Section 6, Telemetry

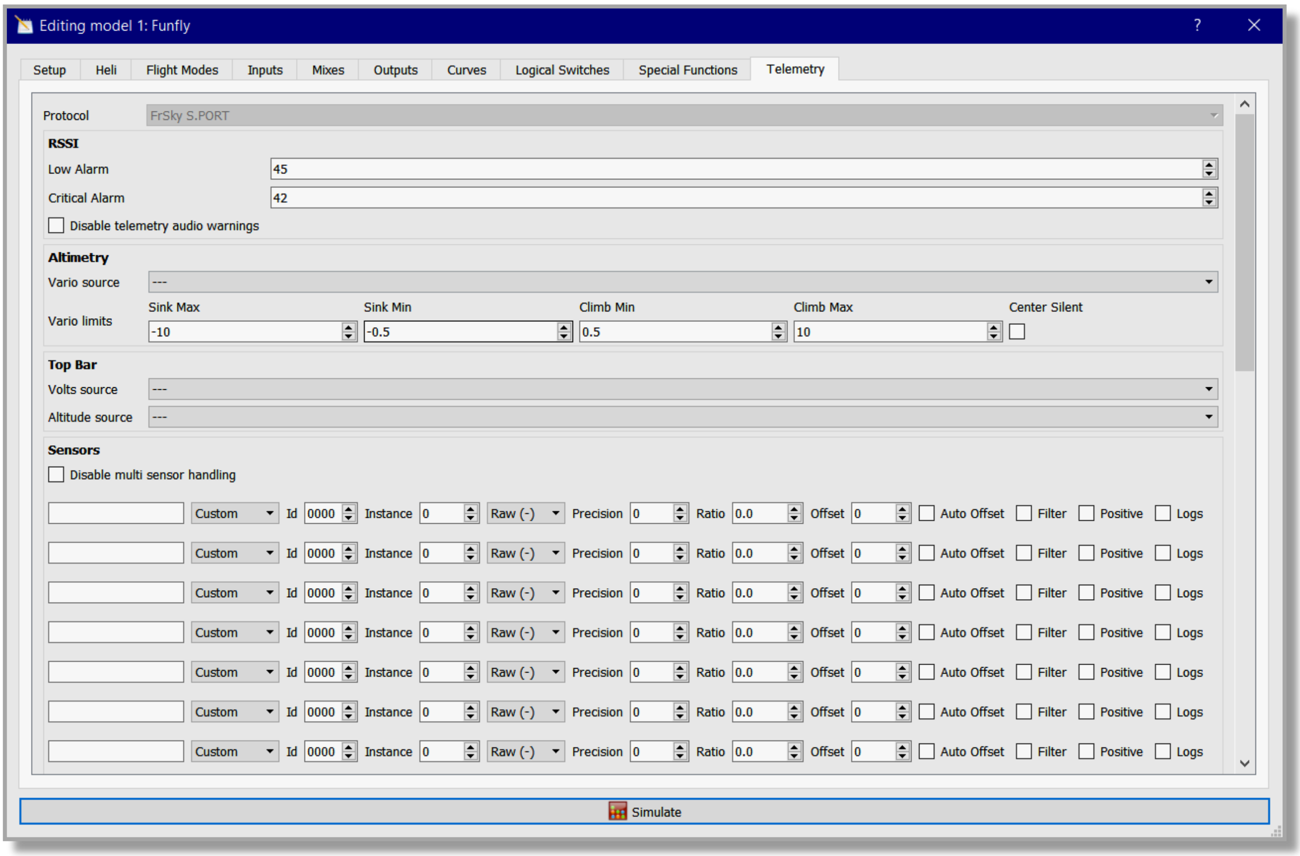
Contents

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

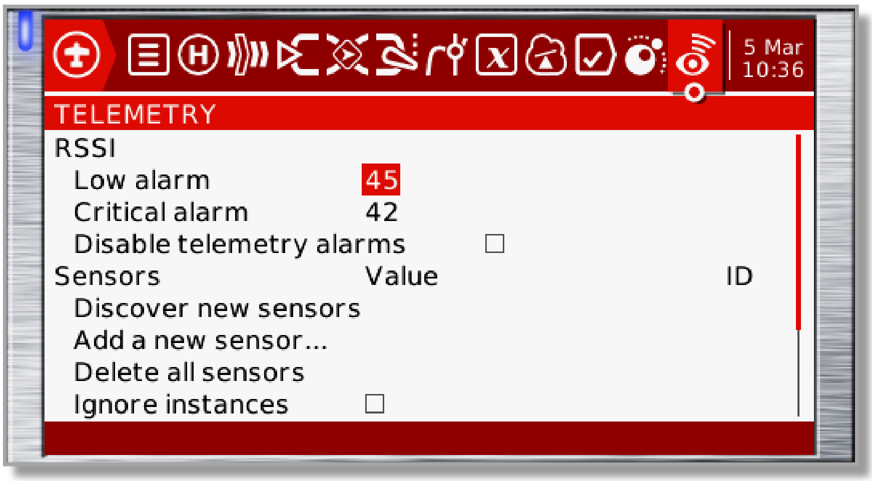
The Telemetry Screen

The **Telemetry Screen** opens a window onto a very comprehensive telemetry system. Together with the available FrSky Smart Port sensors, and those manufactured by third parties, the use of telemetry can considerably enhance the flying experience. For brevity, this manual will only consider the use of telemetry with the FrSky Smart Port sensors. Also it will only relate to the current range of telemetry enabled receivers