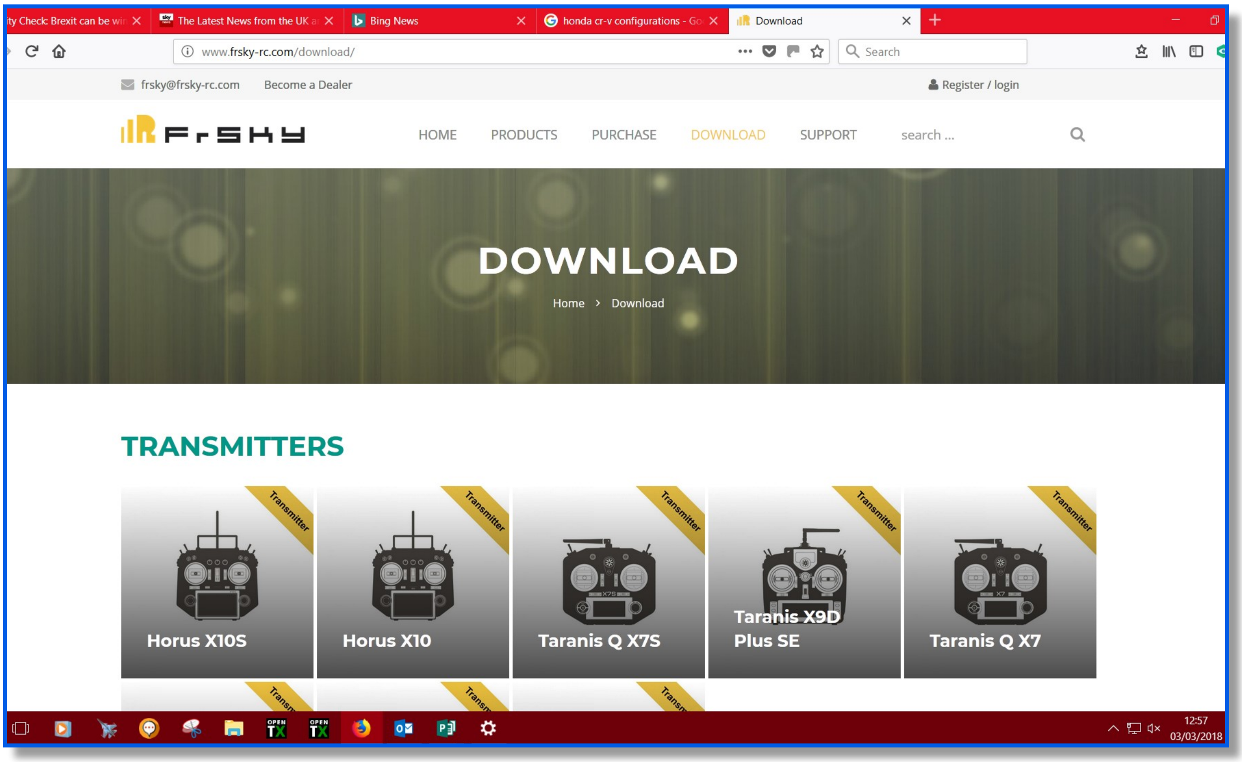
How To … Part 1

Contents

Warning: This method of flashing can only be undertaken if your transmitter uses the standard battery. If you have replaced the standard NiMH battery with a LiPo battery, this can damage the receiver.

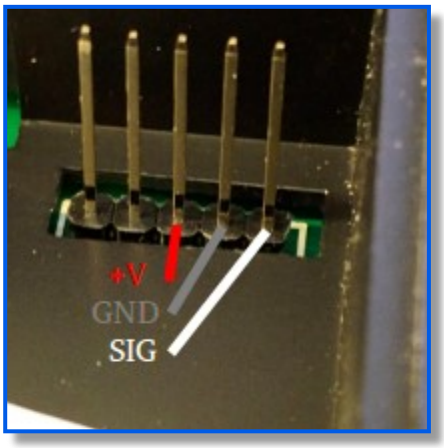
1. On your computer find the copy of the SD Card (if not copy the SD card from the Taranis), and check a sub-folder called FIRMWARES exists.
2. Go to the FrSky website http://www.frsky-rc.com/download/ and download the latest firmware. This website screen is frequently changed, so do not assume it will look exactly like this:



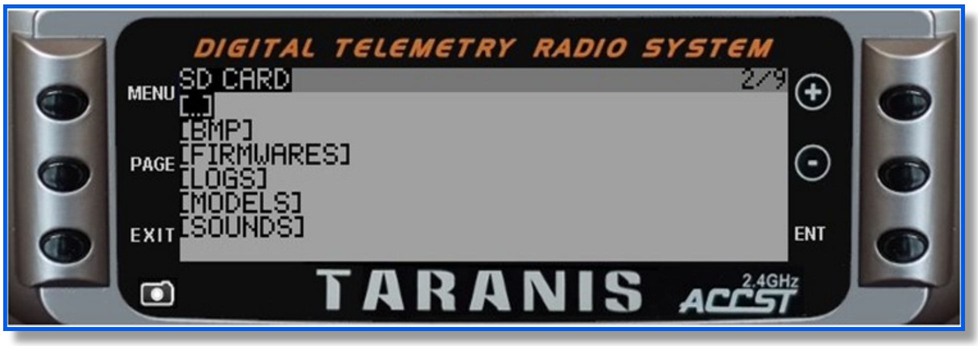
3. Above shows the XJT firmware (for the transmitter), the X4R, and the X6R/X8R. These are the latest LBT (listen before talk) EU upgrades to meet EU regulations.
4. Go to your normal download folder, and with Windows 10, a right click on the appropriate file will give the options to extract these files to a particular folder. Otherwise an unzip program will be needed to unpack the files.
5. Select the FIRMWARES folder on your computer version of the SD card, and copy the XJT, X4R and X6R/X8R to this folder.
6. These files will need to be copied to a similar folder on the actual SD card in the transmitter. In OpenTX, with the transmitter in boot mode and connected to the computer, this can be

achieved using the SD Card Synchronisation feature.

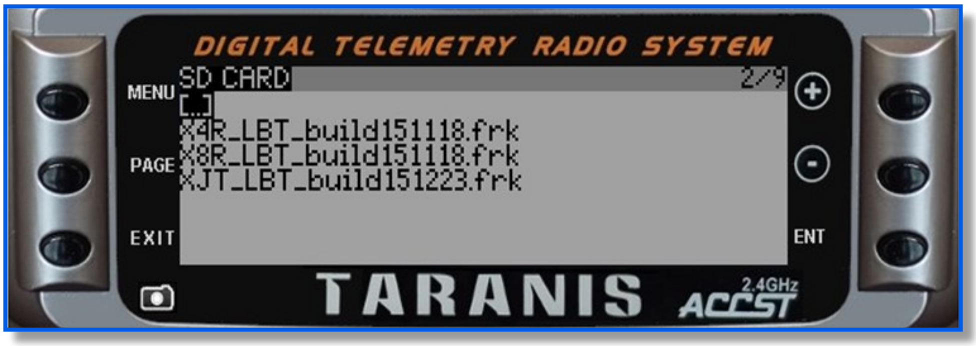
7. Next you will need to make a lead up to transfer the files. The easiest way is to use an extension lead with plugs at one end and sockets at the other and swap the red and brown pins at one end if using Spektrum colours, or the red and black for the Taranis telemetry colours.
8. Connect the reversed end into the back of the Taranis with the cover removed. Note that the signal lead is towards the bottom of the transmitter, the middle lead is the black lead. The plug at the back has five pins, only the bottom 3 are used.
9. Connect the other end of the lead to a sensor lead already plugged into the Smart Port on the receiver. Ensure that when flashing a receiver, there is no power source connected to the receiver, except from this lead. If upgrading an X4R receiver, you will need to use the special sensor lead that comes with the X4R, with the small JST plug at one end, and a standard 3 pin servo connector fitted to the other end. (Ignore the 4th wire.)



10. Now switch on the transmitter, and with a long press of the menu button go into the **Radio Setup** menu system.



11. Page through to screen 2:
12. Then select **FIRMWARES**:
13. Select the appropriate file. **XJT_LBT** will flash the transmitter, the other two files flash the appropriate receivers. Both the transmitter and receivers should be updated to this latest version.



14. Finally, with the model secured, (and if electric, preferably with the propeller removed) check that all the model functions perform as expected, and that failsafes are still working. It should not be necessary to rebind the receiver.

Notes:

Why EU Firmware?

FrSky's X-type RF modules and D16 mode receivers were not compliant with the ETSI EN 300 328 v1.8.1 standard. They issued a number of firmware updates since January 2015 in order to sell on the EU market.

What is an RF module and what's D16?

The RF module handles the communication to the receivers. The Taranis has an internal XJT RF module (physically just a circuit board), and external XJT modules are available for use in other radios. They are generally just referred to as the XJT module, whether they are the internal or external modules.

The XJT module uses several protocols to talk to the receivers. D16 is one of them, and it's used for the X-range of receivers, i.e. X4R, X4R-SB, X6R and X8R. There are two more protocols, D8 and LR12. D8 is used for legacy (i.e. historic) receivers, those starting with D or V in their name, e.g. D4R-II, D8R-II+, V8FR-II, VD5M, etc. Lastly, the LR12 protocol is for the long range receiver L9R.

What is a firmware

The Taranis has two sets of firmware. One is the operating system **OpenTX** (OS), the other is for the RF (XJT) module. These firmwares are independent of each other. Updating the **OpenTX** operating system on your Taranis does not update the XJT module, and vice versa.

If you have a pre-2015 Taranis and buy new D16 mode receivers manufactured for sale in the EU, you wouldn't be able to use them. You can't just update the OS to the latest version and expect it to bind with your new receivers. It's important to understand the difference between the two firmwares.

The Crucial Bit!

This is important, your XJT module and any D16 receivers must match. Either both have EU firmwares, or both have non-EU firmwares. The D8 and LR12 mode receivers are not affected by

