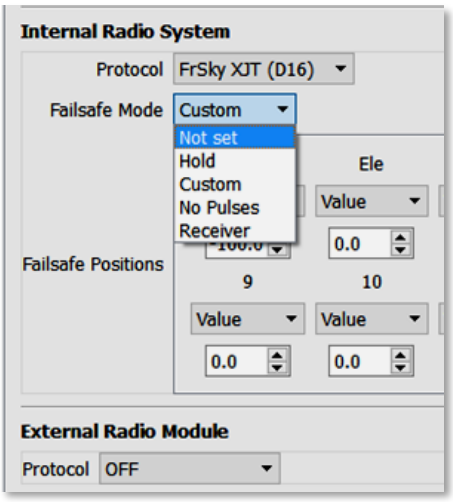


The **Internal Radio System** refers to the XJT module inside the transmitter which sends the appropriate signals to the model. The software for the XJT module is provided by FrSky, and is not part of the **OpenTX** software. There are 3 protocols available:

- **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.

Another feature of the **Internal Radio System** is only found on the Horus radios. On the X12, this allows either just the internal antenna to be used or the internal and external antennas to be used together. On the X10, that feature is slightly different. It allows **either** the two internal antennae **or** the external antenna. At the time of writing, this shows incorrectly on the Companion.

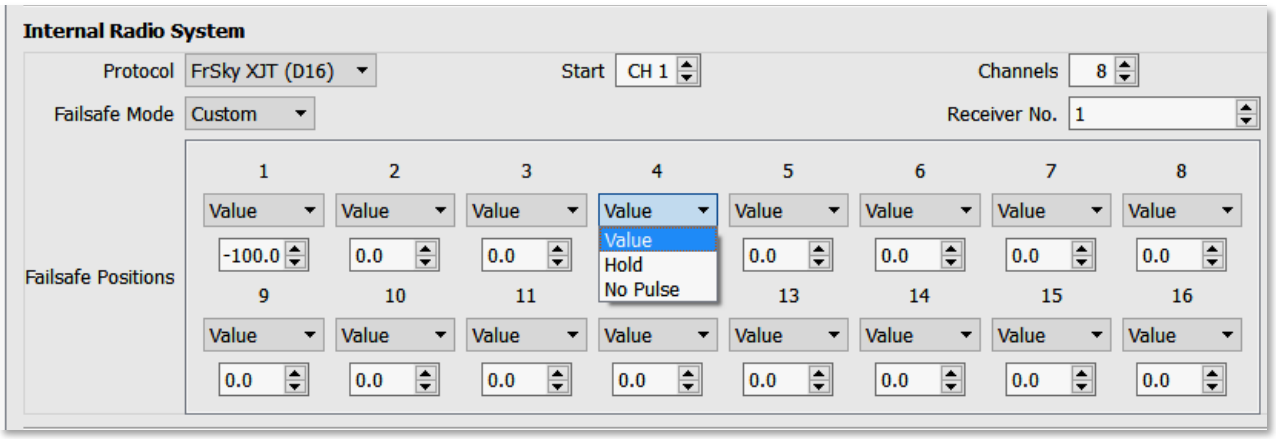
The Failsafe Mode



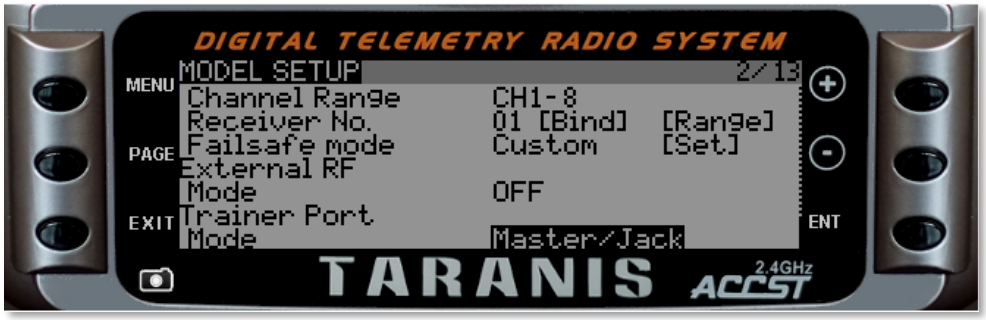
This is only available when the **Internal Radio System** is selected and is a very important function that should always be set. Indeed if a failsafe is not set, a warning will appear every time the model is selected on the transmitter. There are 4 options:

- **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

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.

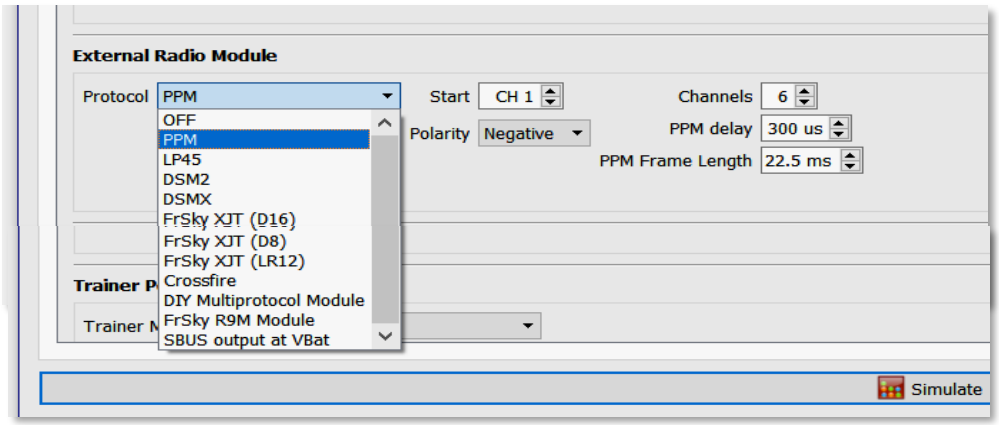
Range

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 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 RSSSI 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.

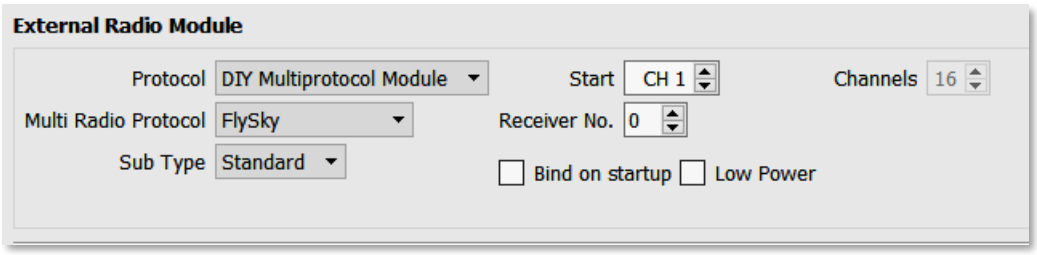
For the FrSky system the **RSSI** should read about 100 at 1m and every time you double the distance between the transmitter and receiver, the **RSSI** level should drop by 6. That's why it goes down quickly at first then less and less as the model flies away since you have to double the existing distance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

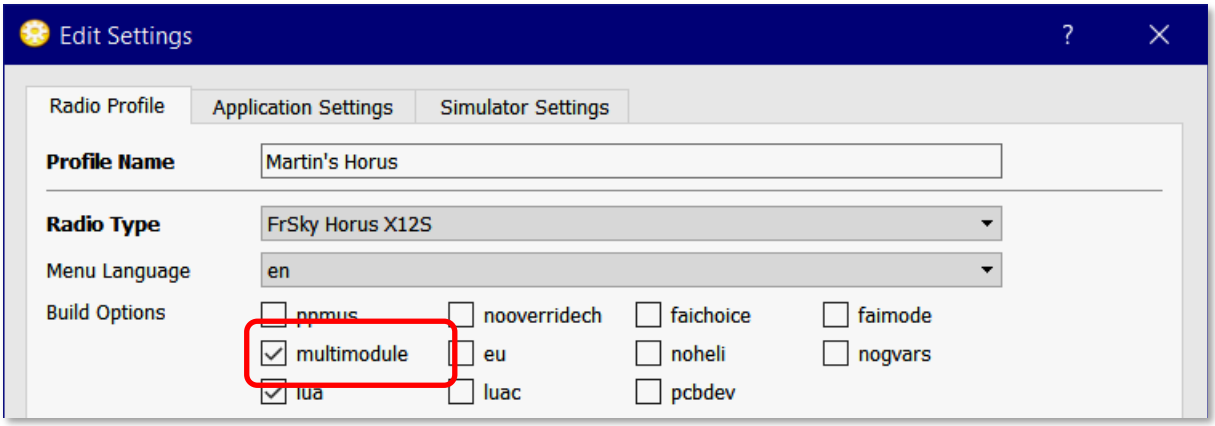
The External Radio Module



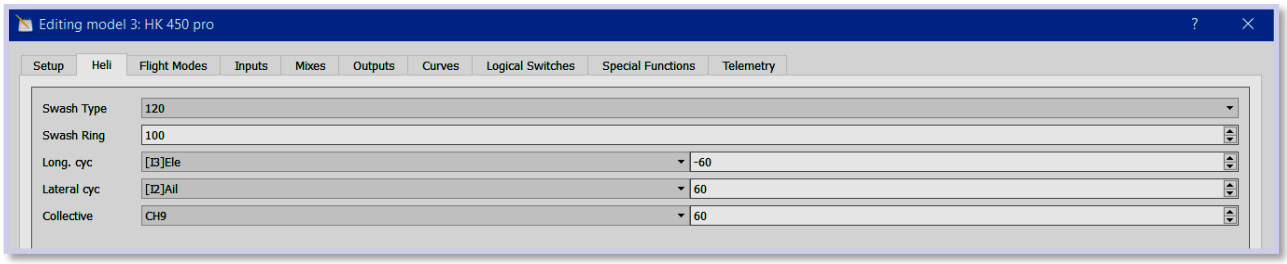
On the FrSky radios, one can also have external radio modules which plug in at the back of the case to talk to other makes of receiver. The module plugs inside the X9E radio. This module can be configured by turning the **Internal Radio Module** off then selecting the appropriate protocol for the external module. The settings available depend on the module type.

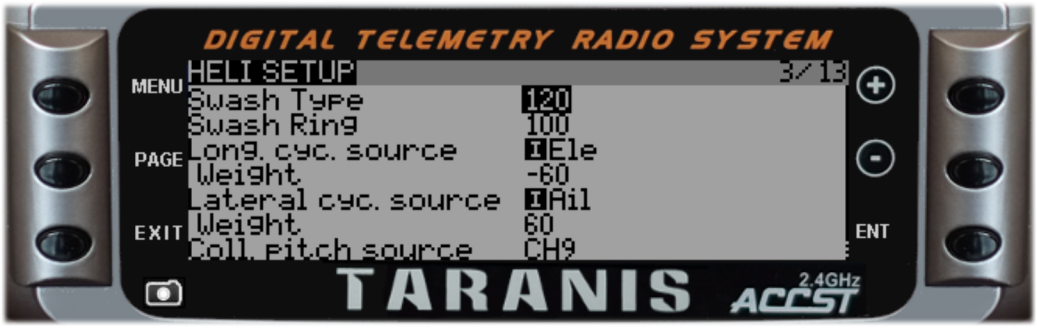
The **DIY Multiprotocol module** is a development which allows one module to have a configurable range of protocols to suit different makes of receiver. Over 30 main protocols are currently supported, and within each of these, several sub types may be available. In order for this protocol to be available, the **multimodule** build option must be selected in the **Radio Profile** (see section 4).





The Heli Screen





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

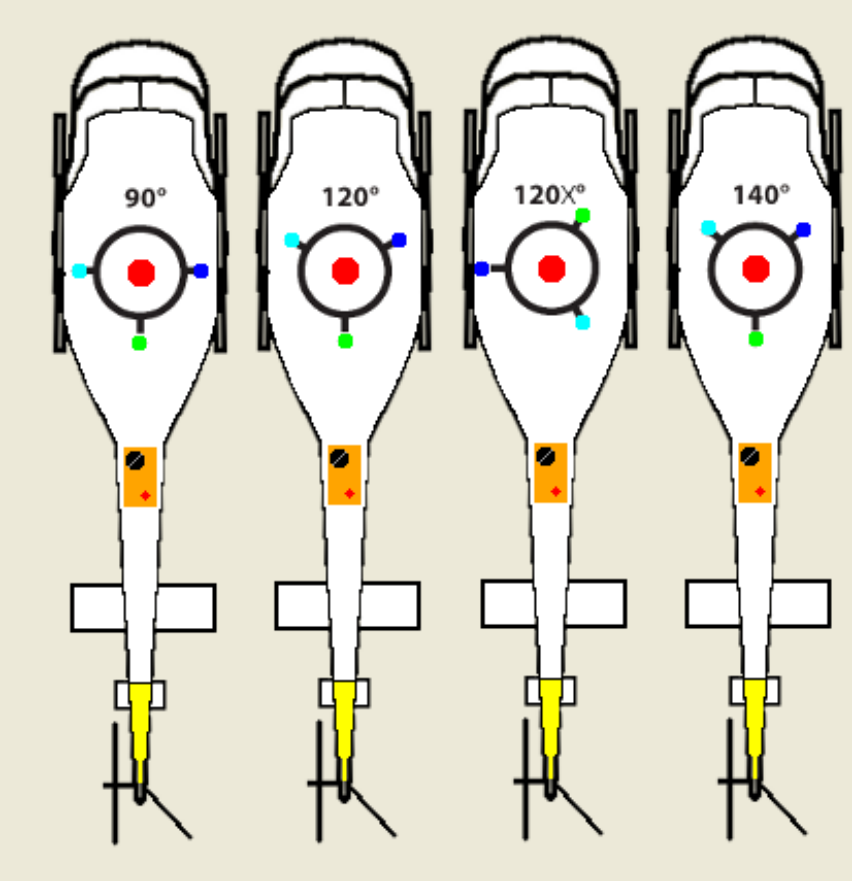
The inputs of this mixer are the aileron and elevator joysticks, plus the virtual channel selected in "Collective Pitch Source". This channel would see entries added on the **MIXER** page for one or more pitch curves to adjust the weight of the collective. The outputs of the CCPM mixer are **CYC1**, **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**