available, i.e. the “X” series. Every sensor can be used everywhere for voice announcements, in logical switches, in inputs for proportional actions, and can be displayed on custom telemetry screens. One big advantage is that values can be seen directly on the telemetry setup page, so no need to add one to a custom screen just to have a look at it once. New data reception is visualised, and loss of a sensor is automatically detected.

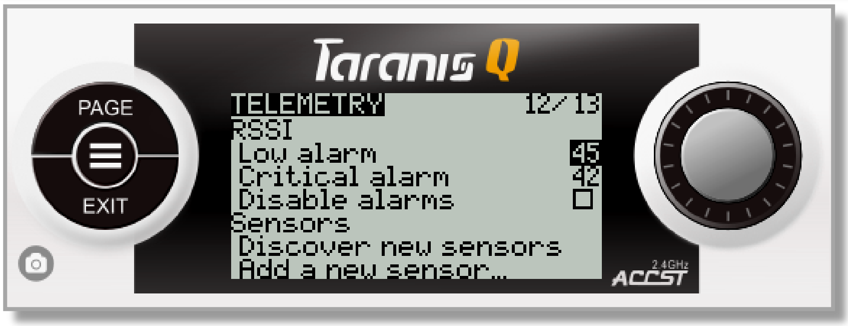
It is important to auto-identify the sensors early on in the programming process, otherwise they cannot be accessed by other screens.