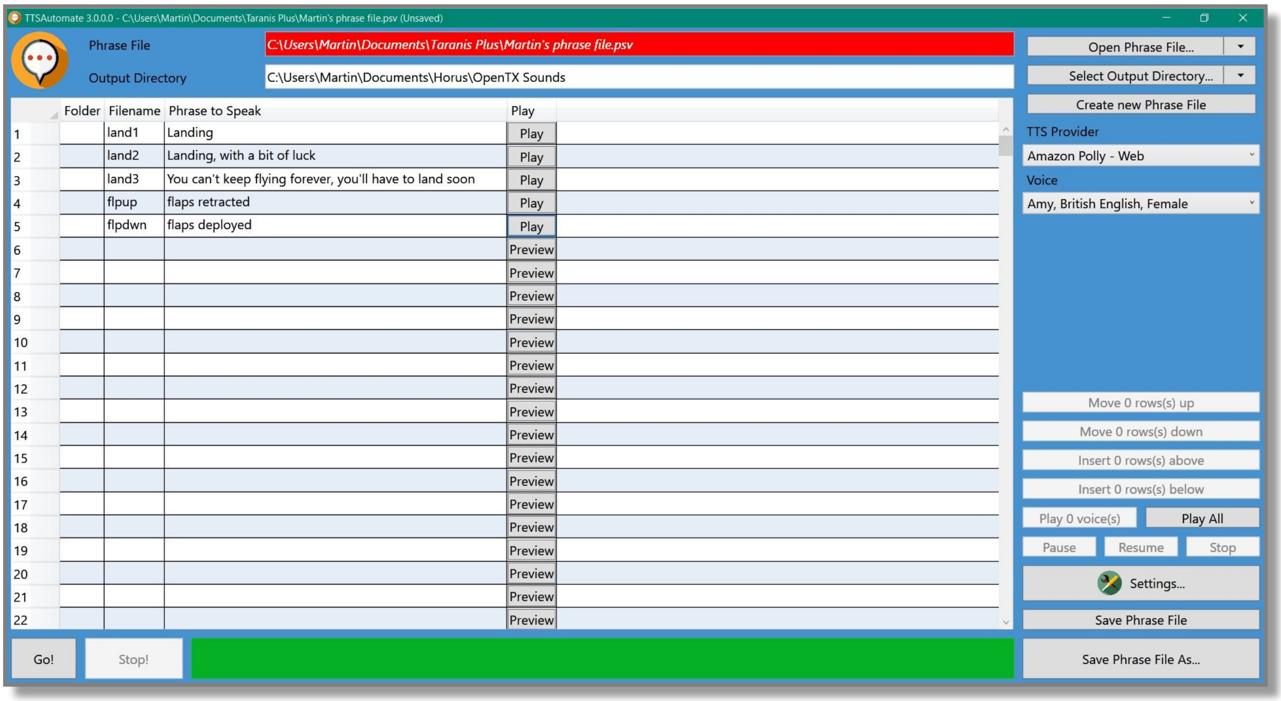
D8 Mode	D16 Mode	LR12 Mode
D4R-II	X4R	L9R
D6FR	X4R-SB	
D8R-II plus	X6R	
D8R-XP	X8R	
Delta-8		
V8FR-II		
V8R4-II		
V8R7-II		
V8R7-SP		
VD5M		

this choice. If you're sticking with EU firmwares, then make sure to upgrade both the XJT and D16 receivers to the LBT firmware. The latest LBT firmware (2016) fixes some issues present in older EU firmwares.

How to Create a Sound File

My thanks go to Liam O'Hagan (CaffeineAU), not only for a super little program but also for allowing me to include it in this documentation.

An important part of **OpenTX** is its ability to use speech to provide information of the status of the model whilst it is flying and its telemetry together with such things as changes to switch positions etc. Essentially, there are two facets to the speech function in **OpenTX**. The first uses in-built functions to play values of particular parameters or telemetry, together with some system announcements. On exploring the SD card in the transmitter, these can be found in the **SYSTEM** subdirectory in the **SOUNDS** directory. The second facet is to simply play the speech file as directed by a **Special Function**. This facet is clearly open ended and there can be as many speech files as the user wishes. A popular file of speech announcements for the UK is the “Amber” file, unfortunately this does not cover every conceivable need for users. It is possible to create extra speech files and this used to be rather a laborious task as one needed to get it in the right format to work. **OpenTX** requires mono “.wav” files saved in a particular format. However, recently an excellent new program has been available to create these speech files especially for **OpenTX**. It is called **TTSAutomate**. TTS stands for text-to-speech, and **TTSAutomate** will convert text into a speech file and save it in the correct format for **OpenTX**. A quick search of the internet will find where this program is available.



TTSAutomate is a superb little program where one simply types in all the extra messages required, and at a press of a button these are saved in the appropriate format using the chosen voice. Not only can the actual .wav files be saved, but the whole phrase file can be saved, so that it can be added to or extended at a later time. There is a plethora of voices provided in numerous languages and accents.

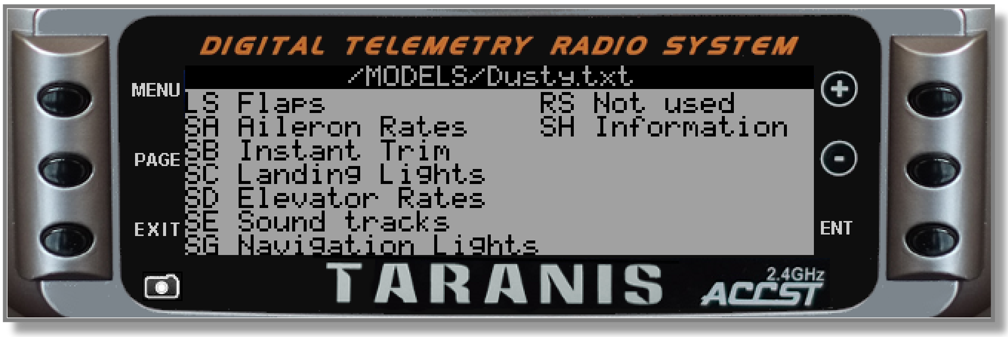
	Folder	Filename	Phrase to Speak	Play
1	wav	land1	Landing	Preview
2		land2	Landing. Anyone got a spare bin bag.	Preview
3		land3	You can't keep flying forever, you'll have to land soon	Preview
4		flpup	flaps retracted	Preview
5		flpdwn	flaps deployed	Preview
6				Preview

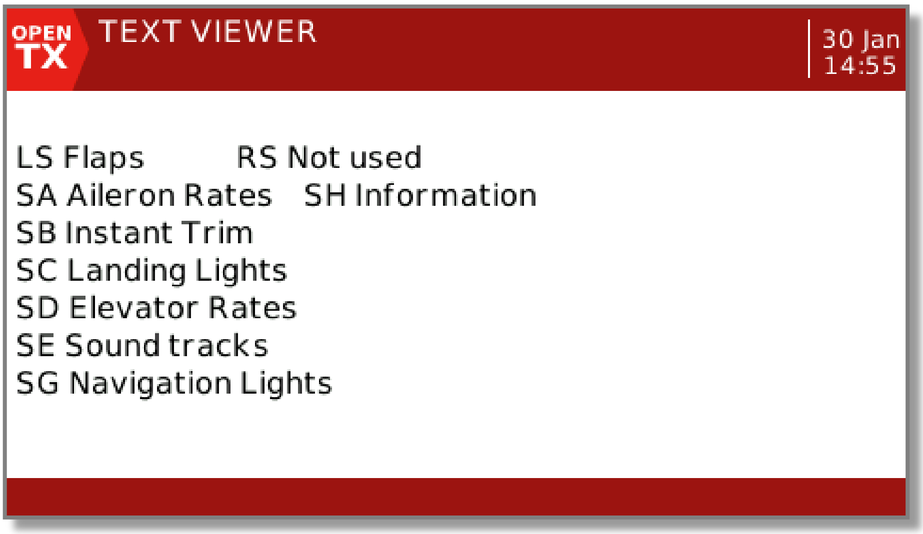
It is important to get the setup of this program right. First, in the **Settings** menu the filename must be set to a limit of 6 characters. This is all that **OpenTX 2.2** will allow. Earlier versions of **OpenTX** would allow 7 characters. Also ensure the **Encode MP3 to WAV** box is clicked. You also need to set an output directory both for where the phrase file will be stored, as well as each individual ".wav" file will be stored. The only thing to be careful of is not to use any characters generally forbidden in filenames such as: **~ " # % & * : < > ? / \ { | }.**

Open Phrase File (or Ctrl-O) opens an existing "psv" or phrases file. **Output Directory** (Ctrl-P) allows the user to choose where to save the audio files. The files will be saved under two directories in the selected folder; one for "mp3", one for "wav". Remember, **OpenTX** requires voice files in a "wav" format. A TTS provider must be selected and a voice selected using the drop down boxes. If you want to modify any of the phrases, edit the phrase in the list, and preview again to hear the change. You will note that the **Play** button changes to **Preview** after you edit the line. If you have modified your phrases file, or created a new phrases file, you can save the file with the **Save Phrases File** (Ctrl-S) button or save the file with a new name using the **Save Phrases File** As... (Ctrl-A) button. You can move lines up and down, by selecting the lines (click on the row header and using the buttons. You can add new rows above / below selected lines, or delete lines. Subfolders in the Folders column are supported. For example: if you specify voice\user, then the files will be generated in _\mp3\voice\user_ and _\wav\voice\user_.

How to Create a Text File

When the radio is turned on or the model is selected, display a check list text file and play an audio file. Text files can also be called up with a long press of the **Enter** button and selecting the **View Notes** option. These are examples of a text file:



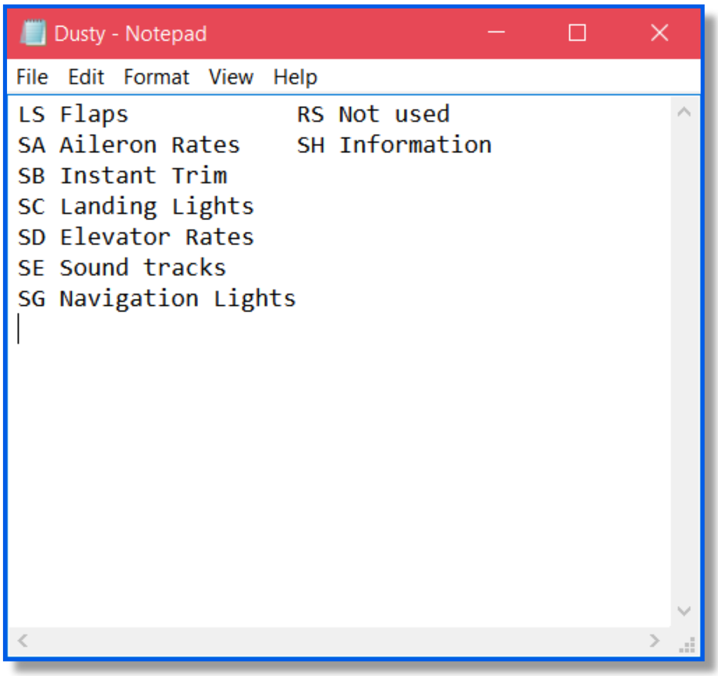


Information that could be included in the text file:

- Switch operation, e.g. flight modes, timer control
- Slider operation, e.g. sound volume control
- Pot operation
- Telemetry data that is valid
- Trainer setup
- Pre-flight check list
- Battery size
- Reminders

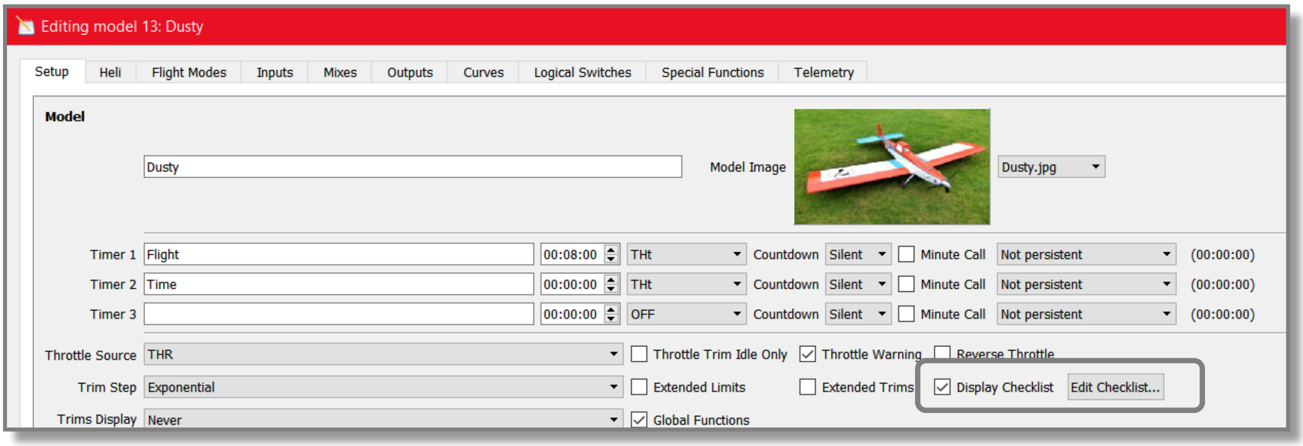
To Create a New Text File

Using a text editor, like Notepad in Microsoft Windows, create a text file with name 'MyModel.txt". MyModel must be exactly the same as the name of the model, without blanks. The width of the Taranis X9 and Horus screens is 35 characters so each line must be limiter to 35 characters - this includes spaces, the X7 is 20 characters wide. The screen has 7 lines below the heading but you can scroll down to more lines. Notice that on the Horus screen, the text does not fill the width of the full screen. This may well be improved at a later date. The text file must be saved in the **MODELS** directory on the SD card, and should also be copied to the SD card mirror on the computer.