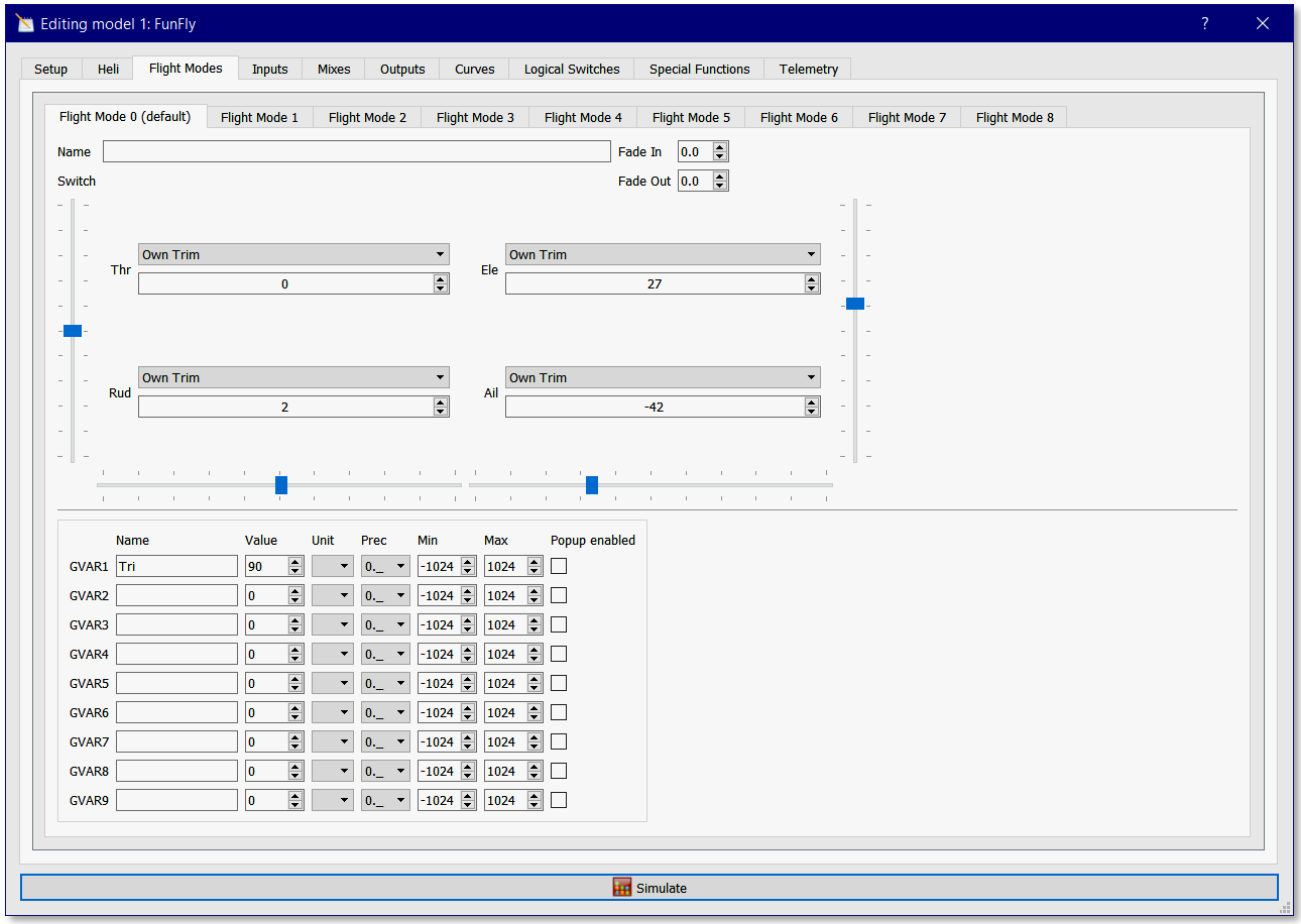
Swash Ring actually crops the "square" nature of the aileron/elevator stick to a circle. If the joystick is moved fully left from the centre, this gives a movement with an amplitude of 100%. It is the same if the joystick is centred and moved fully down. But, as the joystick can do a square, if the joystick 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 ±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.

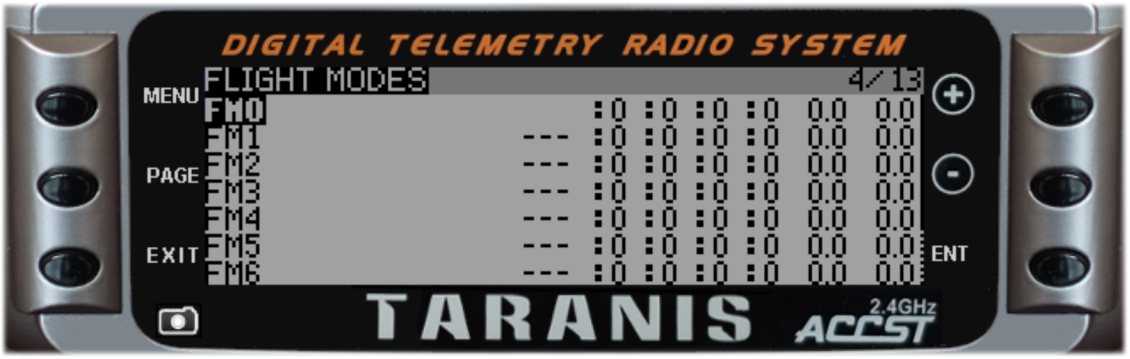
Setting **Swash Ring** to 100 will "round off" the corners, and never allow an input of more than 100% total deflection with the joystick. Setting the **Swash Ring** to less than 100 will reduce the overall servo movement further, however it is probably better to adjust the servo linkages to reduce travel,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

The Flight Modes Screen

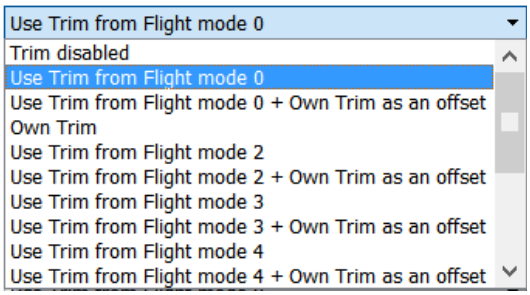




Nine flight modes including the default one are available for use. Each of them has the following:

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

- 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.

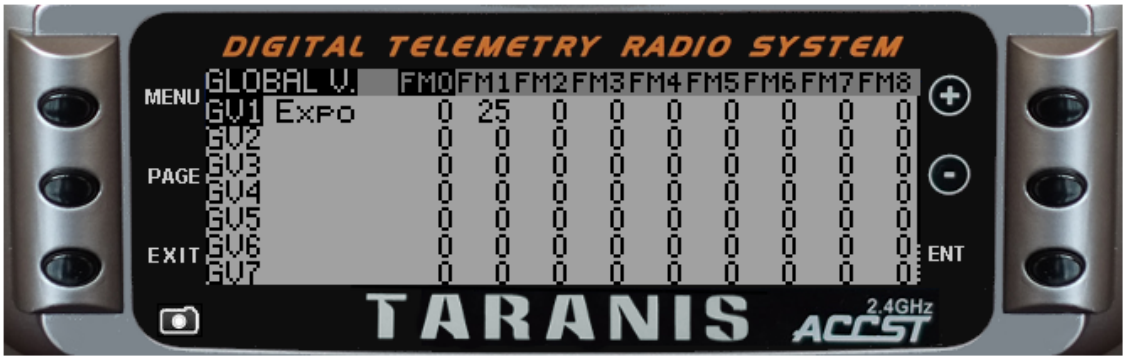


The priority of the flight modes is such as the first FM of 1-8 that has its switch ON is the active one. When none has its switch ON, the default FM0 is active.

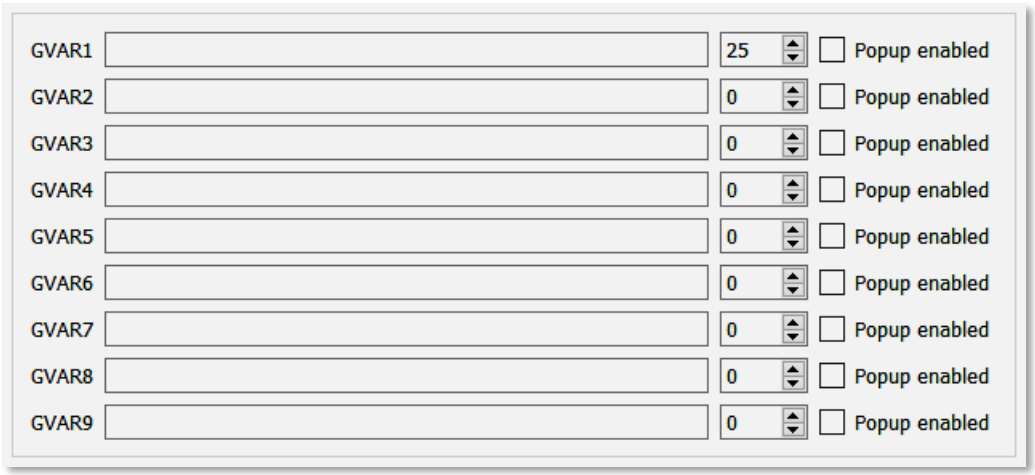
Global Variables

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. **Global Variables** are global to a model not between models. 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.

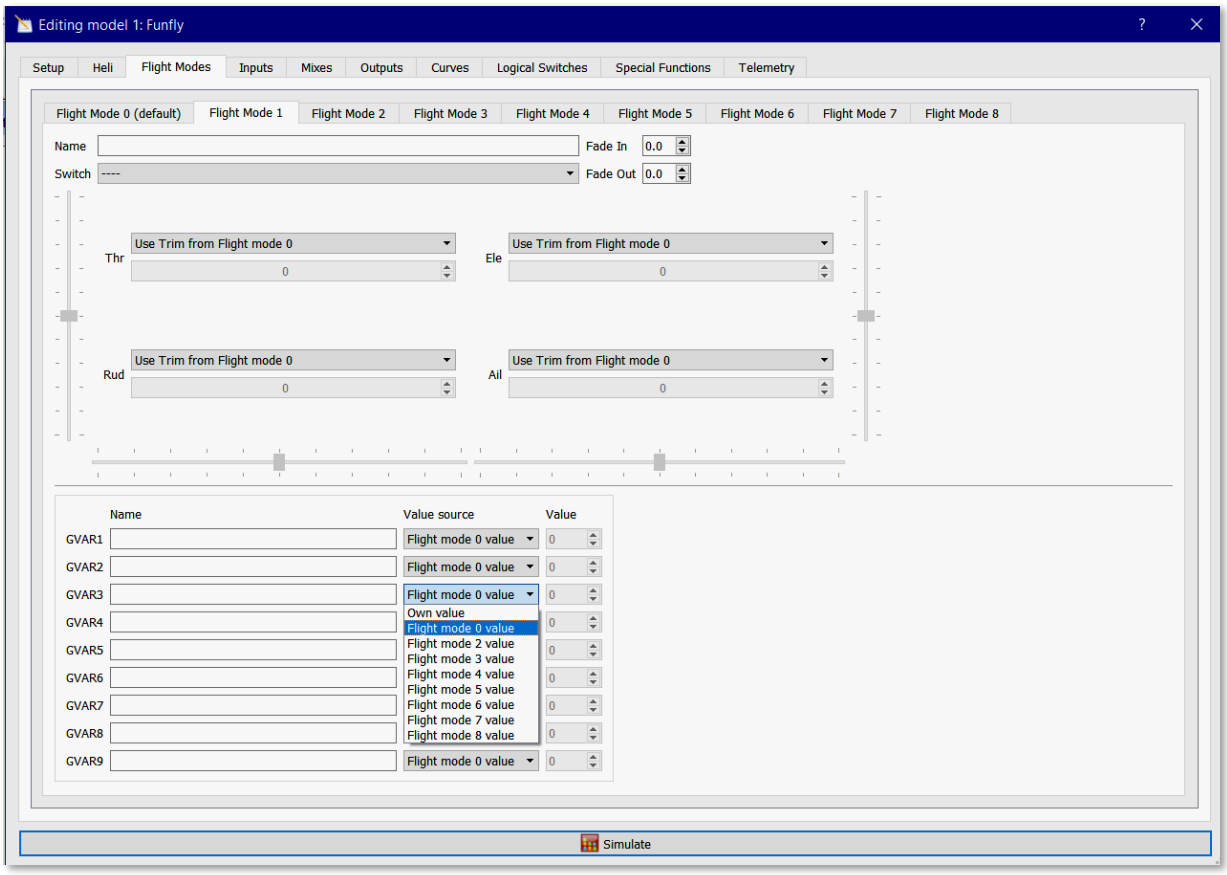


The global variable options differ between flight modes. The following screenshot is using flight mode 0.

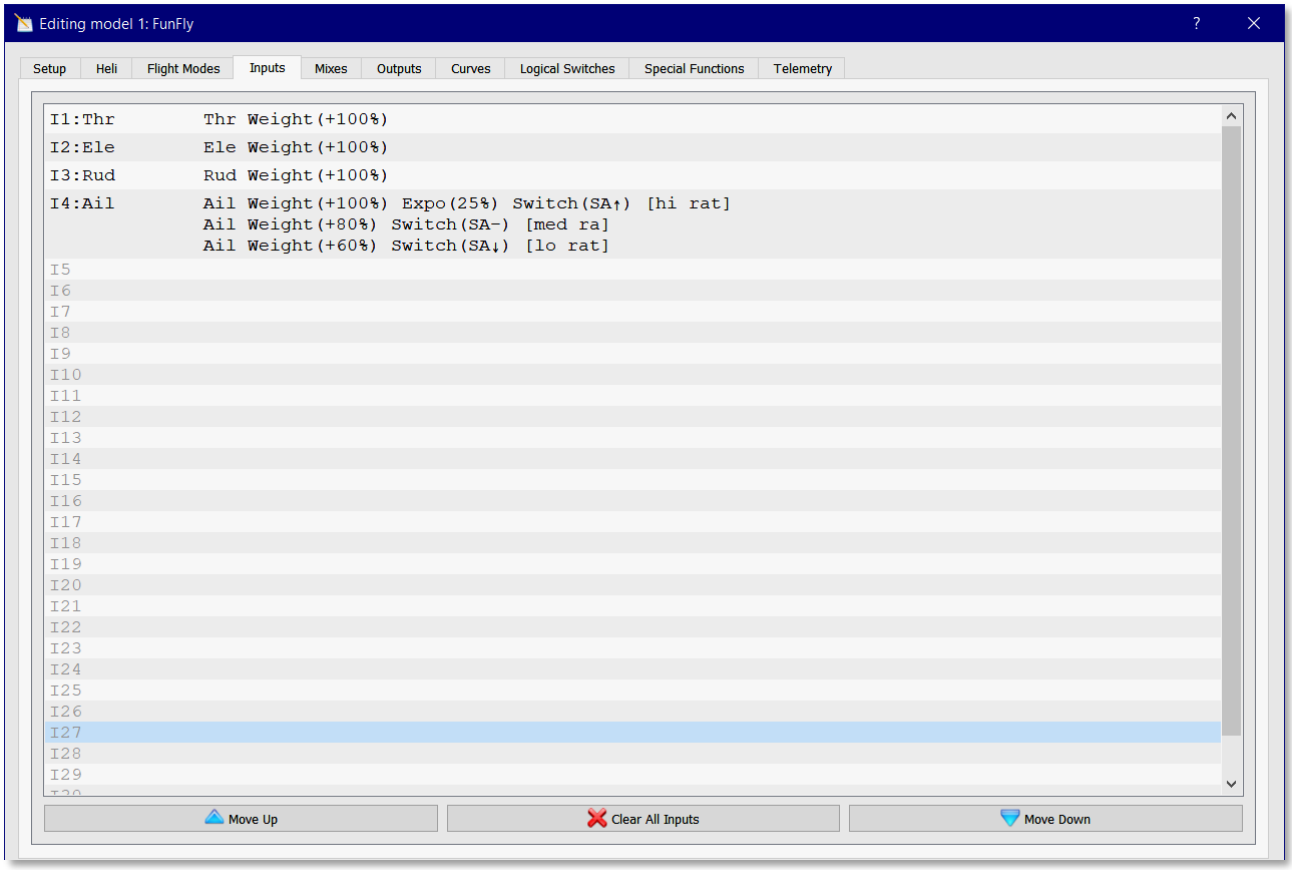


On the **Companion**, **Global Variables** appear as above. If popup is enabled, the value is displayed on main screen when the **Global Variable** is edited.