Horus **Model Edit** telemetry screen

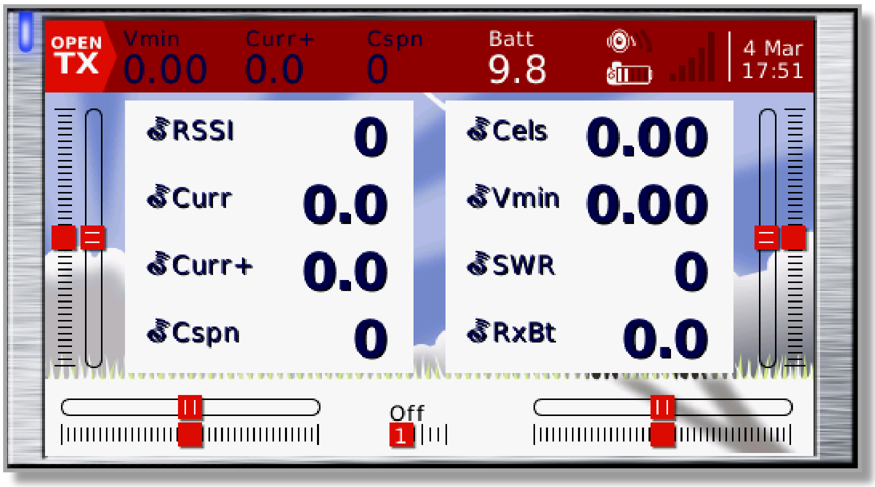


Taranis Q X7 **Model Edit** telemetry screen



Taranis X9D **Model Edit** telemetry screen

Sample Horus telemetry screen



Telemetry Key Points

- ✪ FrSky Sport sensors can be discovered anytime the radio and receiver/sensors are powered up. This, obviously, cannot be done using the **Companion**.
- ✪ Each value sent from the receiver is treated as a separate "sensor" with its own properties (unit, decimal precision, ratio/offset) and options (auto offset, filtering, persistent storage at power off, logging enabled). Each sensor has its own user-defined name and keeps track of its own min/max value.
- ✪ Sensors can be duplicated within **OpenTX**, so for example a given value e.g. altitude from the same vario sensor can be displayed/announced/logged simultaneously in different units, or with different options (absolute altitude and altitude above start point with auto or manual offset,...)
- ✪ Multiple physical sensors returning the same value are supported as long as there is a way to differentiate them. Each FrSky sensor has its own default ID set. Therefore two identical sensors cannot be connected to one receiver without changing the ID of one of them. This can be done with the FrSky SBUS servo channel changer (SCC) as required so each ID is unique in the smart port chain. (The software for this can be downloaded from FrSky website.) Thus individual motor currents of an octocopter with 8 FCS-40A sensors can be handled, or individual cells of a 12S battery can be monitored by using two voltage sensors.
- ✪ "Calculated" virtual sensors can be manually created to combine values or extract extra data. Values can be added, averaged or multiplied, the minimum or maximum of a set of up to 4 values van be extracted. This also takes care of "special cases" like calculating GPS distance (2D or 3D), getting the value of a particular cell of a lipo cell sensor, calculating mAh consumption etc. For example **Power** can be calculated easily by multiplying the related voltage and current, total voltages of multiple lipo cell sensors can be added to get the total voltage of series-wired packs, the minimum cell of each of them can be extracted and the lowest of all can be found using the **Minimum** function.
- ✪ Each sensor can be reset individually with a special function, so no more losing all your min/max values when you just want to reset altitude offset to start point.
- ✪ The telemetry also logs all the switch and joystick positions from the transmitter if logging is enabled.
- ✪ Each sensor has two S. Port connectors so a number of sensors can be daisy-chained together.

The FrSky Smart Port Sensors

FAS-40S, FrSky 40A Ampere Sensor-for S. Port

Curr	Custom	Id	0200	Instance	3	A	Precision	1	Ratio	0.0	Offset	0.0
VFAS	Custom	Id	0210	Instance	3	V	Precision	2	Ratio	0.0	Offset	0.00

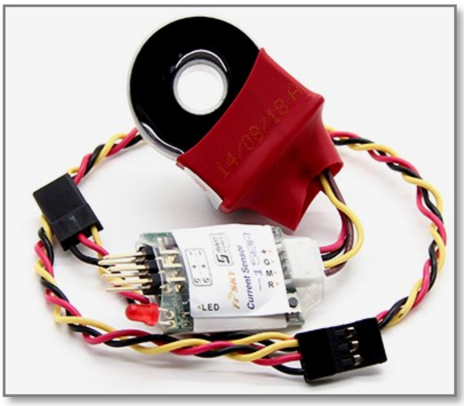
This sensor is supplied with XT60 connectors to plug directly between the battery and the model's power lead. It will measure currents up to 40amps. This sensor uses an in-line resistor and measures the voltage drop across the resistor. If using near its current limit, check for temperature rises in the device.

This sensor will also measure the lipo pack voltage, **VFAS** but not individual cells.

FCS-150A, FrSky 150A Current Sensor for S. Port

Curr	Custom	Id	0200	Instance	8	A	Precision	1	Ratio	0.0	Offset	0.0

The FrSky FCS-150A is a current measuring sensor for applications drawing up to 150 amps. Simple setup of the FCS-150A means the ESC battery lead is passed through the sensor loop noting the current direction indicator on the side of the sensor. No reading will be obtained if the direction is incorrect.