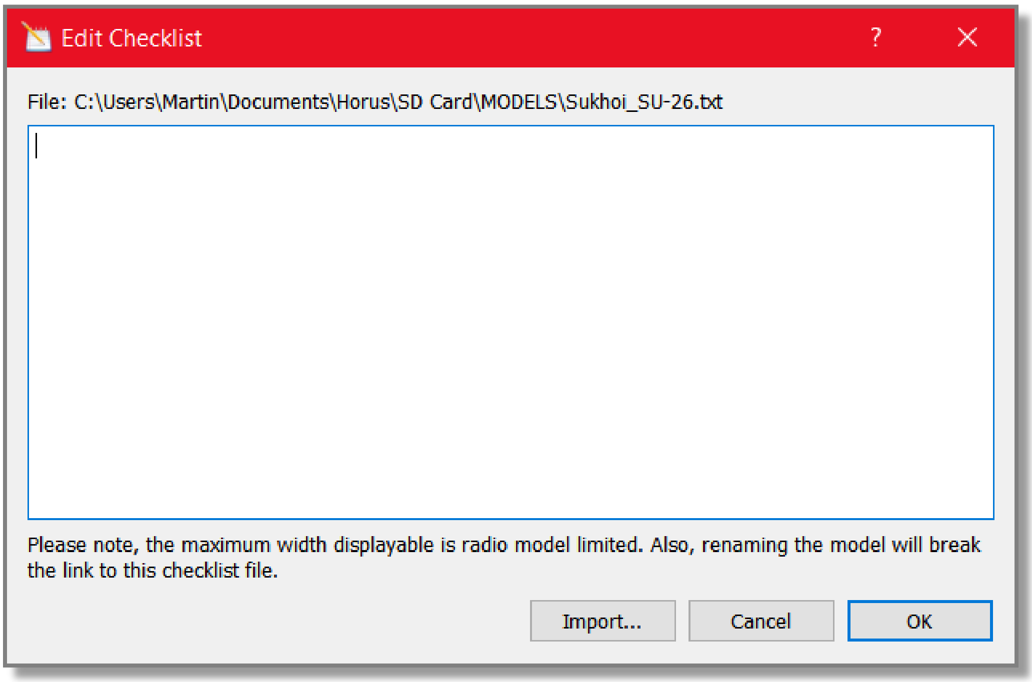


Once a text file has been created it must be enabled on the **Setup** screen for the model.

A new feature of **OpenTX 2.2** is the ability to edit the text file from within **OpenTX Companion**, or to import a text file from another model and then edit it.



Click on the **Edit Checklist** tab, and the following screen will appear:

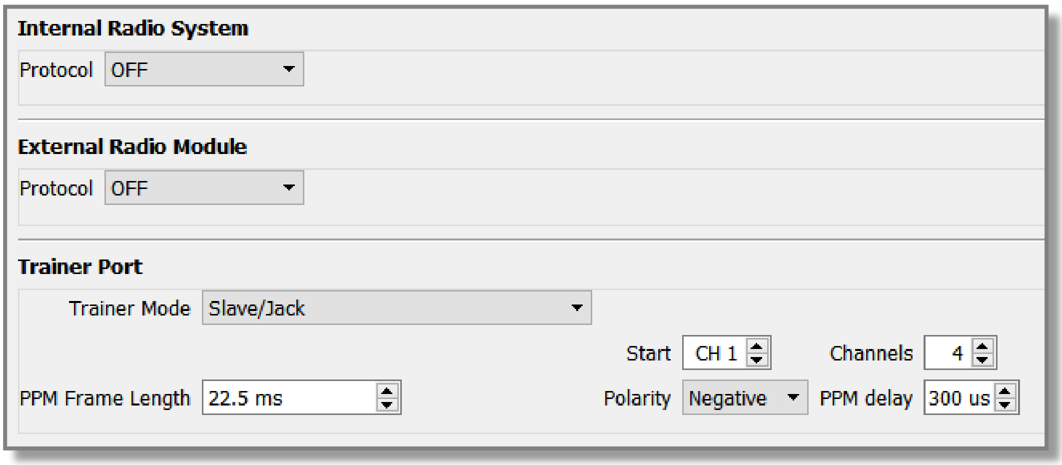


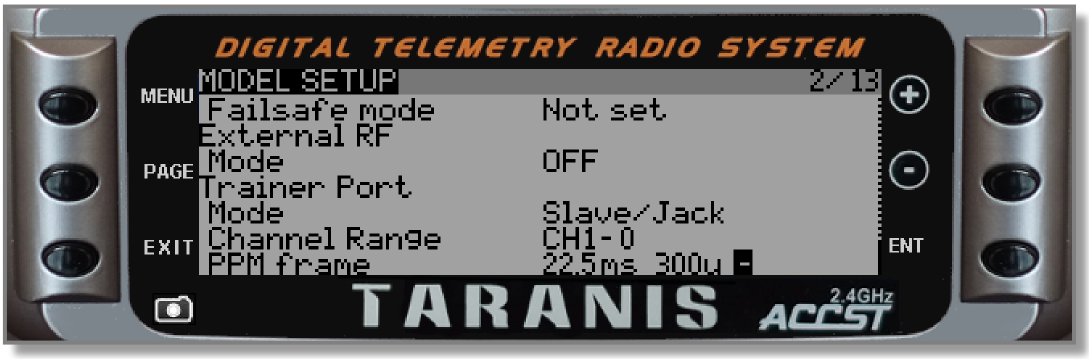
Now a new text file can be created, imported from another model or edited. Do note the word of caution here. If the model name is changed, the user must manually change the text file name also.

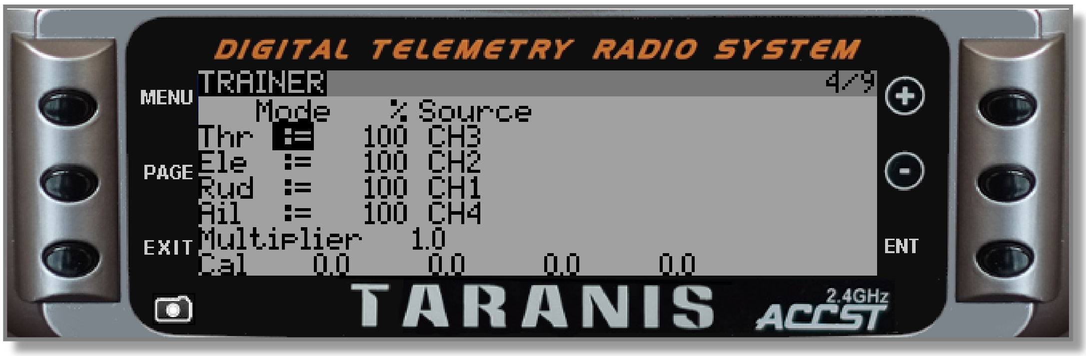
A Trainer System Using Two Taranis Transmitters.

Whether one uses just the two transmitters to set up the buddy system or presets some of the functions first using the **Companion** does not matter.

1. Set up a new model on the **Slave**, taking note of the channel order, and setting the **Trainer Mode** to **Slave**.







OpenTX has much flexibility for use with a buddy box system. The Taranis normally needs a 3.5mm mono jack lead to connect two transmitters together, however a stereo lead will work usually (see note at the end). There are now some third party wireless modules available that fit in the module bay of some transmitters, but not all. The Horus has Bluetooth available also.

The basic buddy box concept is simple. There is a **Master** transmitter, which actually transmits to the receiver, and a **Slave** transmitter where, if possible the RF signal is switched off. The **Slave** transmitter passes a copy of the output signal to the **Master** via the 2 core trainer cable. The **Master** sees these inputs as Input Channels 1 – 16, if all are available on the **Slave**.

The inputs can be calibrated and scaled or multiplied. Each of the inputs is mapped to a joystick and the mapping can include a weighting e.g. 125% and whether the signal is absolute (no master input when the student is in control) or additive where the **Master** and the trainer signals are added together and the resulting sum applied to the model. The second option sounds very clever but having thought about it doesn't really seem that useful.

The Taranis will buddy with other makes of transmitter too. Most Spektrums work although a few will only work with a mono buddy lead. Buddying a Futaba to Taranis you will need a JR to Futaba plug trainer lead. The Horus X12S (but not the X10) has a Futaba style socket also.

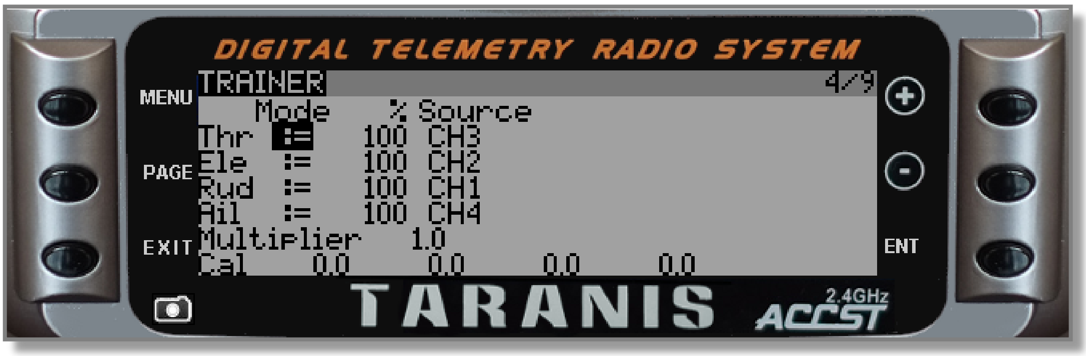
Before Starting

Before you start you need to decide on some basic things:

- If your **Slave** has dual rates will you use these with your student or will you use the dual rates on the **Master** for both instructor and student?
- You can configure the **Master** to take all of the flight control inputs from the buddy box or just some of them. Initially control can be limited, with more control being added to the program or via switches as the student progresses.
- What control will you want to hand control across to the student? An obvious and simple choice would be to use the spring loaded switch **SH** but you could use any of the other physical or logical switches.

Note that the **Internal Radio System** should also be turned off.