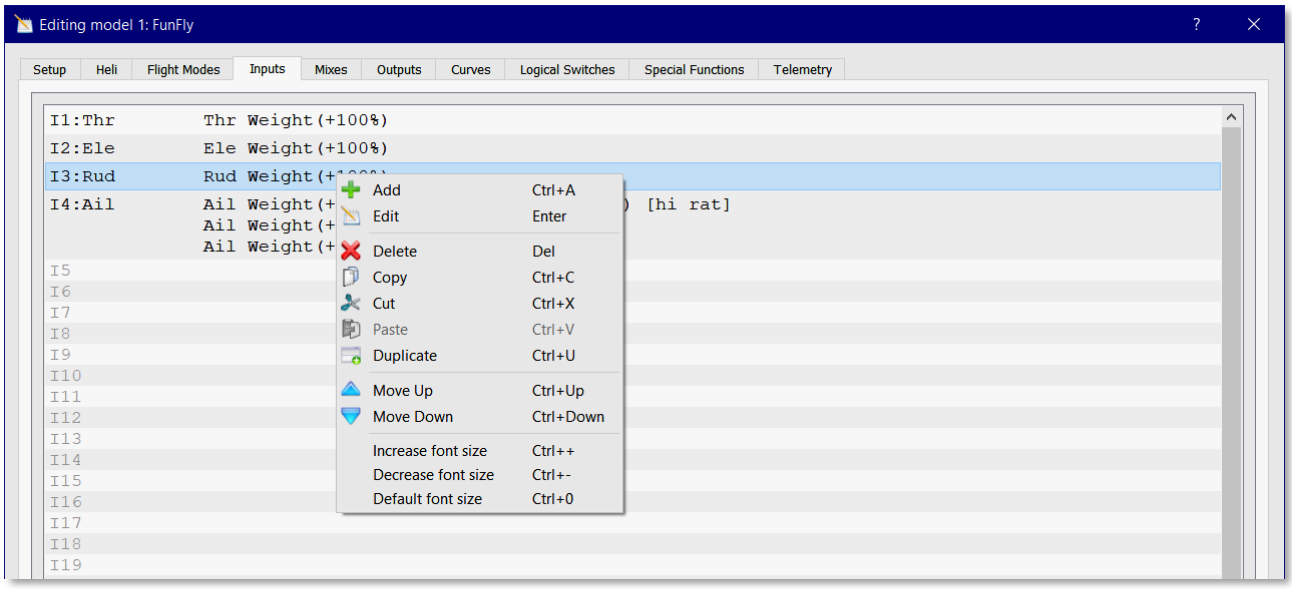
Flight modes 1 to 8 have an extra drop down box, **Own value** available. This box enables the global variable to take the value from any of the other flight modes.



The Inputs Screen



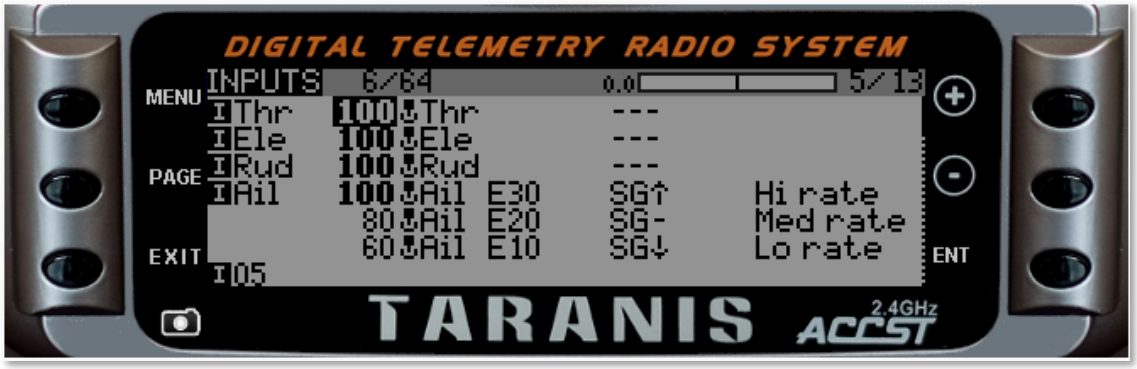
The **Inputs** screen is where the transmitter controls are linked to the inputs that will be manipulated by **OpenTX**. The left hand column shows the inputs defined, highlighted with a capital **I**. The next column shows the actual transmitter controls that are linked to these inputs; in this case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



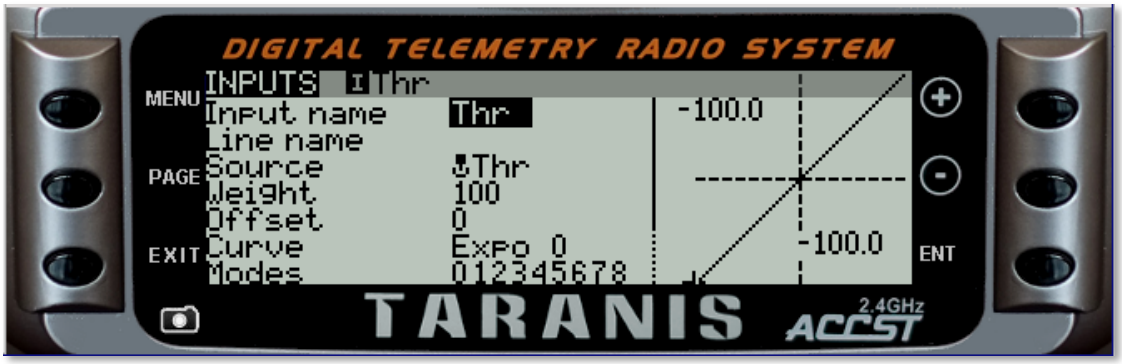
These are largely self-explanatory, and some lead to further sub-menus.
Toggle highlight will highlight the particular channel, and all instances where this channel is shown. Useful with complex model setups.

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.

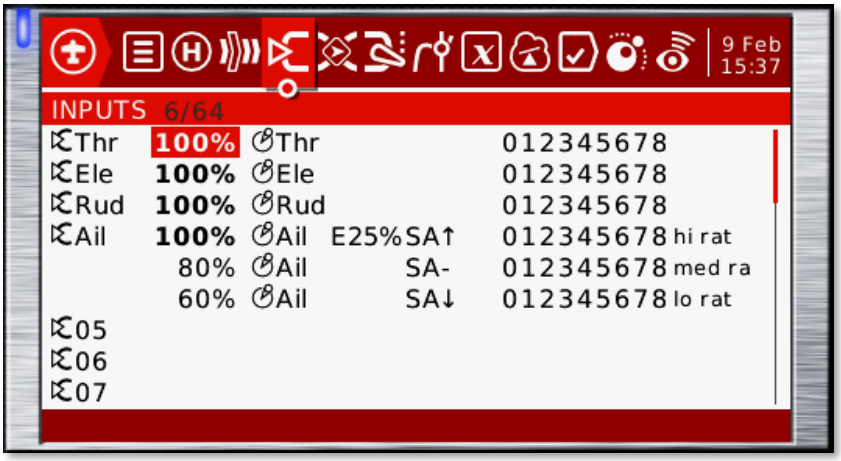
Programming on the transmitter also gives these menus, using a combination of long and short presses will allow the various functions to be accessed.

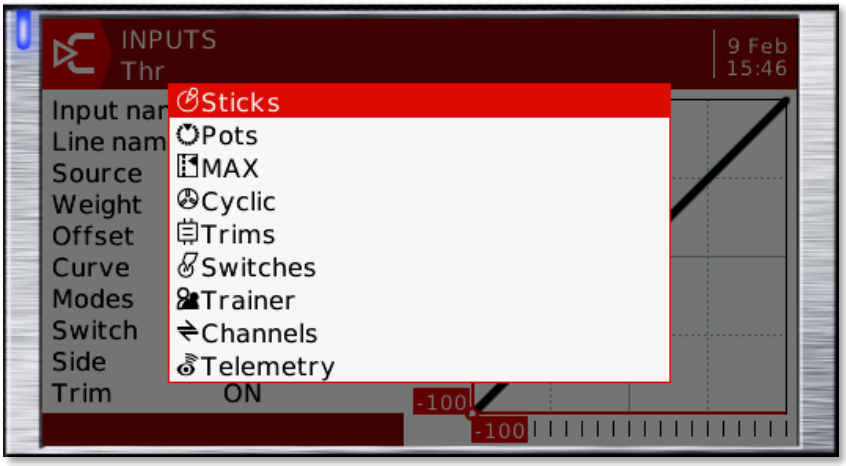


A long press of the **ENT** button pulls up a series of options including an edit option. This opens up another window with features that match the **Companion** edit window. :

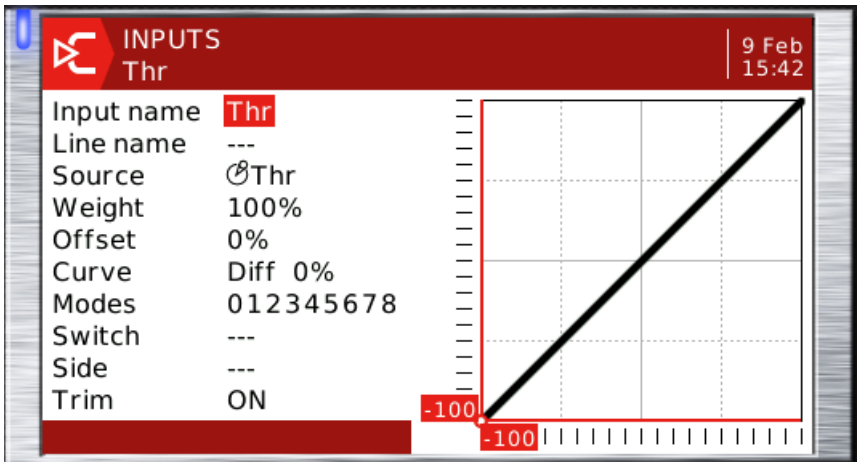


The Horus has a similar screen though the screen symbols are different.

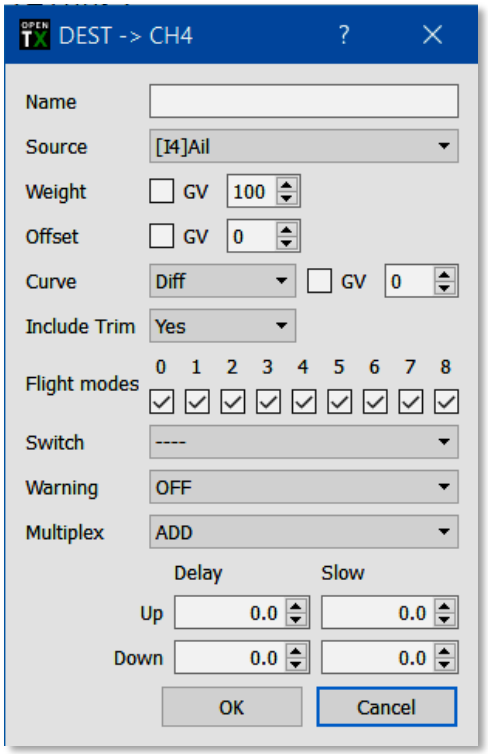




This is the edit window.



The Input Edit Window



- **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 sick 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 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 **Inputs** screen:

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

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 way of ensuring this does not happen, both are equally valid.

Method 1

Always assign every switch position every time:

```
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]
```

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:

Method 2

```
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]
```

Use a “catchall” line:

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.