Notice this sensor has the same **Id**, but a different **Instance** to the FAS-40S. It does not have a lipo voltage measuring capability.

FLVSS, FrSky Lipo voltage sensor for S. PORT

Cels	Custom	Id	0300	Instance	2

This sensor is used to monitor the voltages of LiPo packs. It incorporates a colourful easy-to-read (even in the dark!) OLED display showing both total voltage and individual cell voltages. Simply plug in to the lipo balance plug to view the voltage readings. The sensor can also be used as a handy, pocket-size, stand-alone lipo checker. As this sensor reads each individual lipo cell voltage, it can be configured to give a range of information about the state of the battery, and can be used with the **Logical Switches** and **Special Functions** to give battery warnings. A new version of this sensor, the "Mini" is available without the OLED display.

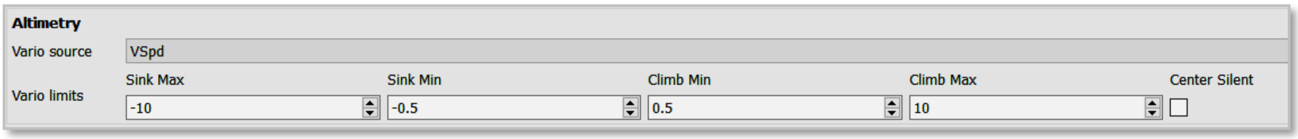
FVAS-02H, FrSKY Variometer Sensor for S. Port

VSpd	Custom	Id	0110	Instance	1	m/s	Precision	1	Ratio	0.0	Offset	0.0	☐ Auto Offset
Alt	Custom	Id	0100	Instance	1	ft	Precision	1	Ratio	0.0	Offset	0.0	☑ Auto Offset

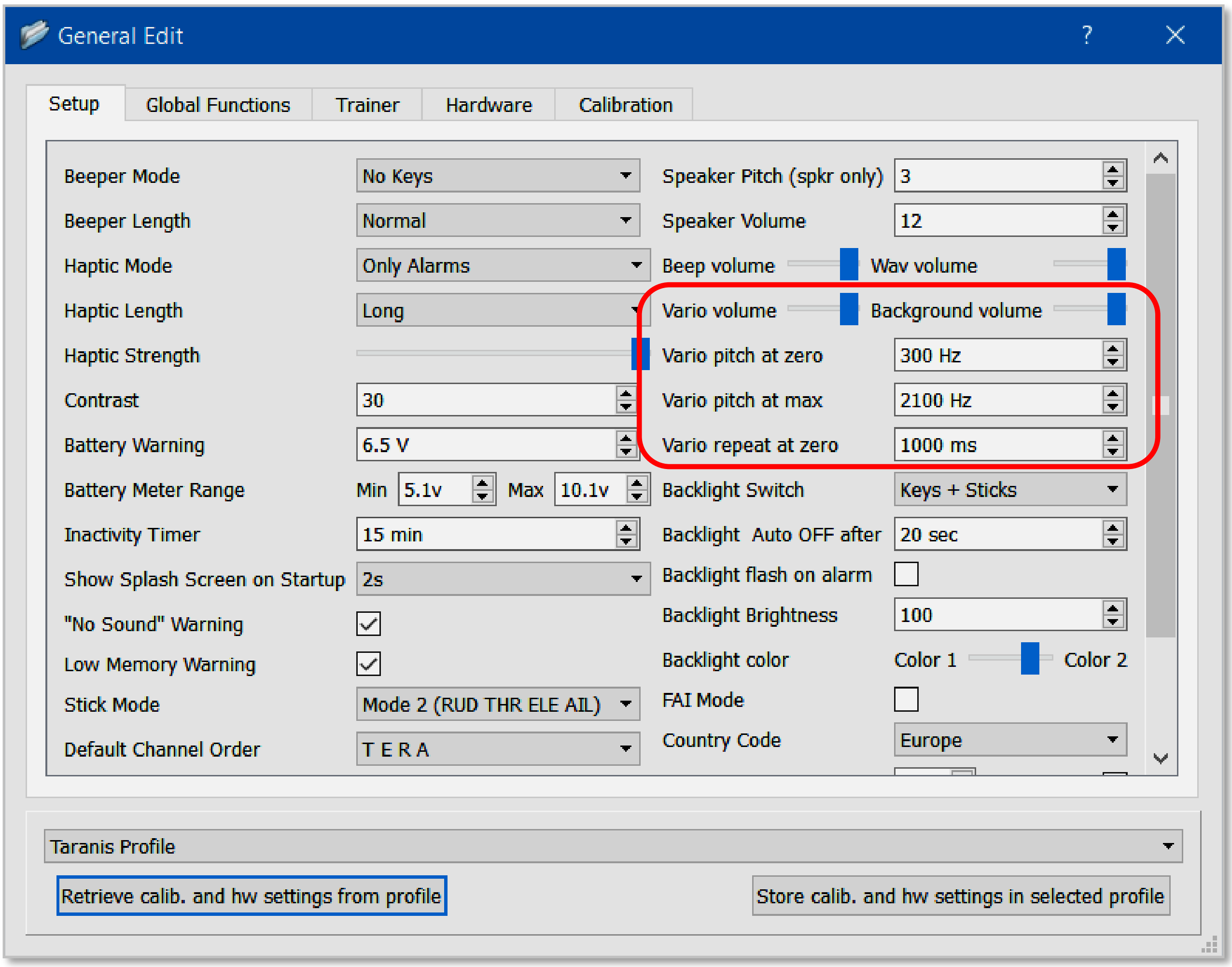
This sensor comes in two variants, normal precision and high precision. It creates two entries in the telemetry, **Vspd**, vertical speed, and **Alt**, altitude. It measures the rate of change of altitude by detecting the change in air pressure (static pressure) as altitude changes. It offers true glider vario features with tones that indicate lift and sink. As with most electronic barometric sensors, strong airflow and also pressure changes in weather will drastically decrease accuracy. The last letter of the part number designates the precision, "H", or high has a precision of 0.1m, "N" or normal has a precision of 1m.