2. Connect the **Master** to the **Slave** and go to the **Trainer** page on the **Master**.

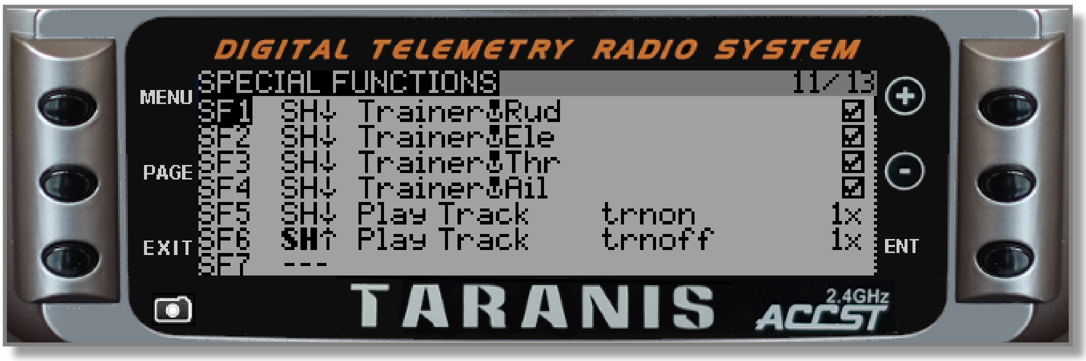


3. Set the four Joystick controls to **Replace :=**. Alter the source as required to get the correct control surfaces operating. The weight should not need altering unless this has been changed on the **Slave**. (With another make of transmitter as the **Slave**, this weighting might well be needed, or it may be simpler to use the **Multiplier**.)

4. Check the throws go roughly between -100 and +100. These are the right hand four numbers on the bottom row of the screen.

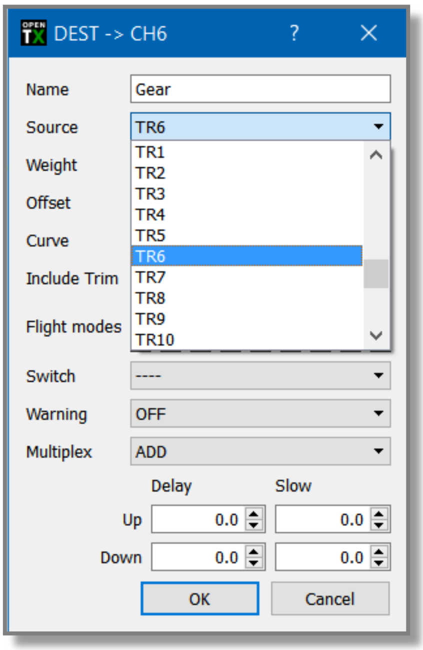
#	Switch	Action	Parameters		
SF1	SH↓	Trainer RUD			
SF2	SH↓	Trainer ELE			
SF3	SH↓	Trainer THR			
SF4	SH↓	Trainer AIL			
SF5	SH↓	Play Track	trnon	▶	No repeat
SF6	SH↑	Play Track	trnoff	▶	No repeat

5. Set all the **Slave** joysticks to their mid positions and click on the **Cal** function to calibrate them.



6. Decide on how control will be passed to the student. For this example switch **SH** will be used in the up position to give control of both joysticks to the student. Here also, voice announcements are given.

   Note, to disable any of the joysticks, simply unclick the "enable" box on the right of the screen.



7. As always, check all the functions work as expected on the model with it safely restrained, or the propeller removed if it is electric.

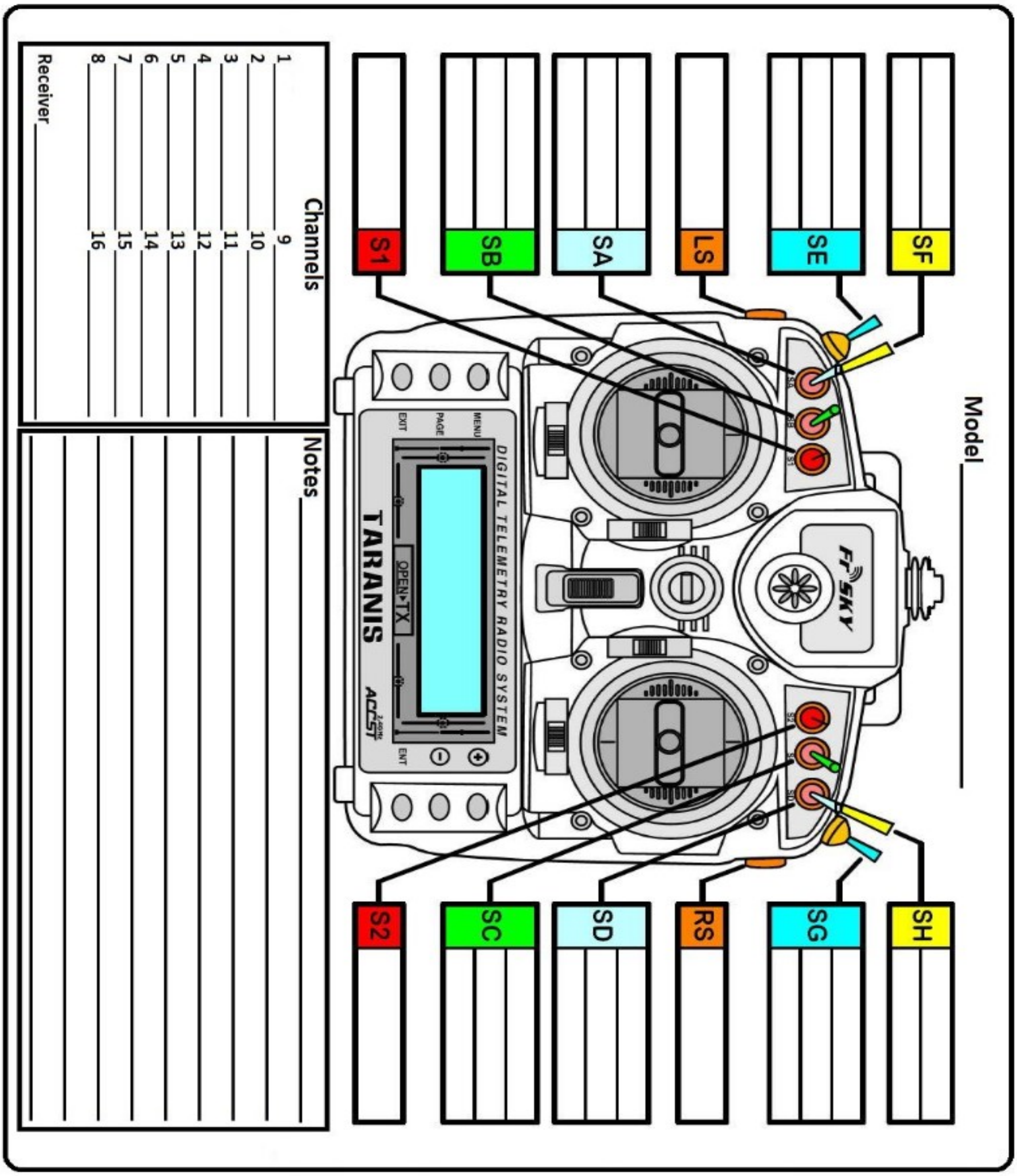
Note 1:

Other slave channels can be used with **OpenTX** if they are available from the Slave transmitter. **TR1** to **TR16** are available on both **Inputs** and **Mixers**. An extension of this could be to have two pilots using two transmitters to control an elaborate 16 channel model, with, say, one pilot handling all the auxiliary functions through their slave transmitter. Thus a joystick on the slave could be used to rotate a gun turret.

Note 2:

It has recently come to light that there is a discrepancy between the wiring of a Taranis X7 and a Horus X10. Thus if you try to buddy these transmitters with a stereo lead, it will not work. A mono lead should, however. This only affects a Taranis X7 to Horus X10 link, not a Taranis X9 or Horus X12.

Once one has set up a few models on the transmitter, the switches and sliders used for each model can get quite confusing. Ideally, keeping the same switches for the same functions is best, but this is not always possible. This chart provides a simple paper method of keeping track of the transmitter settings.



The Flybarless Helicopter

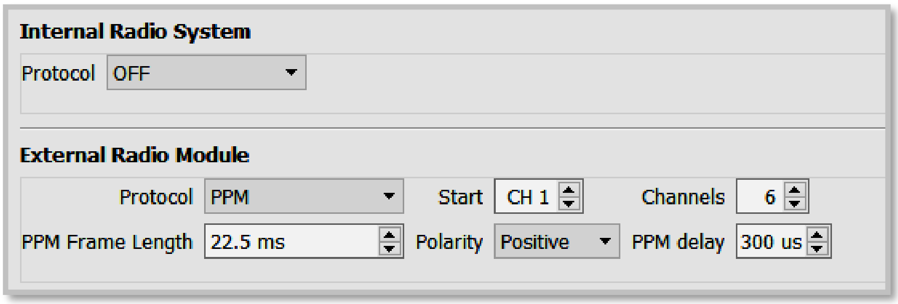
First, to understand how a flybarless (FLB) system works one needs to know how and what a flybar does. A very simple explanation of a flybar function is to add stabilization to the rotor disc by automatically changing the cyclic pitch angles of the rotor blades to help improve cyclic stability and make cyclic control much more manageable. As the name suggests, FBL does away with the flybar and with the help of electronic stabilization systems, "virtually" replaces the flybar (why they are also called “virtual” or "electronic flybars").

The FBL helicopter is much easier to program using **OpenTX** because it does not require the specialist helicopter functions of **OpenTX**. The following example is taken from a FBL helicopter, a Blade 300X fitted with a Spektrum AR7200BX AS3X® (Artificial Stabilization - 3aXis) flybarless system with built in receiver.

The first problem comes with the built in Spektrum receiver. Obviously the normal Taranis transmitter will not work with the Spektrum receiver, so an external XJT module is required. An external module, such as one shown below fits in the opening at the back of the transmitter and has its own aerial.

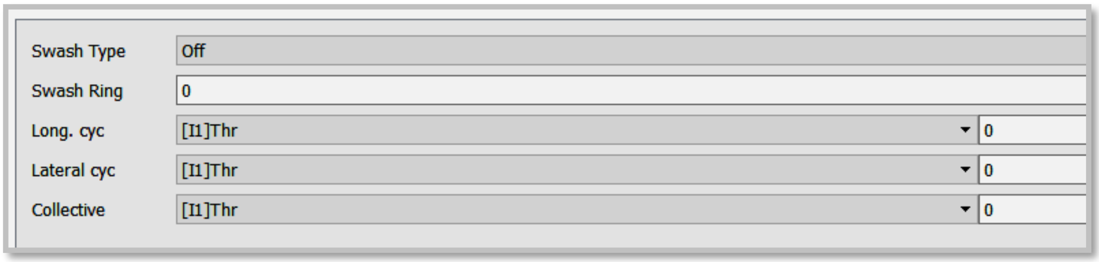
For those not familiar with Spektrum, the black plug at the bottom right is the bind plug. There is a bind socket on the Spektrum receiver, and this plug is inserted before switching on the receiver to bind it to the Orange module.

The first thing, therefore, on the helicopter setup is to tell the transmitter to use this external module. On the **Model Edit Setup** screen, the radio needs configuring to switch off the internal radio and switch on the external module.



The protocol needs to be set to **PPM** for this Orange module, not DSMX or DSM2 as one might expect, and the polarity needs to be positive. The **Frame Length**, **Channels** and **PPM delay** are the defaults that come up when **PPM** is selected.

Now one would think the **Heli** page needs setting, but actually the AS3X module does all the clever manipulation instead, so the **Heli** page is not changed from the default settings:



The rest of the programming is simply a copy of the settings provided for a Spektrum transmitter. The Blade 130X User Guide offers programming for a number of transmitters, it is better to copy the settings from a mid to high end transmitter as **OpenTX** will be able to duplicate all the functions. The Blade 300X User Guide can be downloaded on line to compare settings. Essentially we are going to set up **OpenTX** to provide the same settings as those given for the Blade 130X. However, changes can be made to suit individual preferences without moving away from these basic settings. For instance, as the Taranis has mostly 3-way switches, we might as well use triple rates rather than double.