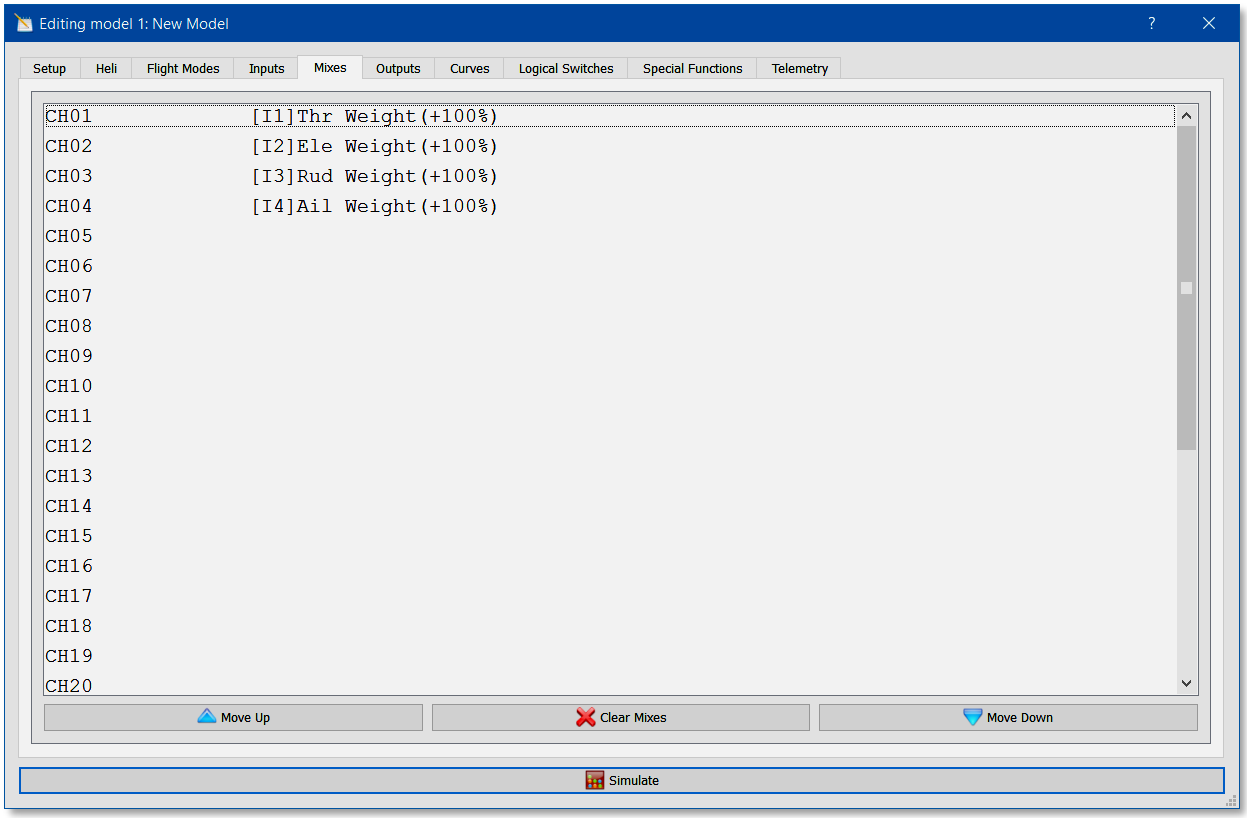
```
Ail Weight(+100%) Expo(30%) Switch(SG↑) [Hi rate]
Ail Weight(+80%) Expo(20%) Switch(SG-) [Med rate]
Ail Weight(+100%) [Catchall]
```

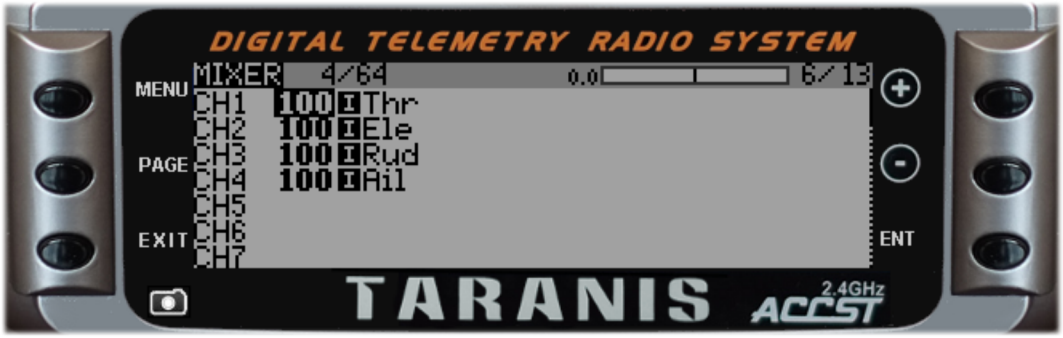
The example above illustrates how important it is to always check out the functionality of model settings before flying with a new configuration or after changing the configuration. The **Companion** simulator is an excellent method for doing this as each combination of switch and controls can quickly 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.

The Mixes Screen

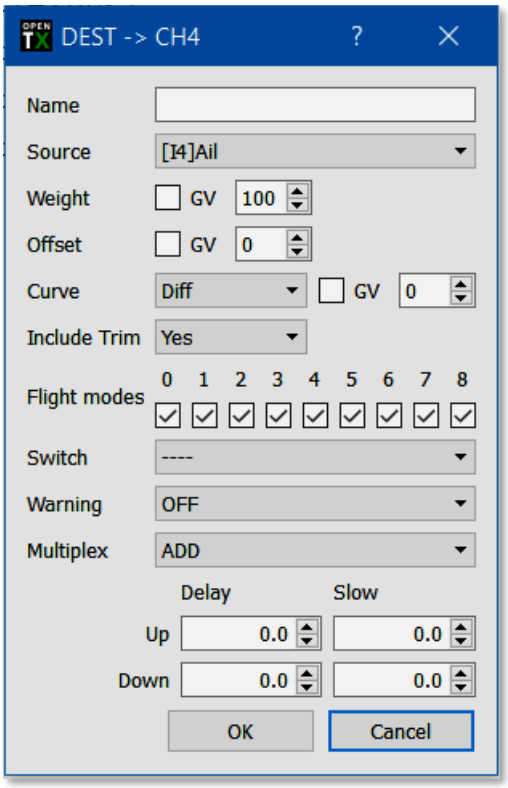
The **Mixes** screen is where the actions on the controls will be mixed and mapped to outputs and thence on to servos. **OpenTX** does not have any predefined mixing functions that relate only to a particular model type or situation, it rather gives you a blank canvas you can build upon. The key to configuring 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 **CH01** 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.





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 64custom (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 **CH02** 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 sick 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.

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 **CH02**, 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.

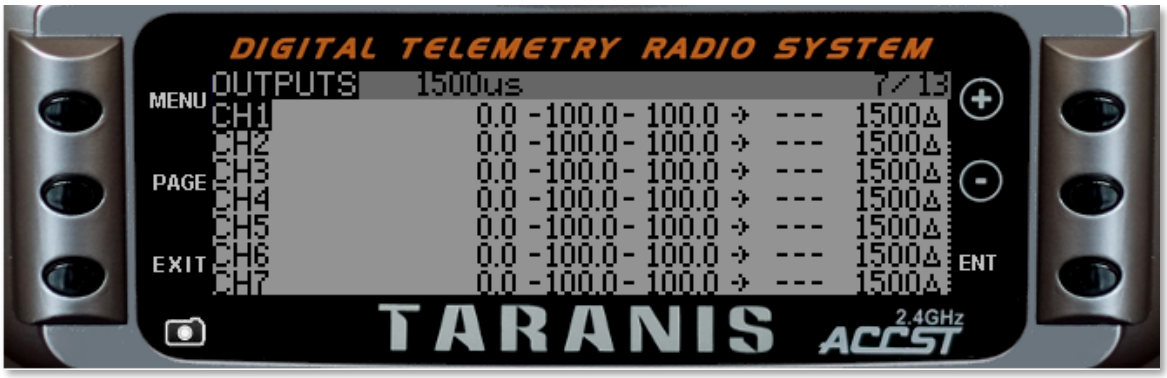
The Outputs Screen

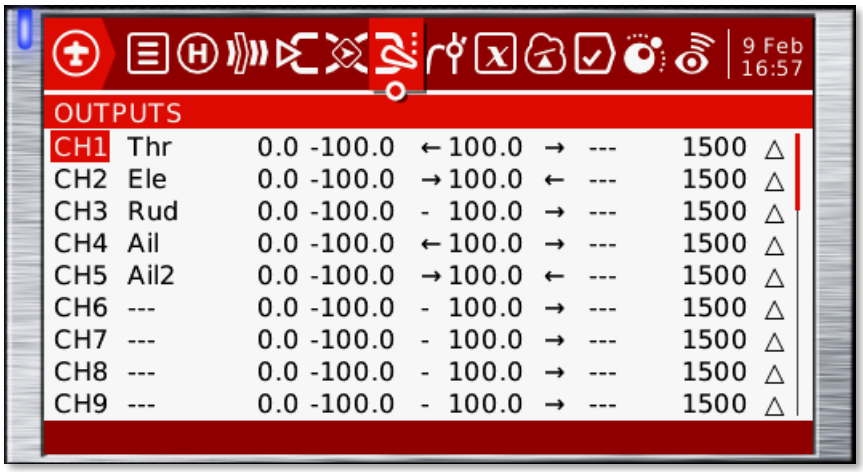
Editing model 1: New Model

Setup | Heli | Flight Modes | Inputs | Mixes | Outputs | Curves | Logical Switches | Special Functions | Telemetry

#	Name	Subtrim	Min	Max	Direction	Curve	PPM Center	Linear Subtrim
CH1		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH2		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH3		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH4		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH5		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH6		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH7		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH8		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH9		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH10		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH11		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH12		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH13		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH14		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH15		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH16		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH17		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH18		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH19		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH20		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	
CH21		GV 0.0%	GV -100.0%	GV 100.0%	---	----	1500us	

Simulate



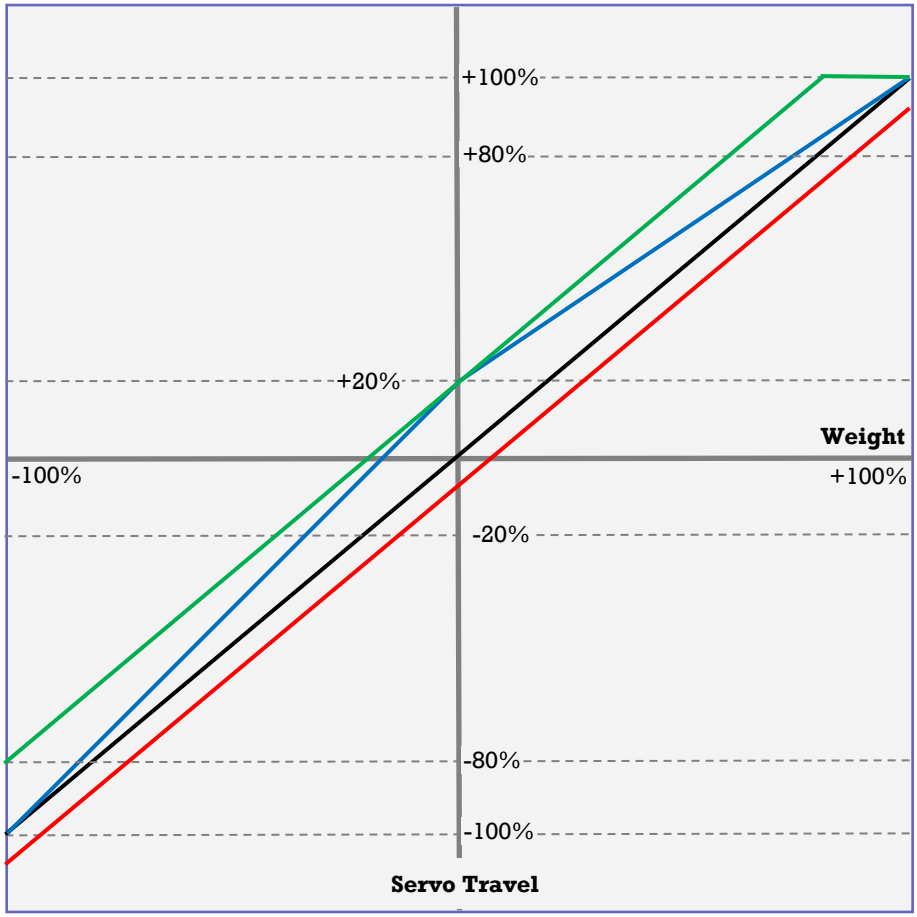


For each channel, we can define:

- **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.
- **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.

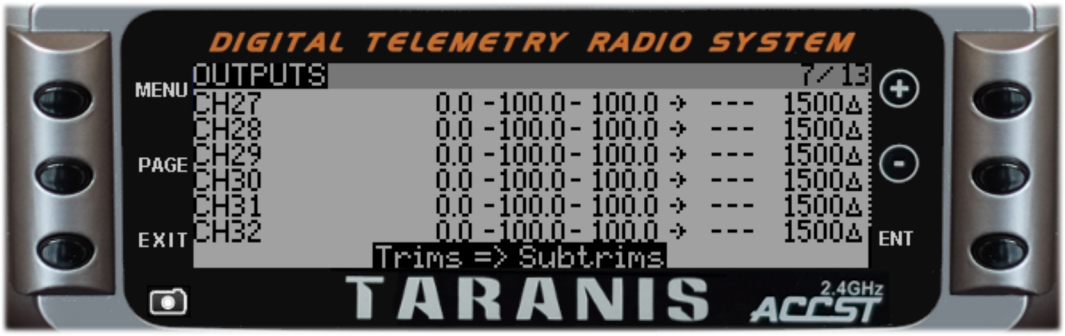
The following diagram illustrates the respective behaviour of both subtrim modes and how centre adjustment compares to them:



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

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

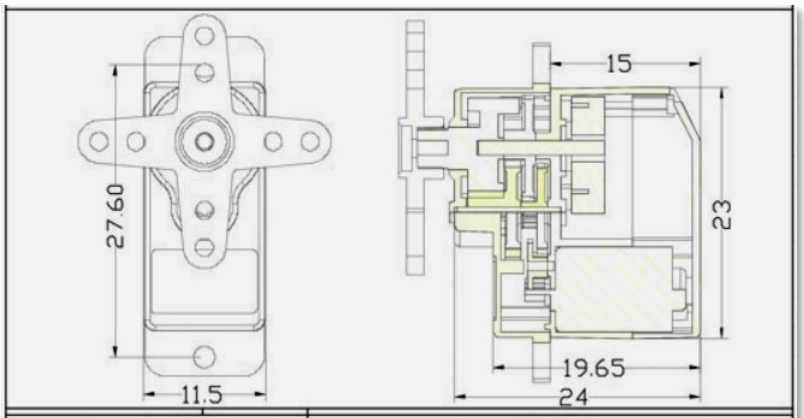
It soon becomes apparent that by using **OpenTX** one becomes far more familiar with the mechanics of how the transmitter, receiver and servo combination works in a model. It is, therefore, worth outlining some of the mechanics of servos here to ensure they are used to best effect.

- **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**µs as the centre. The default 100% travel for **OpenTX** on the Taranis is 1500 ±512µs, or **988**µs to one side and **2012**µs 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 ±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 ±30°, 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.

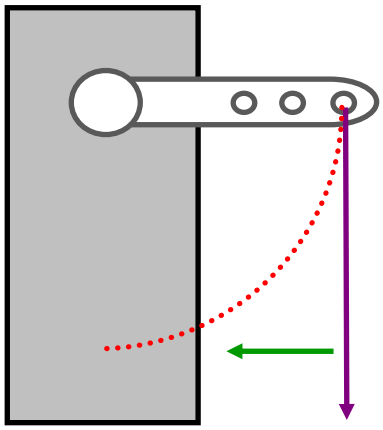
One like this will give the centre position precisely, as shown, or will rotate between fixed points, either 1000µs to 2000µs, or 500µs to 2500µs depending on how the device is set up using the menu system. Clearly the 1000µs to 2000µs setting is perfect for **OpenTX**. There is also a small knob on 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 500µs to 2500µs 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 500µs, others will continue rotating through a full 360° when set to 2500µs.