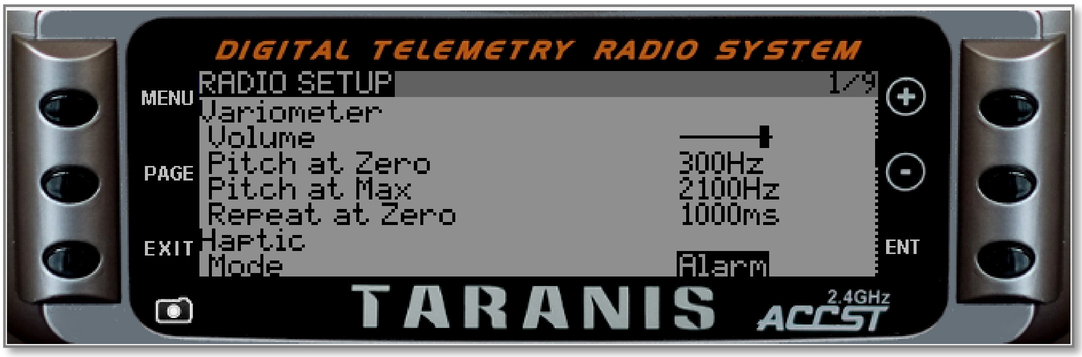
The vario offers two forms of telemetry. The first is a direct readout of height in one's chosen units. The second is a rising or falling tone produced by the transmitter to indicate whether the plane is rising or falling.