Blade 130X User Guide Settings

The transmitter is set to Mode 2.

The Blade User Guide for the Spektrum DX8 transmitter suggests:

1. Servo travel is 100. (Remember Spektrum servo travel is less than **OpenTX**. We will compensate for this later.)
2. There are no sub-trims set.
3. Elevator, rudder and pitch channels are reversed.
4. There are three flight modes, Normal, Stunt 1, and Stunt 2. No expo is set for any flight mode.
5. Dual rates are set at 100 for Normal mode, 85 for elevator, aileron and rudder in stunt modes 1 and 2.
6. Throttle and pitch curves are shown later, and there are no tail curves.
7. Timer is set to 4 minutes.

OpenTX Settings

Flight Mode: **SE↑** Normal

SE- Idle-Up 1

SE↓ Idle-Up 2

Obviously the choice of switch is down to personal preferences.

Throttle cut: **SF↑** Throttle cut enabled

SF↓ Throttle cut disabled

Throttle joystick must be off before toggling the switch to enable the throttle.

Triple Rate: **SD↑** High D/R

SD- Mid D/R

SD↓ Low D/R

Timer: Timer 1: Engine time. Count down starts once throttle opened and remains above 25%.

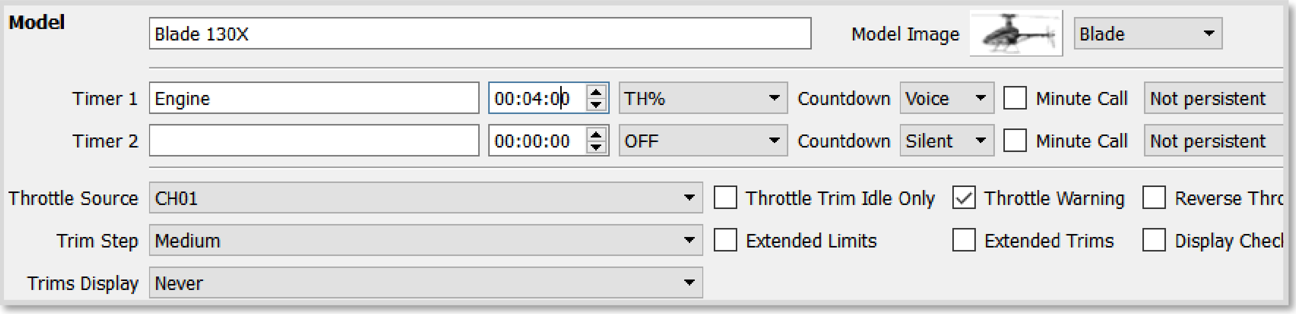
Servos limit: Set to **±80%** in order to match the Spektrum limits.

Curves: Curves are from the user guide.

Curve1	Normal throttle curve
Curve2	Idle-Up 1 throttle curve
Curve3	Idle-Up 2 throttle curve
Curve4	Normal pitch curve
Curve5	Idle-Up 1 & 2 pitch curve

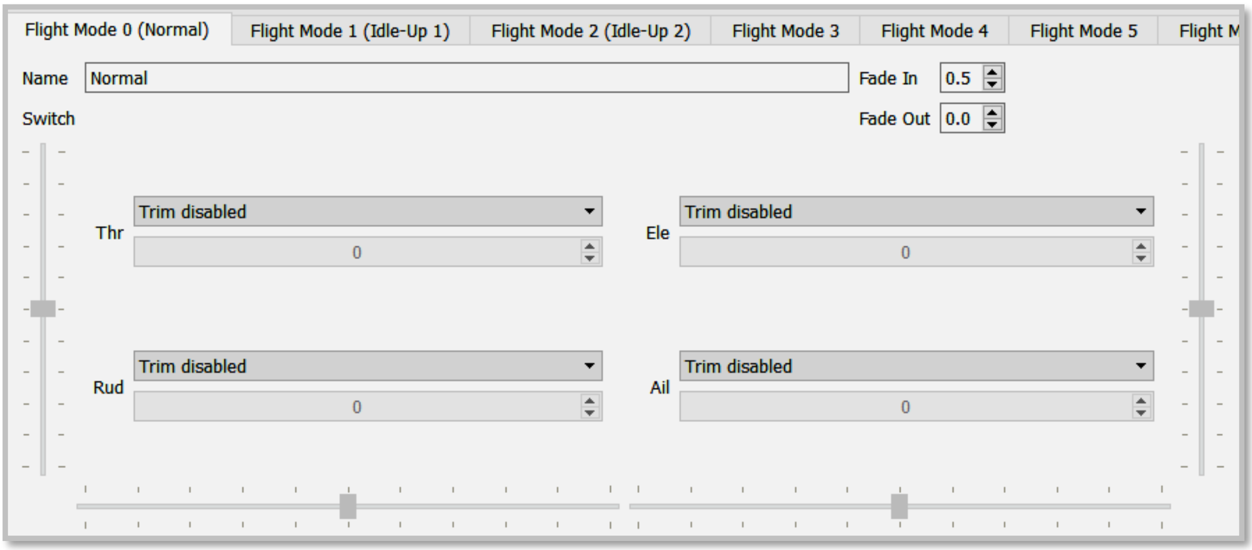
1. Setting timers etc.

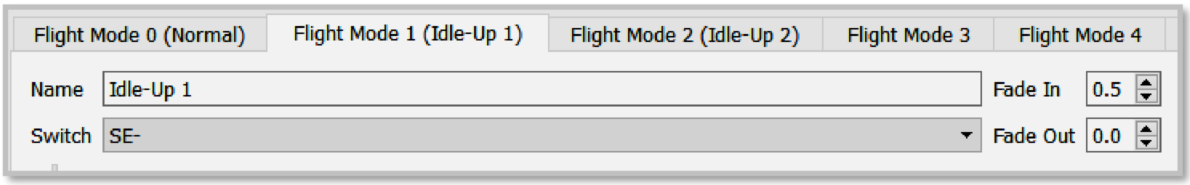
Timer 1 is set to count down from 4 minutes once the throttle is opened more than 25%.

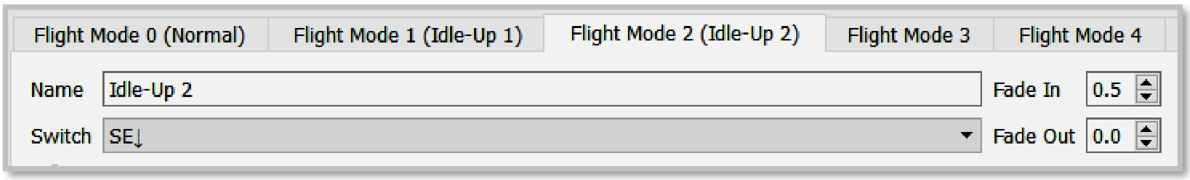


2. The Flight Modes Screen

Three flight modes need setting, **Normal**, **Idle-up 1** and **Idle-up 2**. Switch **SE** is used here. The Blade User Guide advises that no trims must be used as the AS3X receiver interprets those as signal inputs. These will need to be disabled in each of the **Flight Modes** screens.







One can alter the fade in times to suit personal preference.

3. The Inputs Screen

```
[I1]Thr         Thr Weight(+100%)
[I2]Ail         Ail Weight(+100%) Expo(25%) Switch(SD↑) [Hi rate]
                Ail Weight(+85%) Expo(20%) Switch(SD-) [Med rate]
                Ail Weight(+70%) Expo(15%) Switch(SD↓) [Lo rate]
[I3]Ele         Ele Weight(+100%) Expo(25%) Switch(SD↑) [Hi rate]
                Ele Weight(+85%) Expo(20%) Switch(SD-) [Med rate]
                Ele Weight(+70%) Expo(15%) Switch(SD↓) [Lo rate]
[I4]Rud         Rud Weight(+100%) Switch(SD↑) [Hi rate]
                Rud Weight(+85%) Switch(SD-) [Med rate]
                Rud Weight(+70%) Switch(SD↓) [Lo rate]
Input05
```

1. One <u>must</u> follow the Spektrum channel order, not, perhaps the one you normally use. Spektrum is TAER. This does not have to be set on the **Radio Edit Setup** screen however.
2. The recommended Blade setup does not have expo, however this is down to personal preference, and experimentation.
3. While these rates do not match the Blade User Guide, medium rates match the suggested rates for Sport 1 and Sport 2. For normal mode, low rates provide a better setting for the beginner. High rates matches the User Guide settings. Indeed, low rates could be changed down to 50% for learning to hover.
4. While no expo is recommended in the Blade User Guide, this is very much down to personal preference.

4. The Mixes Screen

```
CH01 (Thr)        [I1]Thr Weight(+100%) Flight mode(Normal) Curve(1) [Normal]
               += [I1]Thr Weight(+100%) Flight mode(Idle-Up 1) Curve(2) [IdleUp1]
               += [I1]Thr Weight(+100%) Flight mode(Idle-Up 2) Curve(3) [IdleUp2]
               := MAX Weight(-100%) Switch(L2) [CutOff]
CH02 (Ail)        [I2]Ail Weight(+100%)
CH03 (Ele)        [I3]Ele Weight(+100%)
CH04 (Rud)        [I4]Rud Weight(+100%)
CH05 (Gyro)       MAX Weight(+60%) Switch(!SC-) [Headmode]
               += MAX Weight(-60%) Switch(SC-) [Ratemode]
CH06 (Pitch)      [I1]Thr Weight(+100%) Flight mode(Normal) Curve(4) [Normal]
               += [I1]Thr Weight(+100%) Flight modes(Idle-Up 1, Idle-Up 2) Curve(5) [IdleUp]
```

The first three throttle settings are for the three flight modes. The last throttle setting links to a logical switch, **L2** to disable the throttle. Notice the **Multiplex** setting is set to **Replace**. This is very important for the throttle cut to work correctly. Channel 5, the gyro, and channel 6, the pitch are the standard channels for the Spektrum AS3X. Switch **SC** switches between heading hold and rate mode for the gyro.

5. The Outputs screen

Spektrum systems normally work from -80 to +80, so the weighting needs to be reduced here. Aileron, elevator and pitch are reversed as in the settings provided in the Blade user guide.

#	Name		Subtrim		Min		Max	Direction
CH1	Thr	GV	0.0%	GV	-80.0%	GV	80.0%	---
CH2	Ail	GV	0.0%	GV	-80.0%	GV	80.0%	INV
CH3	Ele	GV	0.0%	GV	-80.0%	GV	80.0%	INV
CH4	Rud	GV	0.0%	GV	-80.0%	GV	80.0%	---
CH5	Gyro	GV	0.0%	GV	-80.0%	GV	80.0%	---
CH6	Pitch	GV	0.0%	GV	-80.0%	GV	80.0%	INV

6. The Curves screen

In **OpenTX** curves go from -100 to +100. Many curves are listed in helicopter manuals using a 0 to 100 notation. Thus it is necessary to convert these to **OpenTX** curve format.

To convert 0 to 100 to **OpenTX** notation, multiply the 0 to 100 figure by 2 and take away 100.

E.g. 60 in 0/100 becomes 20 in **OpenTX** format

(60 * 2 = 120, 120 minus 100 = 20).