Wherever possible, one should look up the manufacturer's specifications. Here one can see the operating pulse width extends from 1100µs to 1900µs. 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 ±30°. This is fairly normal and there is a good reason for this.



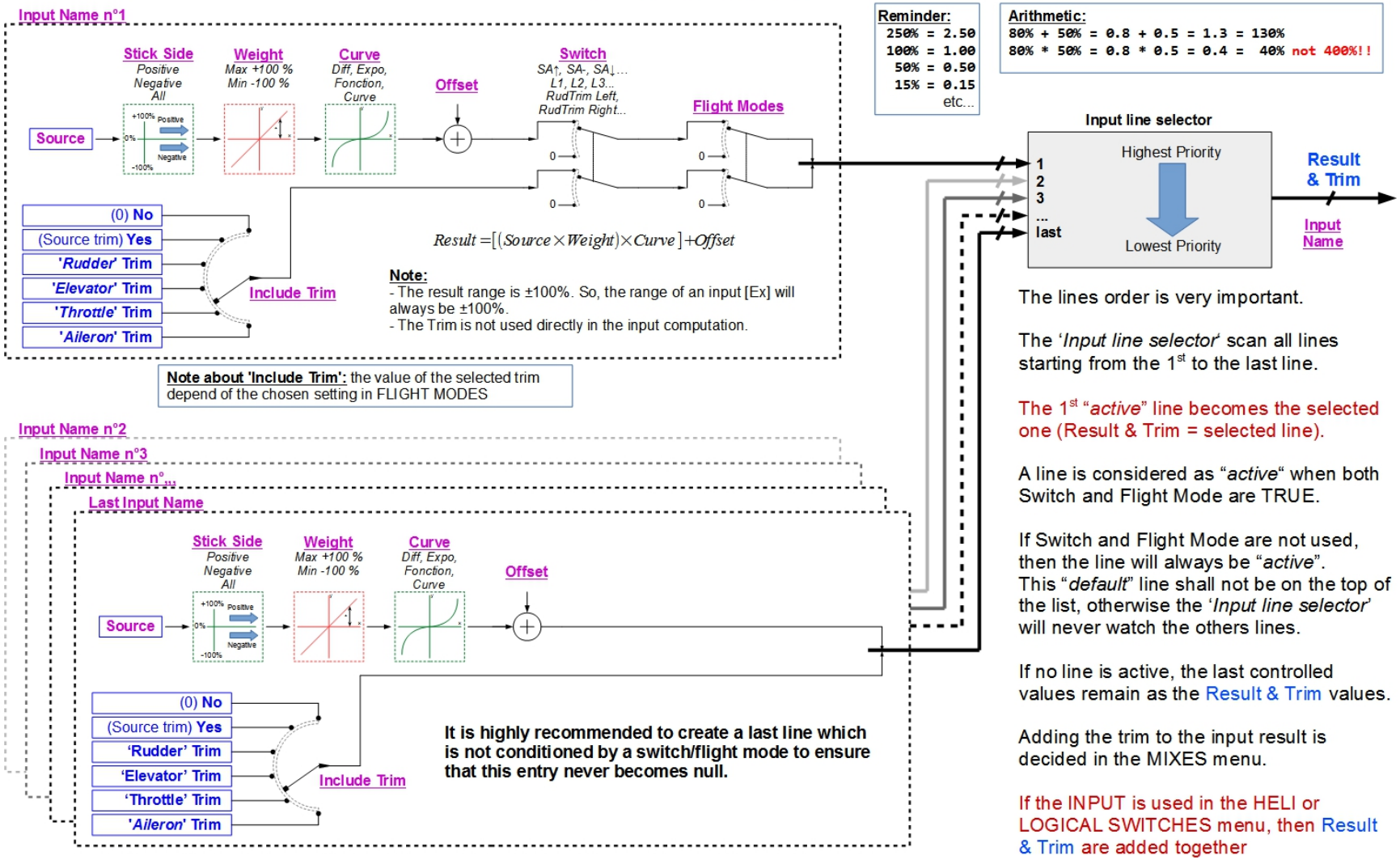
Operating Voltage:		4.8~6.0V
STD Direction:		Counter Clockwise / Pulse Traveling 1500 to 1900usec
Stall Torque:	4.8V	1.6 Kgf.cm(21.0 oz.in)
Operating Speed:	4.8V	0.12 Sec/60° at no load
Stall Torque:	6.0V	2.0 Kgf.cm(28.0 oz.in)
Operating Speed:	6.0V	0.10 Sec/60° at no load
Size:		23X11.5X24 mm
Weight:		12 g
Plug Available		FUT ;JR
Other		Analog ;Metal

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 ±30°, 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



The lines order is very important.

The '*Input line selector*' scan all lines starting from the 1st to the last line.

The 1st "*active*" line becomes the selected one (Result & Trim = selected line).

A line is considered as "*active*" when both Switch and Flight Mode are TRUE.

If Switch and Flight Mode are not used, then the line will always be "*active*". This "*default*" line shall not be on the top of the list, otherwise the '*Input line selector*' will never watch the others lines.

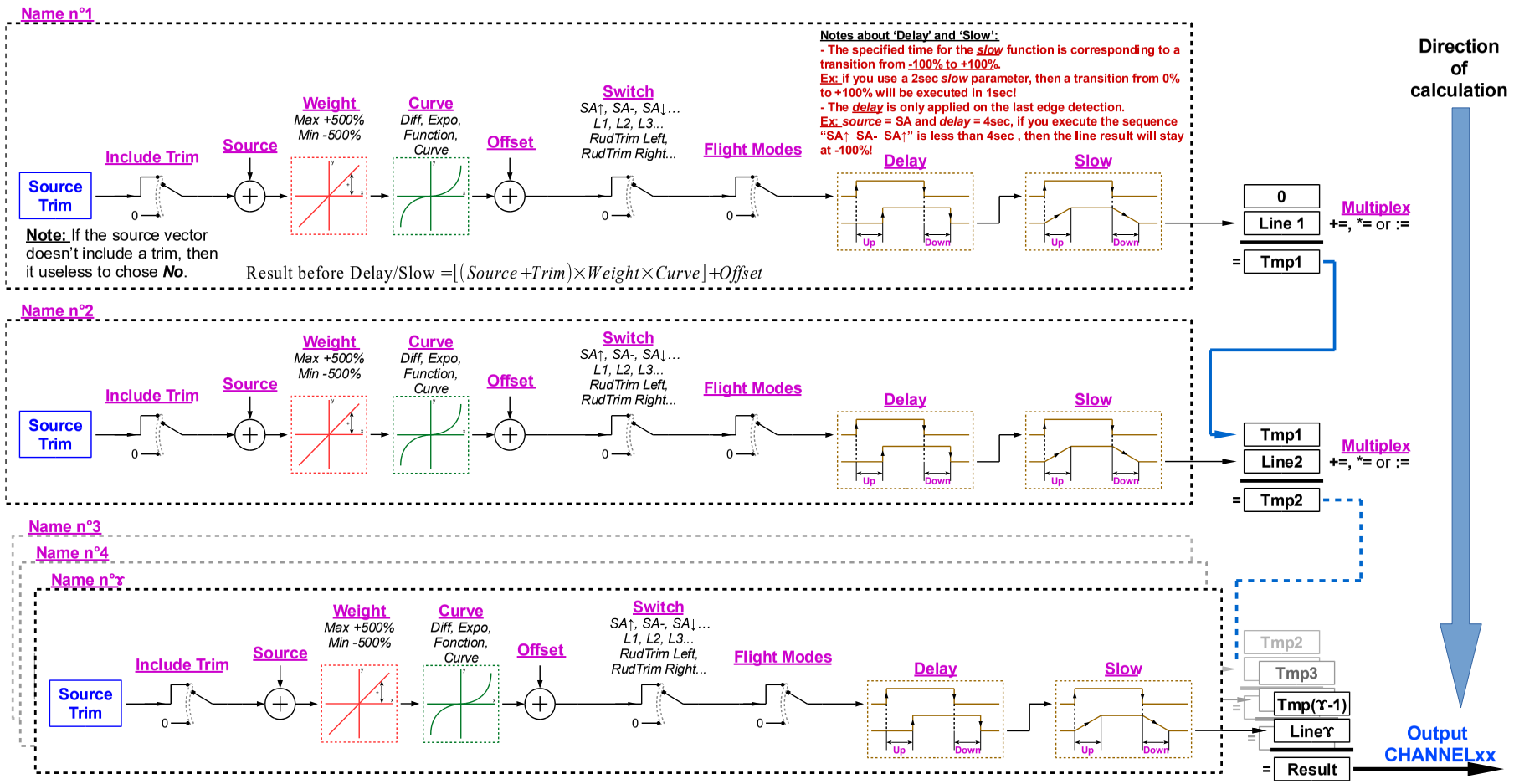
If no line is active, the last controlled values remain as the Result & Trim values.

Adding the trim to the input result is decided in the MIXES menu.

If the INPUT is used in the HELI or LOGICAL SWITCHES menu, then Result & Trim are added together

This diagram reproduced by kind permission of Sébastien Charpentier

MIXES Diagram



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

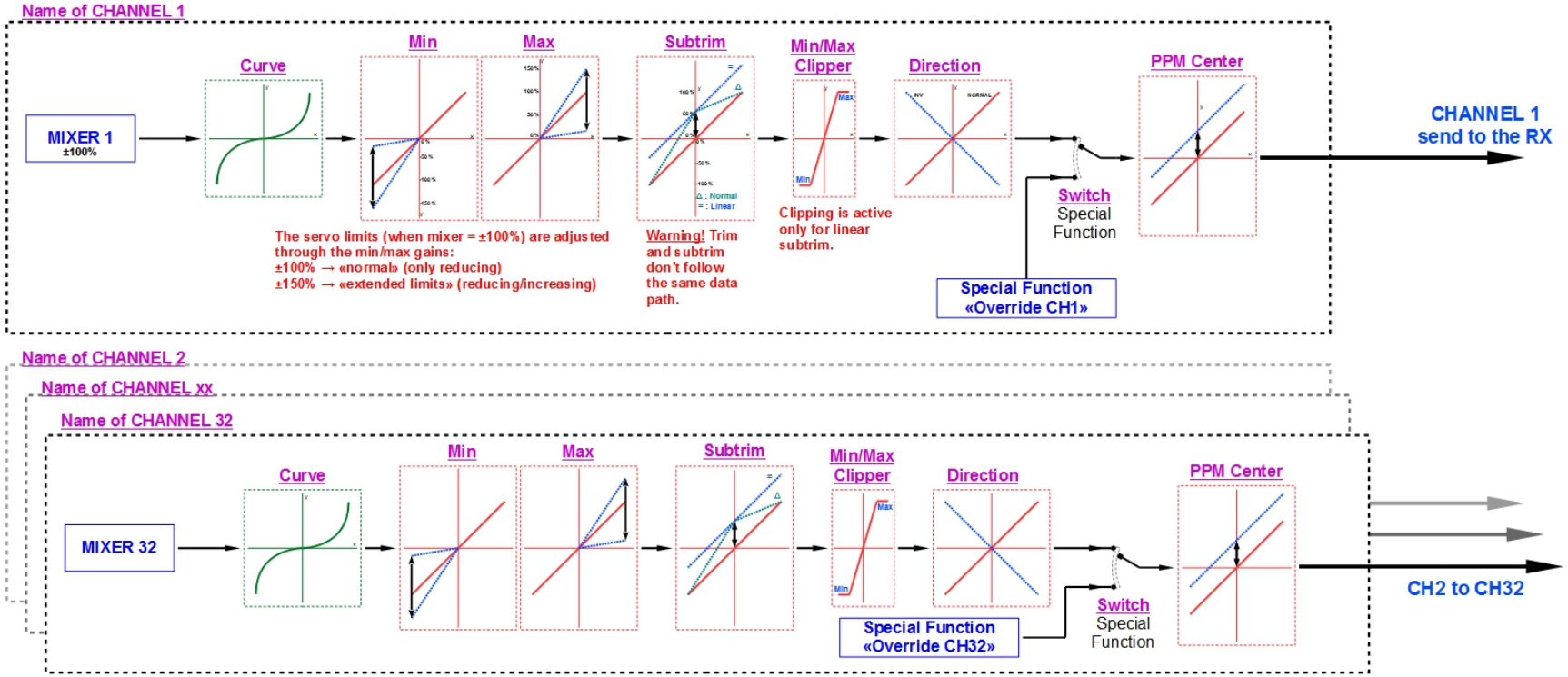
Reminder:
250% = 2.50
100% = 1.00
50% = 0.50
15% = 0.15
etc...

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% not 400%!!

OUTPUTS/SERVOS Diagram



Mode D8 = 8 Channels
Mode LR12 = 12 Channels
Mode D16 = 16 Channels

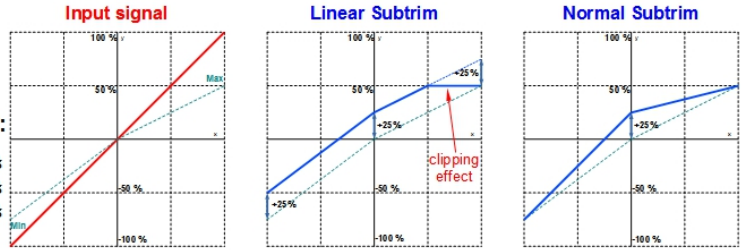
A 2nd TX module can be used in the external JR bay.
In this case, the value above are doubled.

Warning!!
The special function 'Override CHxx' doesn't take in account the min/max, direction, etc. Only the PPM Centre parameter is used. It is probably better to use the 'REPLACE' multiplex in MIXES menu. If you don't take care, servos can be driven out of their mechanical limits and in the wrong direction...

If you don't use (don't like) the 'override' special function, you may completely disable it by ticking the '*nooveridech*' option when updating your firmware from OpenTX Companion.

Example:
Waveform after going through the functions '*Min*', '*Max*', '*Subtrim*', and '*Min/Max Clipper*':

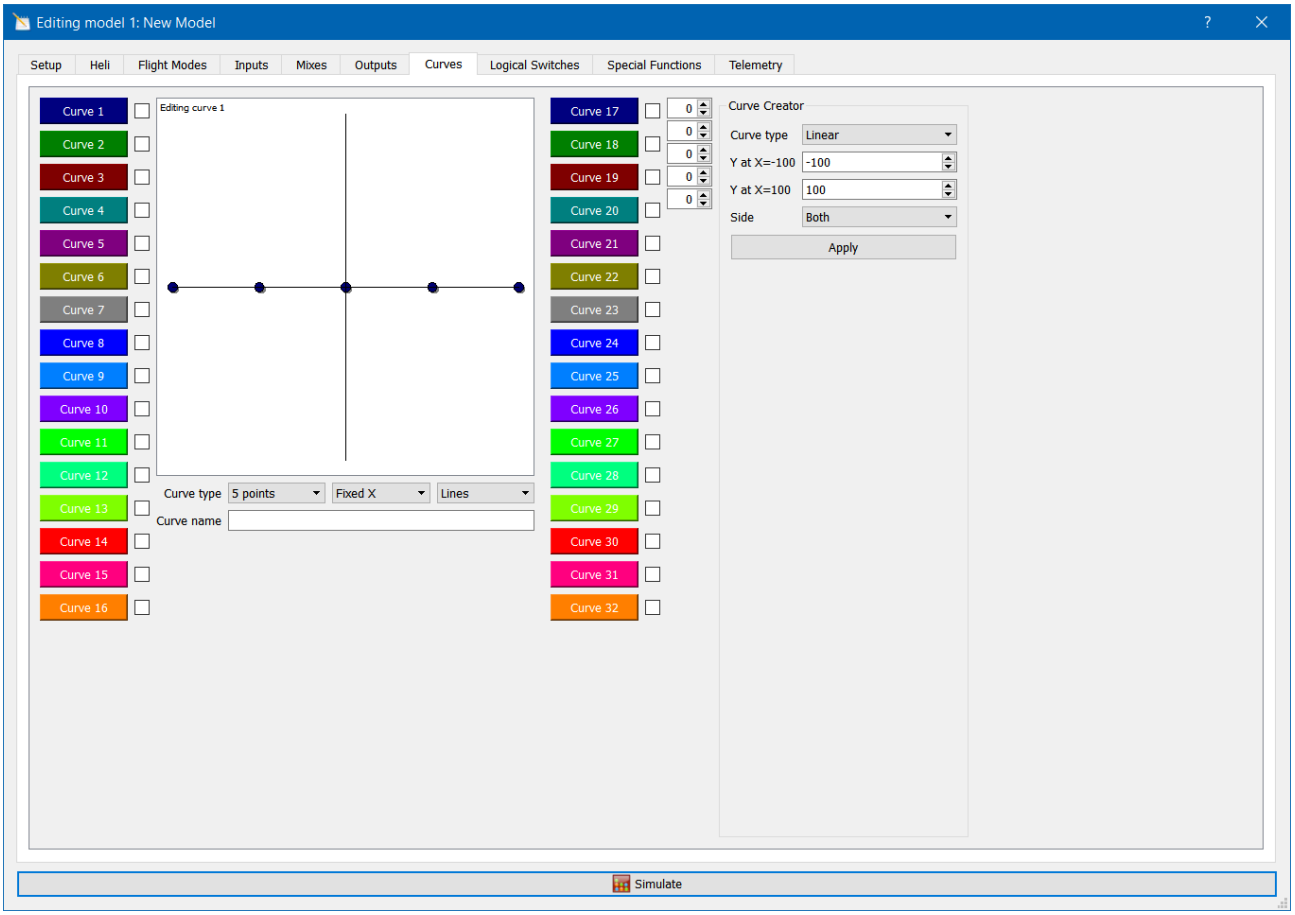
```
Subtrim = +25%
    Max = +50%
    Min = -75%
```

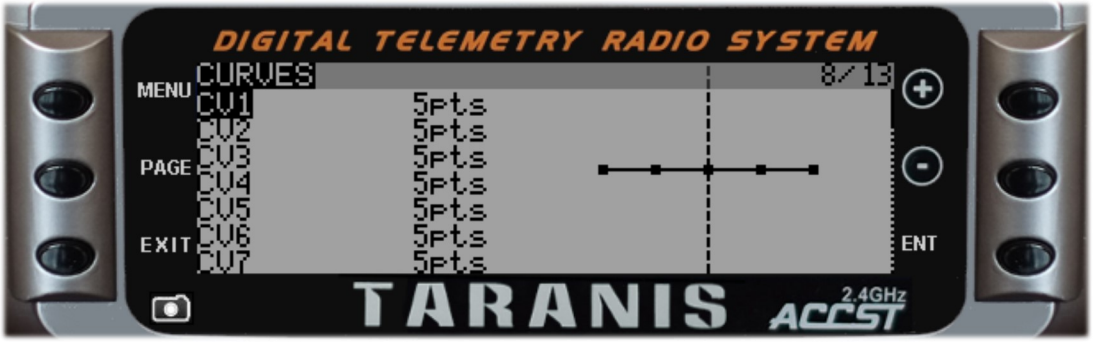


Remarks concerning the 'Linear Subtrim':
The linear subtrim add an '*offset*' from min to 0 and from 0 to max.
- (*Min + subtrim*) → *subtrim* WITH clipping at *Min*
- *subtrim* → (*Max + subtrim*) WITH clipping at *Max*

This diagram reproduced by kind permission of Sébastien Charpentier

The Curves Screen

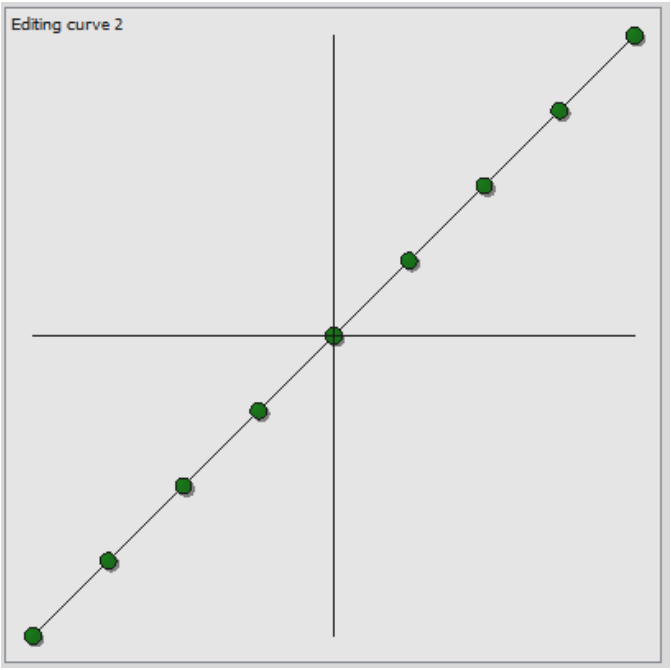




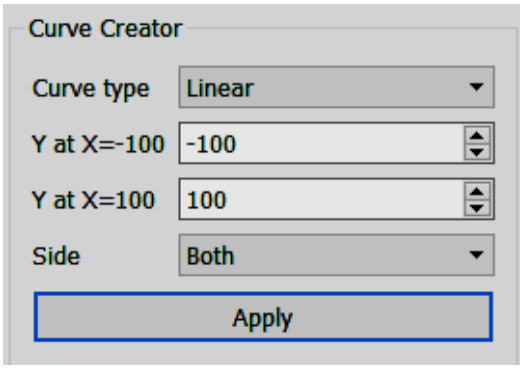
Custom curves can be used either in **Inputs**, **Mixes** or **Outputs**. There are 32 available, and they can have anything from 2 points to 17 points. They can have either fixed or user-definable X co-ordinates. Curves can be drawn on **OpenTX Companion** using either the mouse and moving points directly, by entering values for each point , or by using the **Curve Creator**.

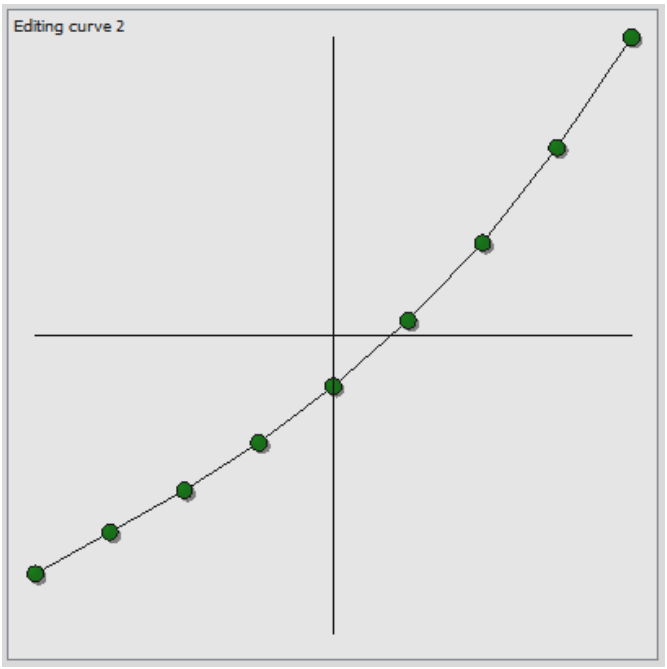
The "X" (horizontal) axis is the input movement, i.e. a joystick, and the "Y" (vertical) axis is the output. The left hand side of the graph is with the input fully negative i.e. with the joystick fully left or fully down.

The **Curve Creator** allows curves to be quickly created. To show how this works, it is easiest to give a few practical examples. Looking at a 9 point linear curve, we get this curve with the following settings in the curve creator.

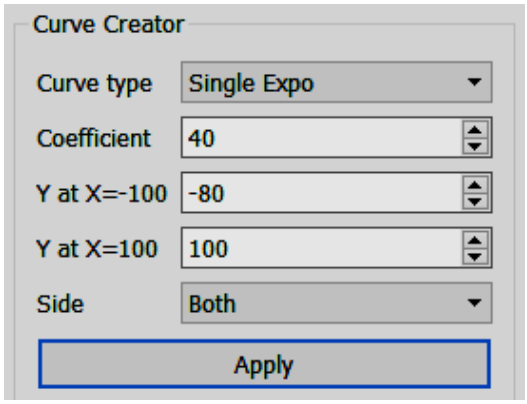


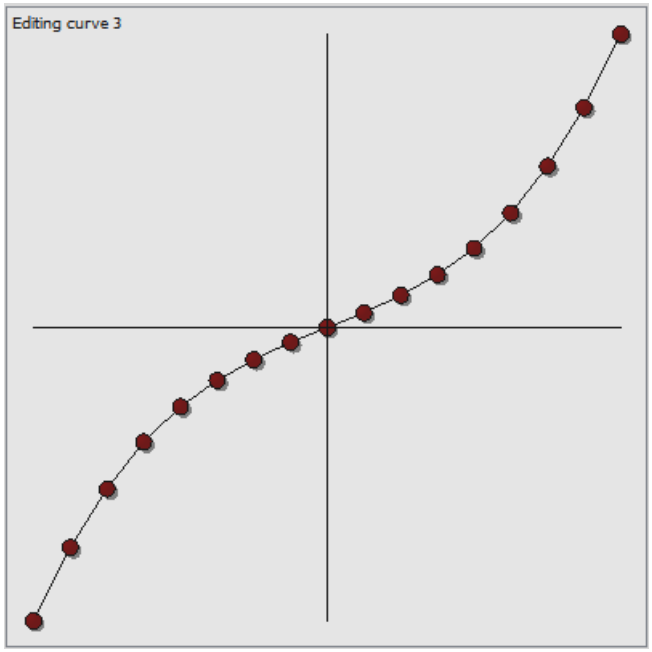
If **Y at X** is reduced to –80 then the curve will completely re-scale to give a curve that starts at –80 and ends at +100.



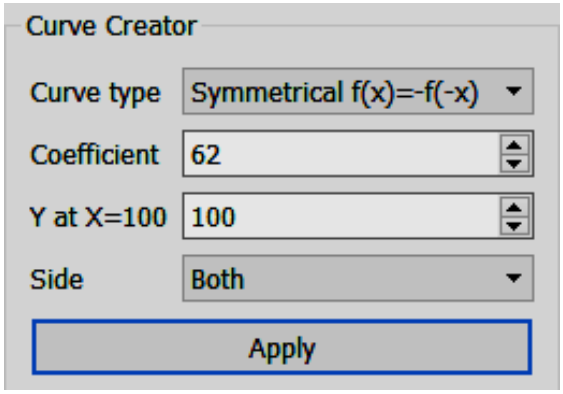


It is possible to also create exponential curves. Here we have an exponential curve which starts at –80 and ends at +100. Notice there is an extra box in the curve creator, **Coefficient**. This is the exponential value and has been set to 40 here.

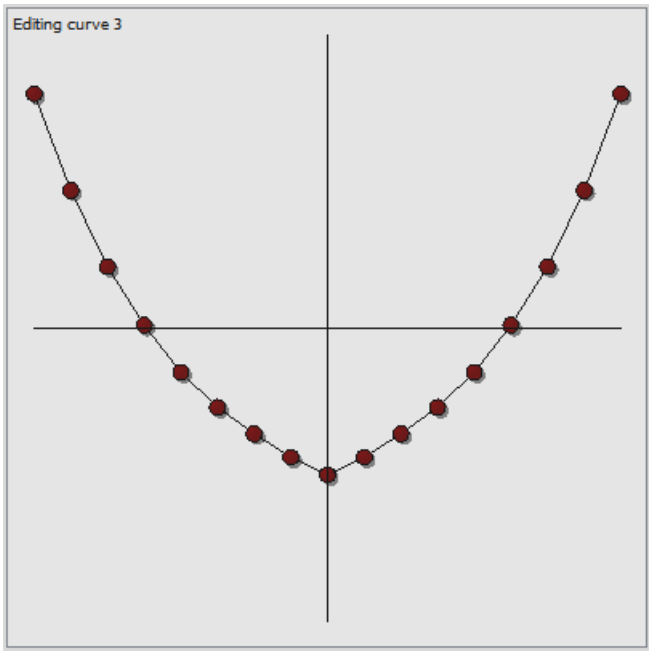


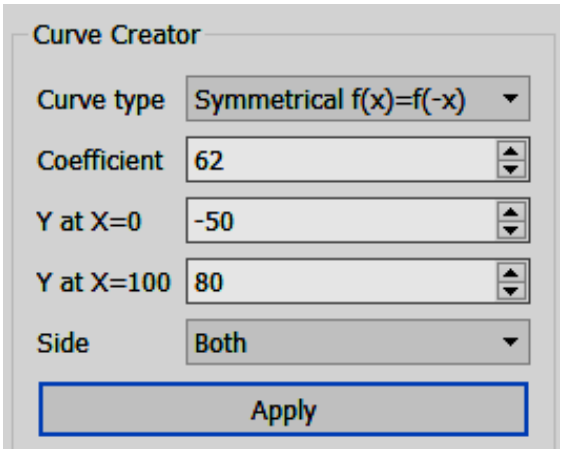


This is a 17 point curve using one of the functions, **Symmetrical f(x)=-f(-x)**.