Throttle Curve			
Curve	**Function**	**Suggested Setting**	**OpenTX Setting**
Curve 1	N	0, 25, 50, 60, 70	-100, -50, 0, 20, 40
Curve 2	1	100, 70, 60, 70, 100	100, 40, 20, 40, 100
Curve 3	2	75, 75, 75, 75, 75	50, 50, 50, 50, 50

Pitch Curve			
Curve	**Function**	**Suggested Settings**	**OpenTX Settings**
Curve 4	N	30, 40, 50, 75, 100	-40, -20, 0, 50, 100
Curve 5	1, 2, H	0, 25, 50, 75, 100	-100, -50, 0, 50, 100

Note that curves 3 and 5 are straight lines, therefore a two-point curve could be used for each, simply using the endpoints, or even simply using weightings instead of curves. However, curves allow for later refinement of the flying characteristics.

7. Logical Switches and Special Functions

Having completed most of the basic setting up to match the Blade User Guide, logical switches and special functions are used to provide a throttle cut and switch verbal feedback.

These logical switches provide a simple method of providing a throttle cut which cannot be accidentally enabled when the throttle is not at a minimum.

#	Function	V1	V2	AND Switch
L1	a<x	Thr	-95	SF↓
L2	Sticky	SF↑	L1	----

#	Switch	Action	Parameters		
SF1	ON	Volume	S2		ON
SF2	FM0	Play Track	nrmmod	▶	No repeat
SF3	FM1	Play Track	idlup1	▶	No repeat
SF4	FM2	Play Track	idlup2	▶	No repeat
SF5	SF↑	Play Track	engdisa	▶	No repeat
SF6	SF↓	Play Track	engarm	▶	No repeat
SF7	SD↑	Play Track	rates_h	▶	No repeat
SF8	SD-	Play Track	midrates	▶	No repeat
SF9	SD↓	Play Track	rates_l	▶	No repeat
SF10	!SC-	Play Track	hedhld	▶	No repeat
SF11	SC-	Play Track	ratemd	▶	No repeat

SF1 puts a volume control on pot **S2**.

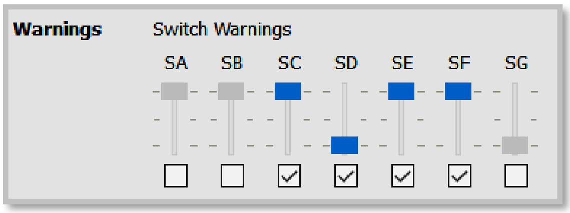
SF2 to **SF4** give a voice alert for the three modes, though now sport 1 and sport 2 are called by their more normal names, **Idle Up 1** and **Idle Up 2**.

SF5 and **SF6** give a voice alert for the motor arm/disarm.

SF7 to **SF9** give a voice alert for the rates.

SF10 and **SF11** give voice alerts for the gyro mode.

Finally, now the switches have been assigned, back on the **Setup** screen the switch warnings can be set. It is obviously very important to set **SF**, the motor disable switch warning.



The CCPM Electric Helicopter

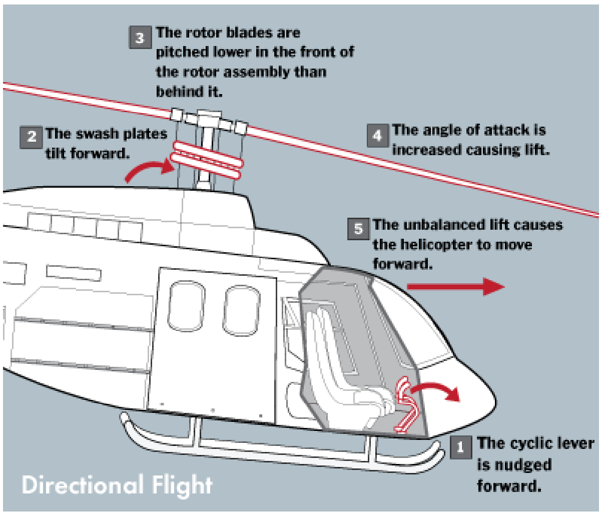
This "How To" deals with setting up a CCPM electric helicopter. This example will be based on a 450 sized electric flybar helicopter with a 120° swash plate, and a mode 2 transmitter.

CCPM is an acronym that stands for Cyclic - Collective - Pitch - Mixing. This feature is only found on RC helicopters with collective pitch (as the name suggests), it doesn't apply to fixed pitch helicopters. In the previous "How to", the mixing for the three servos that control the collective were handled within the specialised gyro stabilized control system and combined receiver. This system handles all the clever mixing required to get the (usually) three servos linked to the swash plate to function correctly.

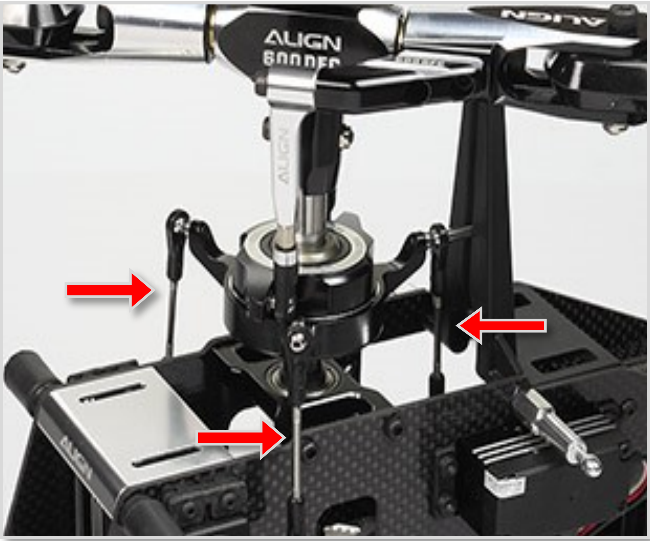
The complex collective cyclic pitch mixing for these three (or occasionally four) servos can also be handled by an appropriate transmitter. **OpenTX** offers excellent features to allow CCPM helicopters to be programmed, and thus a simple 6 channel receiver with a separate tail gyro can be utilised in the helicopter. The downside is that one needs to have a greater understanding of how such CCPM helicopters work, though maybe that is not really a downside, just a steeper learning curve. The positive side is, of course, the greater flexibility offered by **OpenTX** together with a better understanding of how that system works to be able to tune the helicopter more effectively.

The reader is advised to use this section in conjunction with Part 3 of the Reference Guide where the **Heli** page is covered in the **Model Editor**.

For collective pitch RC helicopters, the direction the helicopter will travel and the amount of lift is governed by the swashplate. The bottom part of the swash plate stays aligned with the helicopter body and does not rotate, the top part of the swash plate turns with the helicopter blades. The top part of the swashplates determines the angle of the blades as they rotate. To tilt forward and back or sideways requires that the controls alter the angle of attack of the main rotor blades cyclically during rotation, creating differing amounts of lift at different points in the cycle. This is called the cyclic pitch or just cyclic. Strictly speaking, cyclic is the combination of aileron and elevator. To increase or decrease



overall lift requires that the controls alter the angle of attack (collective pitch, or just collective) for all blades collectively by equal amounts at the same time, resulting in ascent, descent, acceleration and deceleration. The cyclic pitch control changes the angle of selective rotor blades as they spin, so if the blade on one side produces slightly more lift the opposite blade always produces slightly less lift, thus steering the helicopter left or right. Similarly if the blade tilts forward it will produce more lift making it move forward.



Three servos vary the height and angle of the swash plate. The group of 3 servos that change the swashplate height and angle are called cyclic servos. This causes a problem with raw signals. For example, with a single elevator signal one cyclic servo must go up and another must go down. This changing of the control signals to cyclic signals is called CCPM. This photo shows a typical 120° helicopter swashplate. The round device on the main rotor shaft is the swashplate. The three links connecting the swashplate to the three servos can be clearly seen though the servos themselves are not visible.

1. The Inputs Screen

The first part of setting up the helicopter is quite straightforward as we set up the **Model Editor Inputs** screen and allow for triple rates and expo. These should be adjusted for personal preference.

```
[I1]Thr        Thr Weight(+100%)
[I2]Ail        Ail Weight(+100%) Switch(SD↑)
               Ail Weight(+85%) Expo(20%) Switch(SD-)
               Ail Weight(+70%) Expo(30%) Switch(SD↓)
[I3]Ele        Ele Weight(+100%) Switch(SD↑)
               Ele Weight(+85%) Expo(20%) Switch(SD-)
               Ele Weight(+70%) Expo(30%) Switch(SD↓)
[I4]Rud        Rud Weight(+100%) Switch(SD↑)
               Rud Weight(+85%) Expo(20%) Switch(SD-)
               Rud Weight(+70%) Expo(30%) Switch(SD↓)
```

2. The Heli Screen

The complex mixing that is going to take place to convert the basic aileron and elevator inputs into the cyclics is set up in the **Heli** screen. That is further complicated by the collective. The transmitter only has four joysticks to control five functions, engine speed, collective, rudder (yaw), aileron (lateral cyclic) and elevator (longitudinal cyclic). Thus the throttle controls both the collective and the motor speed using curves to enable both to function correctly.

Swash Type	120	
Swash Ring	100	
Long. cyc	[I3]Ele	-60
Lateral cyc	[I2]Ail	60
Collective	CH9	60

1. The swash type is set to 120.
2. The swash ring is used to limit the swash movement commanded by cyclics alone (effectively roll rate or swash tilt). The range is from 0 to 100. The maths behind **Swash Ring** is a cubic function, not linear. This means with a swash ring 60, the max defection of a cyclic channel is about 50. This will not impact pitch movement up and down. You can still hit servo limits at extremes of pitch and cyclic. If you need to prevent this, adjust the mix weights of the pitch control mixes (i.e. +60% instead of +100%).
3. Now the cyclics and collective are identified, and weightings are added. Once all the **OpenTX** settings are entered, the direction of each servo can be checked. Any reversal can be carried out here (e.g. **Long. cyc**) or on the **Outputs** screen.
4. The collective is set to channel 9. This will be explained in the **Mixes** screen below.

3. The Mixes Screen.

It is on this **Mixes** screen is where the process really becomes clear. First, we need to delve into the problem of the one control handling two functions, motor speed and the collective. A phantom channel is created to handle this. As was seen above, channel 9 was used. Why channel 9? We could actually use any unused channel, and in **OpenTX** there are 32. If one assumes an 8 channel receiver, and allows for all channels being used, then channel 9 is the logical choice, and when using the **Companion**, it will still be visible on the screen with the other channels being used. Thus one can see the complete picture on the screen at the same time. Channel 9 is then the collective source for the swash, as defined on the **Heli** screen.

```
CH01         [I1]Thr Weight(+100%) Switch(SE↑) NoTrim Curve(1) [Norm]
          += [I1]Thr Weight(+100%) Switch(SE-) NoTrim Curve(2) [Idle Up]
          += [I1]Thr Weight(+100%) Switch(SE↓) NoTrim Curve(3) [3D]
          := MAX Weight(-100%) Switch(L2) [CutOff]
CH02         CYC1 Weight(+80%) [Pitch]
CH03         CYC2 Weight(+50%) [Roll]
CH04         CYC3 Weight(+50%) [Roll]
CH05         Rud Weight(+70%) [Yaw]
CH06         MAX Weight(+60%) Switch(!SC-) NoTrim [gyrohold]
          += MAX Weight(-60%) Switch(SC-) NoTrim [gyrorate]
CH07
CH08
CH09         [I1]Thr Weight(+70%) Switch(SE↑) NoTrim Curve(4) [Collect]
          += [I1]Thr Weight(+50%) Switch(SE-) NoTrim Curve(5) [Collect]
          += [I1]Thr Weight(+70%) Switch(SE↓) NoTrim Curve(6) [Collect]
CH10
```

At first sight this looks complicated. There are three modes, switched using **SE**. There is **normal mode**, **Idle 1** or **sport flight**, and **idle 2** or **3D flight**. Here, actual **Flight Modes** have not been programmed. On the throttle, each flight mode has a different curve. The weight is left at 100%. For the throttle, the weight is varied using the curves. Switch SF disables the motor, and the **Multiplex** is set to **Replace (:=).** Similarly channel 9 is programmed for the pitch. Each flight mode will also have its own associated pitch. Next, instead of assigning aileron, elevator and pitch to channel outputs 2, 3 and 4 which will control the three servos round the swash plate, assign **CYC1**, **CYC2** and **CYC3**. **OpenTX** will do all the clever maths to ensure that each servo moves in the right proportions. What these do is mix aileron, elevator and pitch together and output signals for the servos on channels 2, 3 and 4. For 120° swash type, **CYC1** is the servo in line with the helicopter from nose to tail.