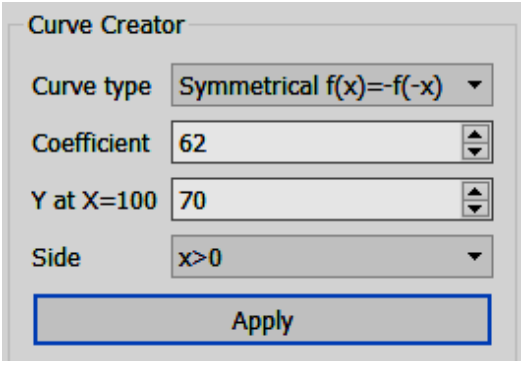


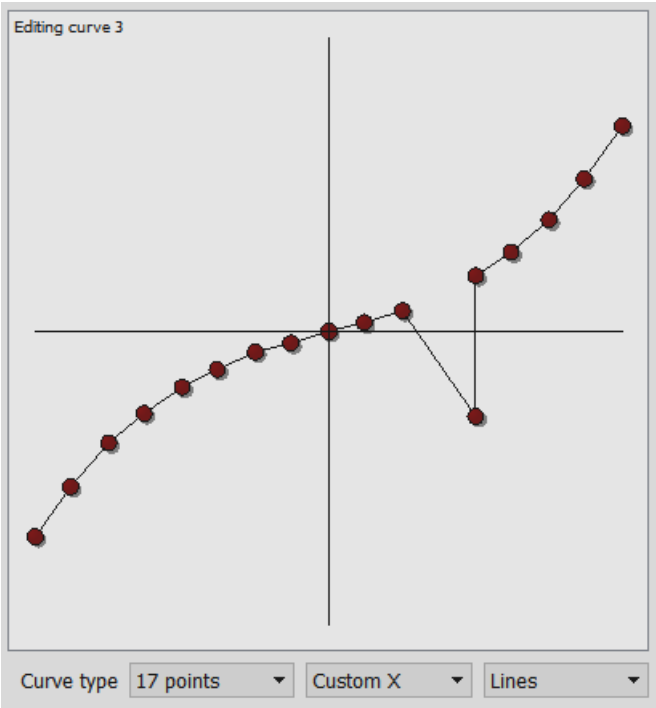
The function **Symmetrical f(x)=f(-x)** gives the following curve. Notice now there is an extra box to define **Y** when **X=0**.





Using the **Side** function, just one side of the curve can be changed. Here the upper limit is reduced to 70, but only for positive values of **X**. Thus there is an easy way of having asymmetrical curves for a variety of purposes.



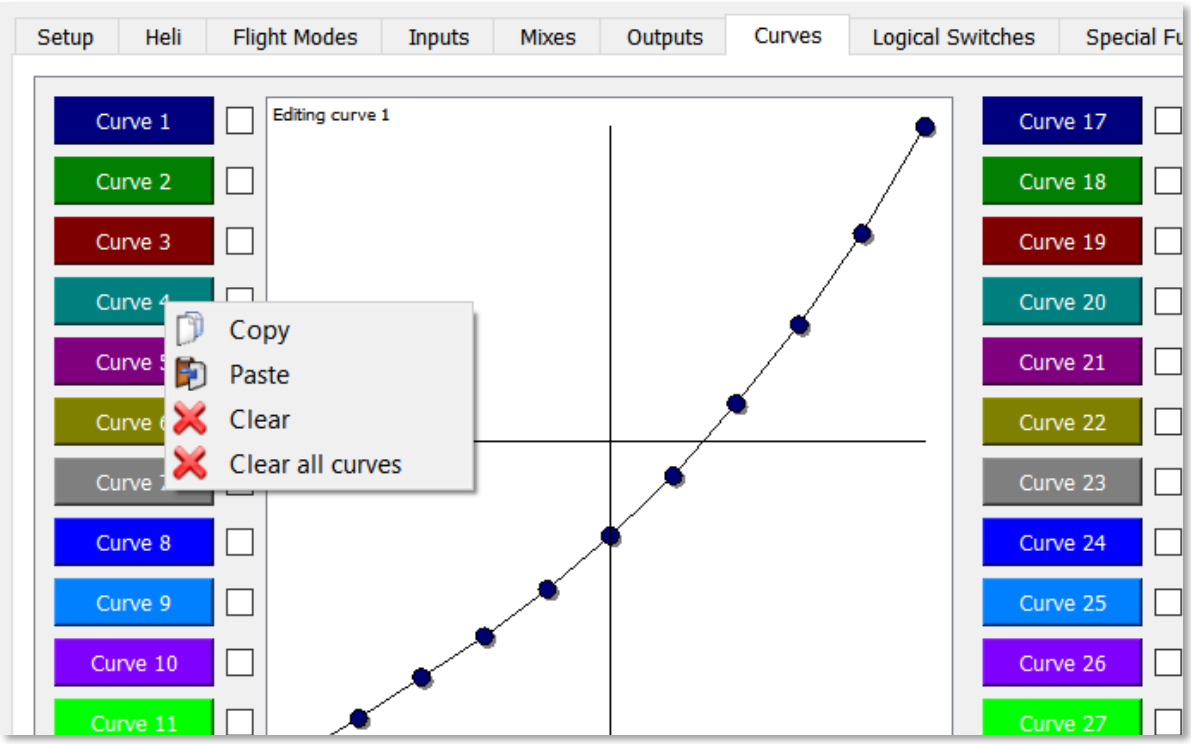


Finally, using the **Custom X** facility, individual points can be moved using the screen cursor to any point between the previous and the next **X** value and to any **Y** value.

Obviously, the scope and depth of the curve facility will entail much experimentation with an actual model in flight to perfect the operation. A simple technique to make this easier is to produce three similar curves, changing parameters slightly and assign them to a switch to be able to change curves in flight. This is easily done on the **Inputs** screen or the **Mixes** screen as shown below where switch A has been assigned to the three curves:

```
[I1]Thr        Thr Weight(+100%) Curve(1) Switch(SA↑)
               Thr Weight(+100%) Curve(2) Switch(SA-)
               Thr Weight(+100%) Curve(3) Switch(SA↓)
```

Another recently added feature that can simplify generating curves on the **OpenTX Companion** is a copy and paste facility. By right-clicking on the mouse, with the pointer on any of the 32 curve numbering boxes, a sub menu comes up to allow copying, pasting and deleting of curves.



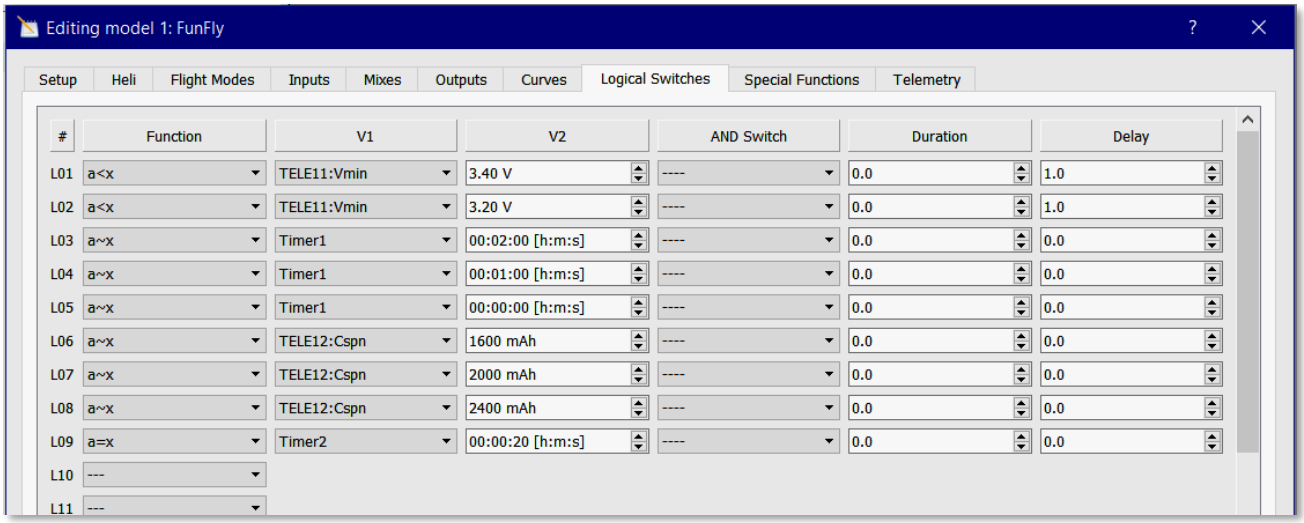
The Logical Switches Screen

This screen, together with the **Special Functions** and the **Telemetry** form one of the most exciting parts of **OpenTX** which gives the system huge flexibility. Essentially it allows the user to create their own "switches" or conditions whereupon some other action will follow. Rather than being turned on or off by the action of physically adjusting a switch, they are turned on and off by evaluating the conditions set on the **Logical Switches** screen. These conditions may use a variety of inputs such as physical switches, other logical switches, sources such as telemetry values, channel values, timer values, or **Global Variables**. They can even use values returned by a LUA model script.

In the screenshot below, 9 logical switches have been set up, and between them they do three tasks. **L01** and **L02** look at a telemetry value, in this case the voltage of the lowest cell of a Lipo battery, and **L01** becomes true when the voltage drops below 3.4 volts. On the **Special Functions** screen, this is used to give a verbal "early warning" of low flight battery voltage. Similarly **L02** can give a critical battery warning. **L03** to **L05** look at the time set by **Timer 1**, used later to give minute countdowns, and **L06** to **L08** look at another telemetry reading, later used to give a verbal indication of the current consumption of a lipo battery. **L09** becomes true when Timer 2 has counted up 20 seconds.

AND Switch, Duration and Delay

The **AND** switch, **Duration** and **Delay** for the switches work in the same for all switch functions.



Logical Switch Functions

Logical Switches are set by choosing the function, then refining the options (or parameters): **V1**, **V2**, **AND** switch, **Duration**, and a **Delay** for each switch. **Logical Switch** functions are given as expressions such as **a = x** where **a** represents the source to be used, given as **V1**, and **V2** (value 1 and value 2) gives the value for **x**. This can be a little confusing, but other conditions can be set also which do not use either **a** or **x**.

a = x — This is used to check if the value of a selectable source is equal to x, a chosen value.

a ~ x — This is used to check if the value is approximately equal. One needs great care when using the equals function because computers like to be exact! As an example, looking back at the previous screenshot, if one tests to see if the consumption is exactly 1600mAh, the actual telemetry reading may jump from 1598mAh to 1603mAh, so the condition is never met and therefore no message will be triggered. Most times it is better to use the approximately equals function rather than the "exactly" equals function. Do not use this function when the value will be precise.

a < x — The source, V1, is less than the chosen value V2.

a > x — The source, V1, is greater than the chosen value V2.

| a | > x — This is used to check if the absolute value (meaning irrespective of + or -) of a source V1 to see if it is greater than a chosen value of x, V2. Because it's an absolute function, whether the returned value is a positive or negative value doesn't matter.

| a | < x — This function operates similar to **| a | > x** except that it is true when **|a|** is less than x

AND — This switch checks that **both** the switches selected in V1 and V2 are true (i.e. ON). If both switches are true then the logical switch is true (ON).

In the above example, **L11** will be true if both switches **SA** and **SB** are set to

OR — This switch checks if **either** of the switches selected in V1 and V2 are true.

Now **L11** will be true if either (or both) of switches **SA** or **SB** are set to their mid positions.

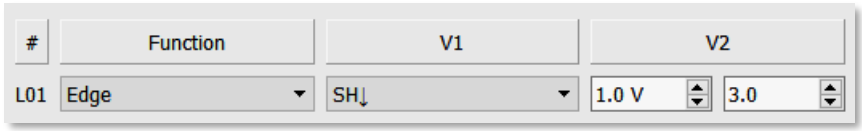
XOR

This switch checks if either, but not both of the switches selected in V1 and V2 are true.

Edge

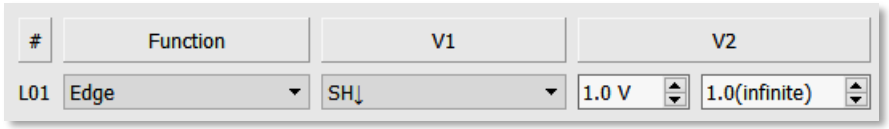
This is a momentary switch which can be activated by another switch (including logical and flight modes). The switch will stay on the length of time identified in duration. If testing with the simulator the switch will briefly flash green at the appropriate time if duration is 0, otherwise it will stay on for the duration time.

- V1 is an activating switch (including logical and flight modes).
- V2: is in two parts. The first, the delay before the switch activates, the second the "time window" for the switch to remain active. E.g.

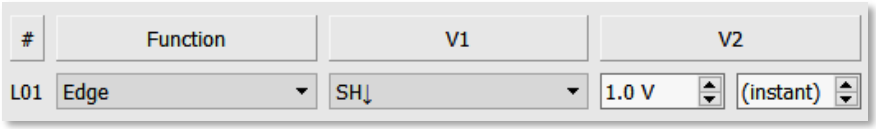


Here **L01** will trigger if **SH** is held on for more than one second but less than three seconds.

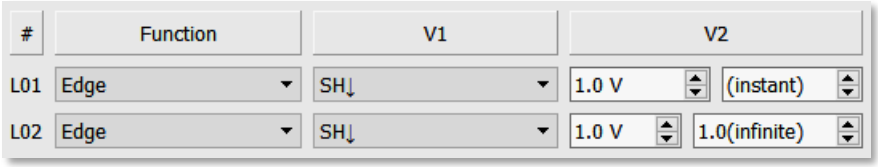
- There are two general settings for **Edge**:
  - **0.0 (infinite)**, the default. The **Edge** switch becomes active when the triggering switch is **released**.



  - **(instant)**, The **Edge** switch becomes active once the triggering switch has been true for the minimum duration selected and then released. This setting is activated by clicking the down arrow next to the default value of **infinite**.



Using the simulator, one can compare **instant** to **infinite**.