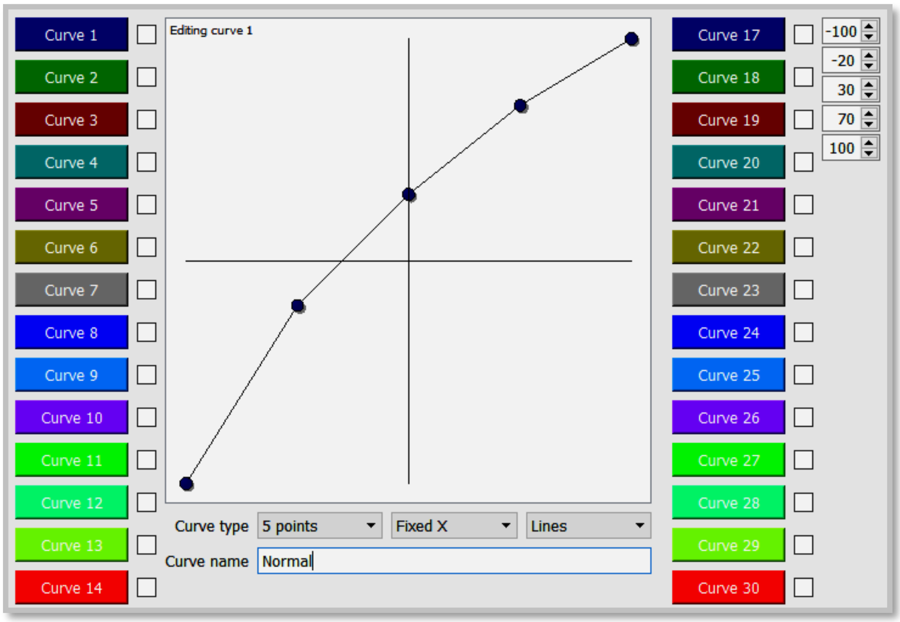
Channel 6 controls the gyro function, either rate mode or heading hold. The handbook for the gyro will give appropriate weightings for this, though they will probably be in 1-100 format rather than the -100,to 100 **OpenTX** format.

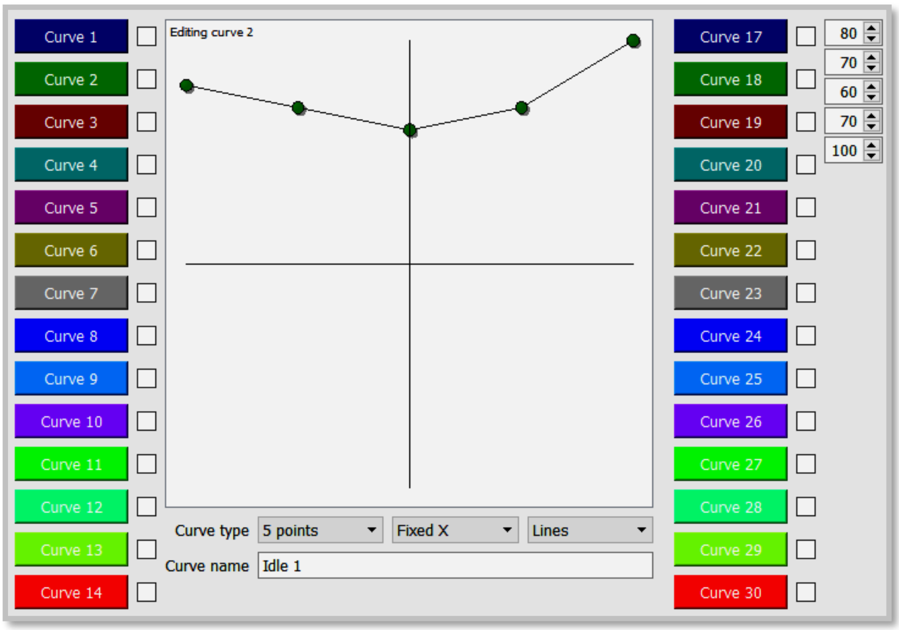
> Ensure **CYC1** is the servo in line with the helicopter travel. CYC2 is CCW round from CYC1 and CYC3 is the matching pair to CYC3. If your swash points are the other way around (ie. CYC1 is South, not North), then reverse the aileron and elevator in the **Heli**-setup page. This way the pitch will still work correctly on the mix, just you have told the TX that the swash is backwards to the TX's expected swash orientation.

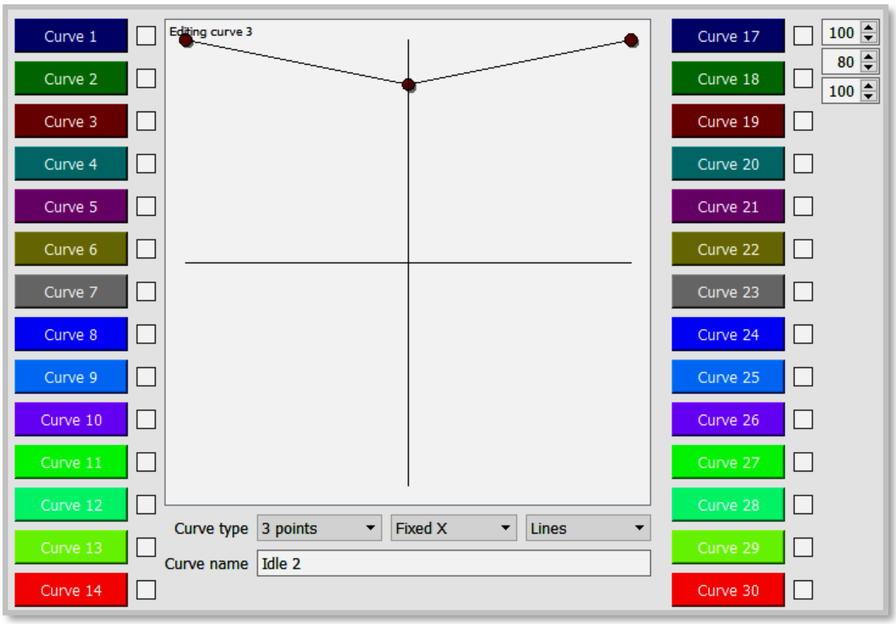
4. The Curves Screen.

The starting point for the curves are the curve guidelines given in the helicopter handbook. Thus:

Throttle Curve			
Curve	**Function**	**Suggested Setting**	**OpenTX Setting**
Curve 1	N	0, 40, 60, 85, 100	-100, -20, 30, 70, 100
Curve 2	Idle 1	90, 85, 80, 85, 100	80, 70, 60, 70, 100
Curve 3	Idle 2	100, 80, 100	100, 60, 100





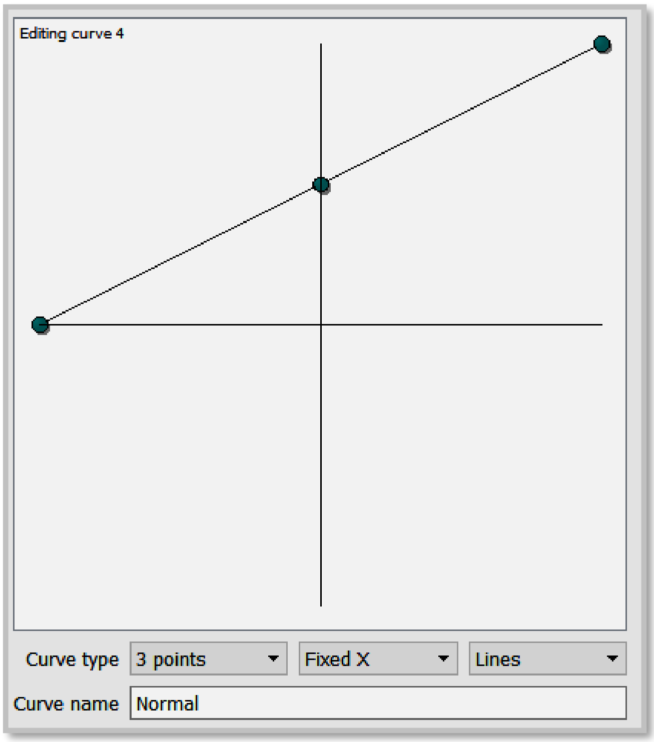


Clearly with having the option of 17 point curves, one could develop these curves to give a much more gradual transition. The whole idea behind a throttle curve is to keep the head speed of a collective pitch helicopter as consistent as possible throughout the collective pitch range. The helicopter handbook makes a good starting guide, but the later process of setting up and testing will allow this curve to be much refined. Notice Idle-up curve 1 and Idle-up curve 2 are almost straight lines. IC engines tend to have a more pronounced curve. This is because electric motors produce very linear torque, the torque is instantaneous, and their speed is directly relative to the voltage they get, not by the load placed on them. More load simply means they will draw more current but the RPM will remain the same (more or less). By their very nature, electric motors are self governing and this is why flat-line, or almost flat-line throttle curves work so well.

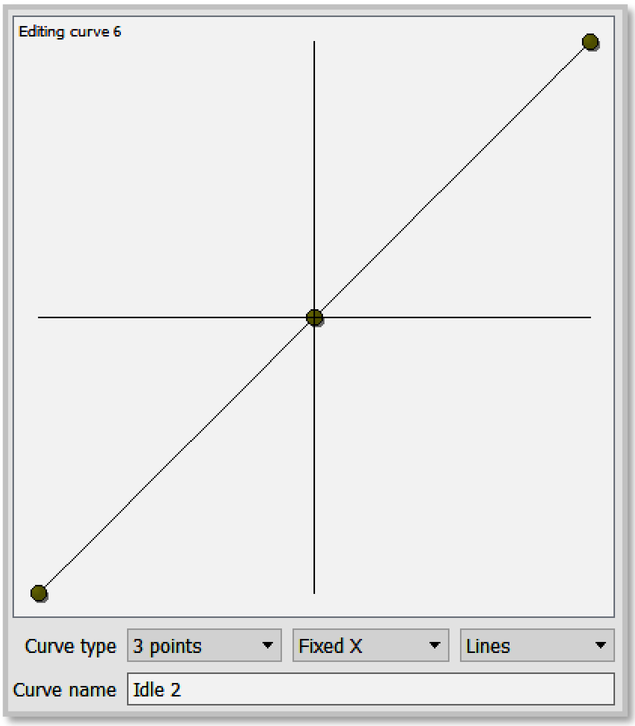
Pitch Curve			
Curve	**Function**	**Suggested Settings**	**OpenTX Settings**
Curve 4	N	0^o, $+5^o$, $+9^o$ to $+11^o$	0, 50, 100
Curve 5	Idle 1	-5^o, $+5^o$, $+9^o$ to $+11^o$	-50, 50, 100
Curve 6	Idle 2	-9^o to -11^o, 0^o, $+9^o$ to $+11^o$	-100, 0, 100

The pitch curves are determined in a similar way.