L02 will flash (i.e. signify true) when **SH** has been held down for one second, **L01** will then flash after **SH** is released. Neither will flash if **SH** is held down for a shorter period of time.

a = b This is used to check if the value of a selectable source is equal to b which is a different selectable source. It differs from **a = x** in that it compares two source values directly without specifying the numerical values for either.

a > b This is used to check if the value of a selectable source is greater than b, another selectable source.

a < b This is used to check if the value of a selectable source is less than b, another selectable source.

d > = x This function compares a change in value to a set value. This is used to check if the delta (the change in value) of a selectable source as chosen in V1, is greater than or equal to x as set in V2.

| d | > = x This is used to check if the absolute value of delta (i.e. delta always positive) of a selectable source is greater than or equal to x as set in V2. It operates the same as **d > x** without the need to specify a positive or negative signed value.

Timer This function is used to turn a logical switch ON or OFF at specified intervals. This is a repeating on/off timer with both variable on and off times.

- ✪ V1: the on time
- ✪ V2: the off time

Again this is very easily seen using the simulator.

Sticky This is a toggle. It can be thought of as another form of an on/off switch. It is turned on by the switch selected for V1 and turned off by the switch selected in V2.

The above function can be demonstrated using the simulator. Pulling **SH** on will first make **L18** true . Pulling **SH** again will make **L18** false, in effect turning it off.

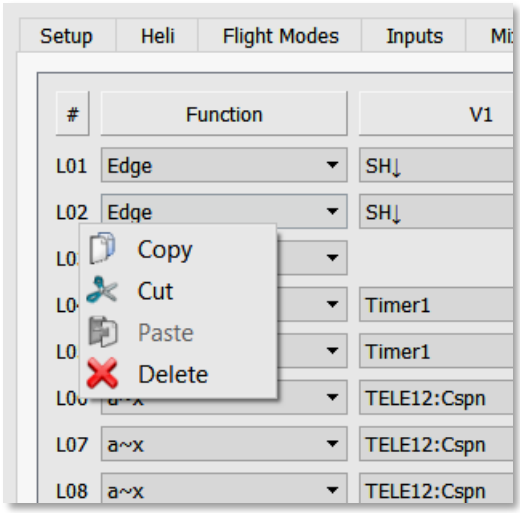
When using **Sticky**, take care as the flip toggle is turned on and off regardless of the **AND** switch on the same line. The logical switch will remain off if the **AND** Switch is false, but the toggle part will continue to turn on and off by V1 & V2, and is not reset by the **AND** switch.

AND Switch Any physical switch, logical switch or flight mode can be selected from those available under the **AND** switch options. Only if this is condition is true and the rest of the switch conditions are true will the switch be on. The switch functions V1 & V2 are evaluated FIRST, then the **AND** switch applied afterwards. **This is important to remember, particularly with the Sticky.**

Duration The length of time the switch will stay ON. If set to 0.0, the switch will remain on until the conditions make the switch off. Any other setting will cause the switch to go off after the number of seconds selected, even if the conditions remain true.

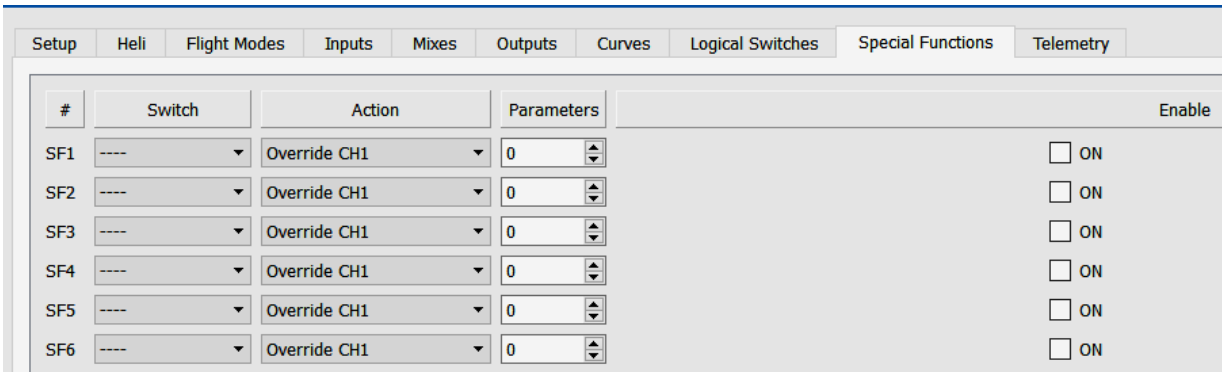
Delay This is the delay before the switch comes on once the conditions are true.

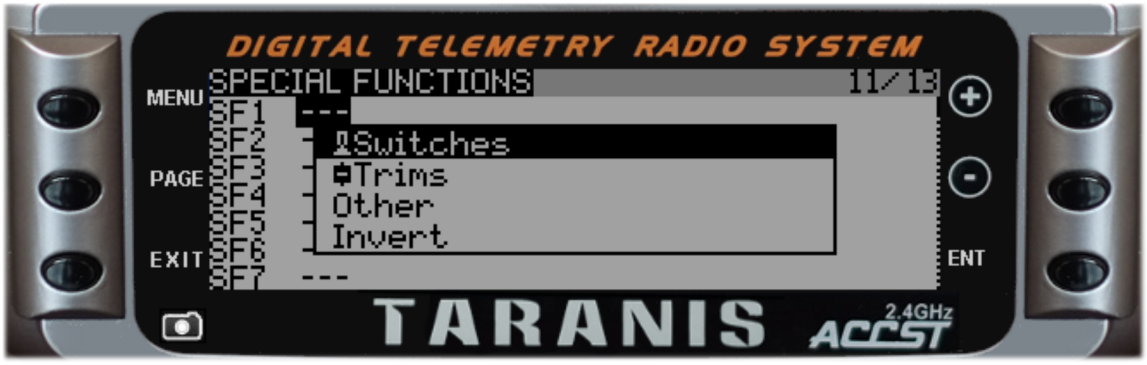
Finally, there is this neat little option. Right clicking on the **Logical Switch** number in the left hand column brings up a small menu to enable **Logical Switches** to be copied, cut or moved. Do note that moving the logical switch to another place on the screen will not alter the logical switch number in other screens of the program. This feature can also be used to copy a logical switch from one model to another. This is done by opening up an **Editing** screen for each model and displaying both windows on the screen at the same time.



The Special Functions Screen

The **Special Functions** screen gives the option of adding all sorts of exciting features to a model program. For every model, there are up to 64 Special Functions available. In addition there are a further 64 **Global Functions** available. These are accessed from the **Setup** screen and are the same as the **Special Functions** expect that they operate on all model settings. Thus for a function one would like for every model, such as being able to adjust the volume level using one of the potentiometers, one can set it in **Global Functions** rather than having to set it each time for each model.





Basically for each **Special Function** there are four parameters, **Switch**, **Action**, **Parameters**, and **Enable**.

Switch

The switch input can be any one of the following

- The physical switches in any position.
- The trim switches when the trims are in the up, down, left and right positions.
- The 64 **Logical Switches**
- **On** This setting means the function is always enabled. e.g. for a volume control.
- **One** This setting triggers the function just once.
- Any of the 9 flight modes.

Action

Override CH01 to CH32	Weight -100 to +100. Forces a channel output to a specific value.
Trainer: Rudder **Elevator** **Throttle** **Aileron**	Enables trainer mode individually for each control.
Trainer	All channels enabled.
Inst. Trim	Adds the current stick position to the respective trims. When activating the selected switch the current stick positions will be added to their respective trims. This is typically assigned to a momentary switch, and used on a maiden flight if you expect trims to be way off. Instead of frantically clicking the trim tabs, you would hold the sticks so that the model flies straight, and depress the switch once. It is best to remove that entry after the maiden flight, to avoid hitting it by mistake and bringing the model badly out of trim again. This can be done by unticking the **ON** box.
Play Sound	This will play any of the simple sounds listed in the drop down parameters box.
Play Haptic	Only works if haptic mode is enabled in the **General Settings**, and the transmitter supports haptic
Reset	Resets Timer 1, 2, 3, flight or telemetry. ✪ The selected timer value is reset to the value set by the timer parameters in the Model Setup screen. ✪ Telemetry resets telemetry values ✪ Flight resets both telemetry and timers ✪ Individual telemetry data can be reset
Vario	Vario will only sound when the designated switch is enabled.
Play Track	Plays any of the sound files stored on the SD card. A list of available sounds will appear in the parameters column in a drop down box. However, the **Companion** must have the correct pathway set (**Companion Settings Menu**) to a computer copy of the SD card for this to work on the **Companion**.
Play Value	Speaks the value of any of a range of controls available in the parameters menu. These can include switch or joystick values, or telemetry values, or time.

Play Script	This will play a LUA script
SD Logs	This sets the frequency in seconds with which data logging is sampled and stored. (see data logging)
Play Script	This will play a LUA script
SD Logs	This sets the frequency in seconds with which data logging is sampled and stored. (see data logging)
Volume	Allows a control to be assigned to alter the volume.
Backlight	Note that this option will override any settings made on the **Radio Settings** menu if set to **ON**. Can give more flexibility to the backlight function if the backlight mode is set to **OFF** in the **Radio Settings** menu
Background Music	Plays selected music.
Background Music Pause	Pauses selected music.
Adjust Global Variable	This allows each of the **Global Variables** to be adjusted
Set Failsafe Internal Module	This allows the failsafe to be set from a switch or other control. Thus, say if one wanted to set the failsafe on a powered glider, it could be placed into a gentle spiralling descent, and the failsafe set for all channels to reproduce this flight.
Set Failsafe External Module	This similarly sets the failsafe, but for the external module.

Enable

There are several options under this heading:

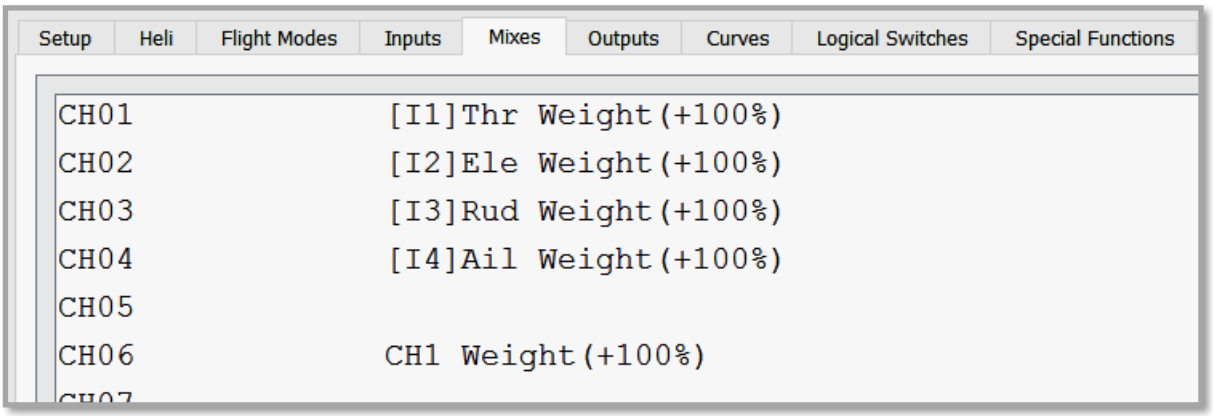
Played once, not during startup	This stops **OpenTX** playing this message when switched on. Some telemetry messages may still be played however, because the program starts before any telemetry data is received. This can be stopped by having a one or two second delay in a **Logical Switch**.
No Repeat	Does not repeat the message. Otherwise setting the option to a time will make **OpenTX** repeat the message with a frequency set by the time stated.
ON	This box is ticked to enable the function at startup. Unticking this box is a simple way of disabling, say, the **Instant Trim** feature rather than having to delete the line of code.

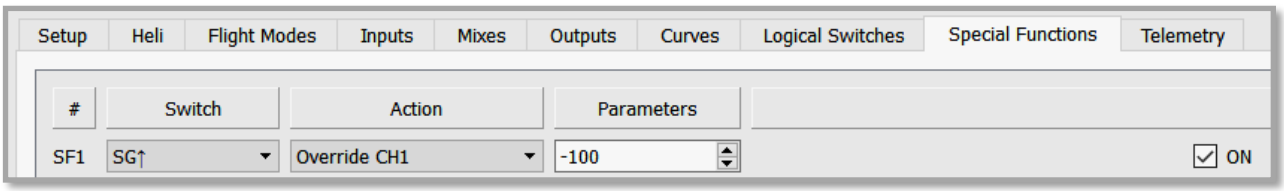
Note on Override:

Override operates on a channel **Output** and is not part of the channel **Mixes** calculation. The consequence of this is that the **Mixes** are not aware of the override or its value. The channel mix value is not affected. The fact that an override is active due to a particular mix condition cannot be counted on as part of the mix calculation.

The purpose of **Override Special Function** is to force the channel **Output** to a particular value while the **Special Function** switch is activated, overriding all **Mixes** belonging to the channel. The **Output** value will be held even if the active mix is changed, unless the new active mix deactivates the **Special Function** switch.

This can be demonstrated thus:

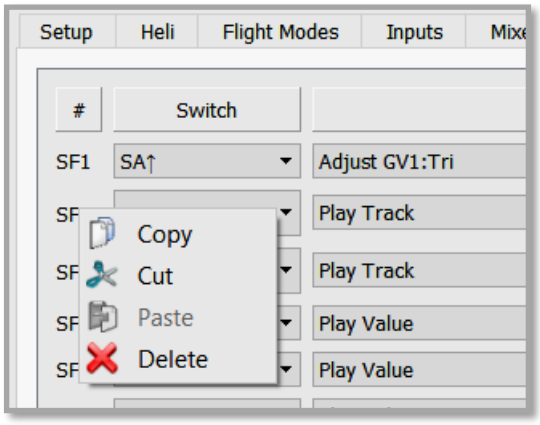




Using the simulator, one can see that **SG↑** channel 1 output is held at -100, whereas channel 6 will still show the channel 1 true value.

Checks also need to be made when using **Override** that servos are not driven past their normal operating range under any circumstances. **Override** takes place in the chain of commands after the **Outputs** screen, so any maximum and minimum limits set there are ignored.

A useful short cut when editing the **Special Functions** screen is right-clicking on the special function number in the left hand column to obtain the following menu box. This enables special function lines to be cut copied or moved not only within the same model functions screen, but between models too. This same feature is available with the **Logical Switches** as shown earlier.



OpenTX Companion will allow several models to be edited at once, with several editing screens opened. By resizing the screens one can simply copy from one screen to the other.