This is not quite so simple, as the settings are given in angles. Assuming 10^o is full servo travel in one direction, each degree equates to about a weighting of 10. Once the settings can be tried on the helicopter, then either weightings here can be adjusted, or more simply the **CYC** weightings can be altered. Again the three point curves can be improved by using a larger number of points to


Editing curve 4
Curve type
3 points
Fixed X
Lines
Curve name
Normal


Editing curve 5
Curve type
3 points
Fixed X
Smooth
Curve name
Idle 1


Editing curve 6
Curve type
3 points
Fixed X
Lines
Curve name
Idle 2

4. The Outputs Screen.

For now the **Outputs** screen is left blank. It may be needed once the actual helicopter is set up.

6. The Logical Switches Screen.

Sticky and **SF** are used here to create a motor arm switch that will only arm if the throttle is at less than -95%. This is mixed on the **Mixes** screen.

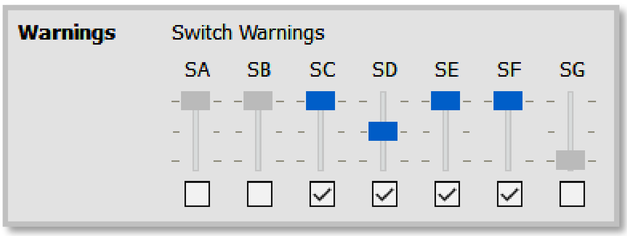
#	Function	V1	V2	AND Switch
L1	a<x	Thr	-95	SF↓
L2	Sticky	SF↑	L1	----

7. The Special Functions Screen.

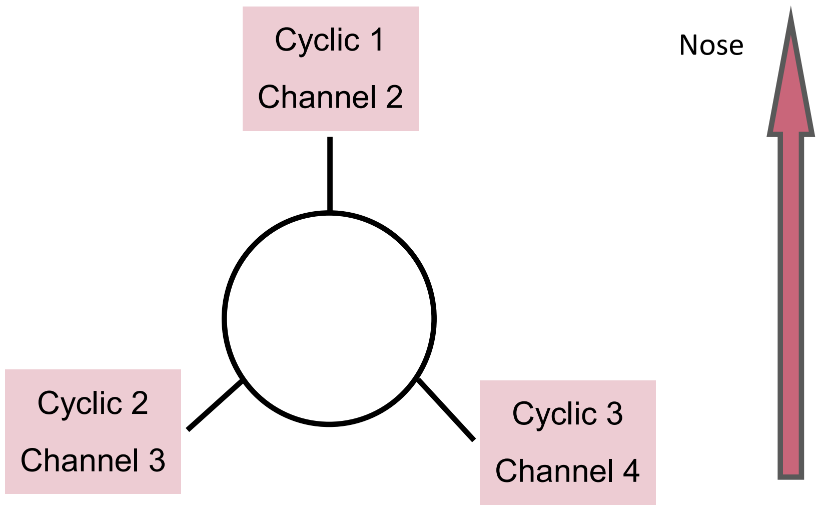
This screen is mostly composed of announced switch warnings.

#	Switch	Action	Parameters		
SF1	ON	Volume	S2		☑ ON
SF2	SF↓	Play Track	engarm	▶	No repeat
SF3	SF↑	Play Track	engdisa	▶	No repeat
SF4	SE↑	Play Track	nrmmod	▶	No repeat
SF5	SE-	Play Track	idlup1	▶	No repeat
SF6	SE↓	Play Track	idlup2	▶	No repeat
SF7	!SC-	Play Track	gyrhh	▶	No repeat
SF8	SC-	Play Track	gyrrate	▶	No repeat
SF9	ON	Play Track	heli3d	▶	No repeat
SF10	SD↑	Play Track	hirates	▶	No repeat
SF11	SD-	Play Track	midrates	▶	No repeat
SF12	SD↓	Play Track	lowrates	▶	No repeat

Apart from remembering to set the switch warnings on the **Edit Model Settings** page, that is all the pre-setup required.



The following basic steps are to ensure CCPM works for the heli. The assumption is that the receiver is mounted on the heli and connected to the cyclic servos. The transmitter must be able to move the servos and you should be able to see the swash move. (i.e. receiver and servos must be powered on, servo horns attached and links to swash on).



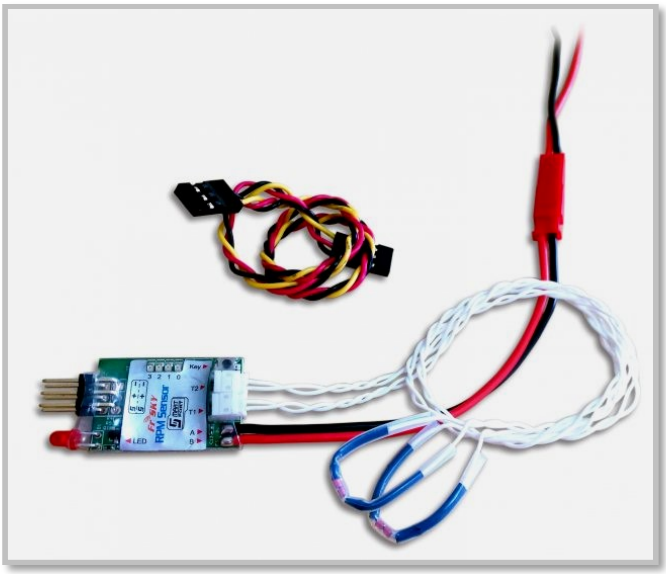
1. Ensure the horns are near 90 degrees when at mid stick. (TH has linear pitch curve and you can use the monitor screen to ensure the transmitter is putting out 0 on Aileron, Elevator and Pitch channels). We can sub-trim them after mix correction is finished.
2. Next is "Find the odd one out". This is where you raise and lower the swash with pitch only and find which servo (if any) goes a different way. The servo that moves different to the others, invert/reverse in the servos screen. You will only need to reverse 1 servo. It does not matter if the swash goes up at lowest collective. This will be changed later if needed.
3. Ensure the elevator (nose-up/down) works the correct way. Centre the collective (swash middle) and pull back on the stick. If the front of the swash plate goes up good. If not, change the long. cyc. direction to **INV** in the swash menu on the **Heli** screen.
4. Now ensure the aileron (roll left/right) works the correct way. Centre the collective (swash middle) and roll to the right. If the swash tilts right good. If not, change the lateral cyc. direction to **INV** in the swash menu.
5. Next check the pitch. If up goes down, change the Coll. pitch direction to **INV**.
6. The swash ring setting on the screen is to ensure the servos are not sent beyond the limits and can be used to set a max swash tilt. This also has the effect of reducing the agility of the helicopter (as the swash cannot tilt as far).
7. Now that all the servos go in the correct direction you can use PPM centre on the 3 cyclic channels in the Servos screen to get the servo horns dead middle without limiting the end points. (Zero out cyc1, cyc2 and cyc3 temporarily to assist with this.)
8. Mechanically adjust the swash height to mid-travel by altering servo/swash pushrod lengths.

9. Mechanically adjust the mid-collective pitch by altering swash/grip pushrod lengths.

10. Use the **Servo** page to adjust min and max servo values for cyclic servos to obtain desired min and max collective pitch (remove the zeroing out of **cyc1**, **cyc2** and **cyc3** first).

11. Test the cyclic pitch is as expected at min, mid and max collective pitch. Adjust by changing the collective weight and the cyclic weights.

12. Finally check the head speed to see if it is within the manufacturer's suggested range. This can be done with a hand-held RPM sensor, but far more elegant is to use the FrSky RPM and temperature sensor, and then log the sensor during the first few flights. Remember, the aim is to keep the rotor speed as constant as possible in normal mode. The motor speed can be adjusted using the appropriate curve, developing it from the simple starting curve given earlier. If the speed is much too high or low to adjust using curves, then consider changing the gearing, or in extreme cases, using a battery with a higher or lower cell count.



13. On the subject of monitoring, the cell voltage and current consumption can also be monitored to check both current usage and to find the reasonable flying time.

How to Create a Kill Switch

It is often useful to have a switch to disable the electric motor on a model. Simple disable switches are readily created, but are not foolproof. The aim here is to have a kill switch that is *almost* foolproof. The system starts off disabled, requires a switch (**SE**) to enable, and crucially cannot be accidently disabled whilst in flight. However, this last requirement does mean that an audible warning must to be used to notify whether the system is armed or not. Switch **SE** in the mid position is used for the "on" state, and has to be moved to the **SE↓** position for more than 2 seconds but less than 5. Similarly **SE↓** is switched again for the same time to disarm. In both arm and disarm, the throttle has to be at -100.

#	Function	V1	V2		AND Switch
L1	a~x	[I1]Thr	-100		----
L2	Sticky	L3	L3		----
L3	Edge	SE↓	2.0	5.0	L1

The core of this kill switch is in the **Logical Switches**.

L1 tests that the throttle is off, and that switch E is not off. **L3** provides the function to test that **SE↓** has been moved for between 2 and 5 seconds, and **L2** uses a sticky to switch between the "on" state and the "off" state. **L2** is used as the control for the motor to be on or off.

This control is seen on the **Mixes** page where the throttle is set to minimum, and then **replaced** (**:=**) with an active throttle when switch **L2** becomes active.

```
CH01          MAX Weight(-100%)
           := [I1]Thr Weight(+100%) Switch(L2)
CH02          [I2]Ele Weight(+100%)
CH03          [I3]Rud Weight(+100%)
CH04          [I4]Ail Weight(+100%)
```

Finally, audible warnings are set up in the **Special Functions** screen.

#	Switch	Action	Parameters		
SF1	!L2	Play Track	thrdis	▶	No repeat
SF2	L2	Play Track	thract	▶	Played once, not during startup
SF3	SE↓	Play Sound	Beep 3		1s

SF3 creates a bleep every second as a guide as to how long to hold the switch down. **SF1** and **SF2** are very necessary, as the motor will not disarm until the throttle is at a minimum and the switch sequence has been completed. The switch position cannot be relied on as a guide as to whether the motor is enabled. The physical switch **SE** can be moved to any position but will have no effect without the other conditions being true. (The Amber voice pack is shown.)

As an alternative, some might prefer to use the momentary switch, **SH** and toggle that for at least 2 seconds, but no more than 5. Personally I think this version is more elegant, however as a mode 2 flier I much prefer to have the throttle cut on the same side as the throttle, and that single momentary action switch is very much in demand for other functions.

#	Function	V1	V2		AND Switch
L1	a~x	[I1]Thr	-100		----
L2	Sticky	L3	L3		----
L3	Edge	SH↓	2.0	5.0	L1

#	Switch	Action	Parameters		
SF1	!L2	Play Track	thrdis	▶	No repeat
SF2	L2	Play Track	thract	▶	Played once, not during startup
SF3	SH↓	Play Sound	Beep 3		1s

There is no change in the **Mixes** screen from the version show on the previous page:

```
CH01             MAX Weight(-100%)
              := [I1]Thr Weight(+100%) Switch(L2)
CH02             [I2]Ele Weight(+100%)
CH03             [I3]Rud Weight(+100%)
CH04             [I4]Ail Weight(+100%)
```

Note:

Although a good idea to still do so, it is not necessary to have a switch warning for the kill switch.