Setting up the Vario



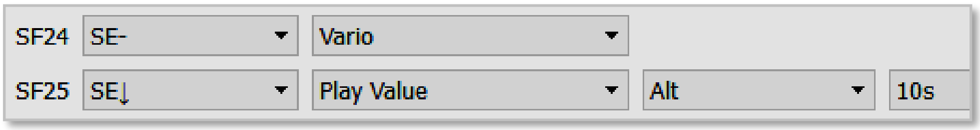
On the telemetry screen is the Altimetry box. First the **Vspd** must be entered as the **Vario source**. Then the limits can be set. They can be adjusted within the range ±17. Also the minimum points can be altered. These settings provide the rise and fall tone. Also there are some vario settings in the **General Settings Menu** on the **Companion**, or the **Radio Edit** on the transmitter:



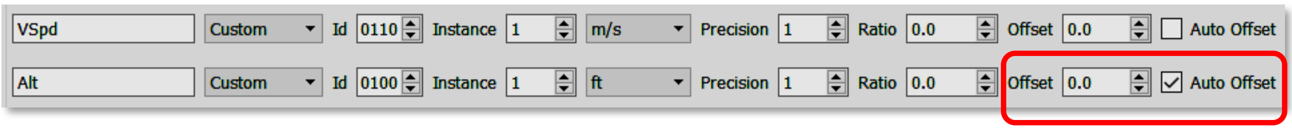


All these settings for the vario are down to personal preference and can be found by trial and error.

Obviously how one sets up the vario is up to the user, but one common method is to use a three position switch, and have the centre position for the vario and program the forward position to read out the height. Having switch SE off will disable both the vario and the height telemetry.



This is easily accomplished using a couple of special functions as shown above using switch **E**. All that is needed to program the vario is to set **SE-** to **Vario**. To give the spoken telemetry height simply select **Play Value** and select **Alt**. The setting in the right hand box is the delay between each reading. This can be set between 1 second and 60 seconds. One second gets very irritating, ten seconds seems to be a useful delay between readings, but as with so much in OpenTX this is down to the user's personal preference.



The **Alt** telemetry, by default, is set up to **Auto Offset** the height. This sets the height to zero at the place where the receiver is powered up. Thus one gets a height relative from ones flying position. If the absolute height is required, uncheck the **Auto Offset** box.

While the Vario might be seen as just the province of glider pilots, it can have other uses too. For the electric powered glider flying off a normal flying field, it opens up more precision in thermalling, or maybe this brings a whole new challenge instead of beating up the field seeing how fast it will go. It could be used with beginners in trainers to learn how to turn and maintain height, or even with more advanced flyers wanting to hone their skills.

GPS-02, FrSKY GPS Sensor for S. Port

GSpd	Custom	Id	0830	Instance	4	kt	Precision	1	Ratio	0.0	Offset	0.0			
GPS	Custom	Id	0800	Instance	4										
GAlt	Custom	Id	0820	Instance	4	ft	Precision	1	Ratio	0.0	Offset	0.0			
Date	Custom	Id	0850	Instance	4										

This particular sensor uses GPS to feed variable directional information.

Gspd Ground speed

GPS GPS co-ordinates in format set in **General Settings**

Galt Altitude

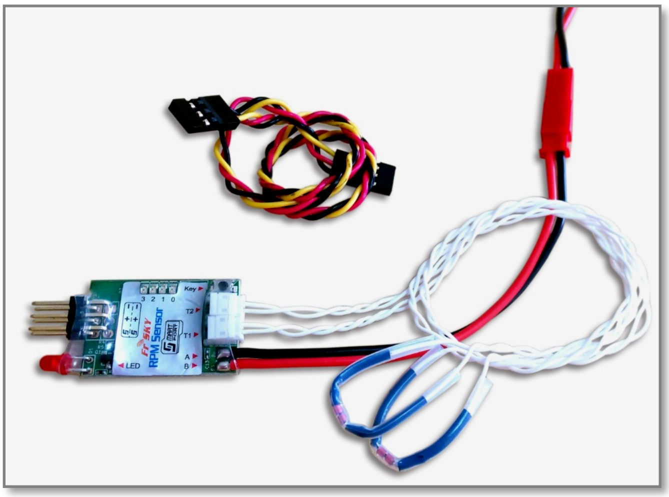
Date Taken from GPS data. Universal time co-ordinated (UTC).

GPS co-ordinate format can be altered in the **General Settings** screen. The GPS sensor does require a few minutes to lock on to the GPS satellites.

SP-RPM, FrSky RPM and Dual Temperature Sensor for S. Port

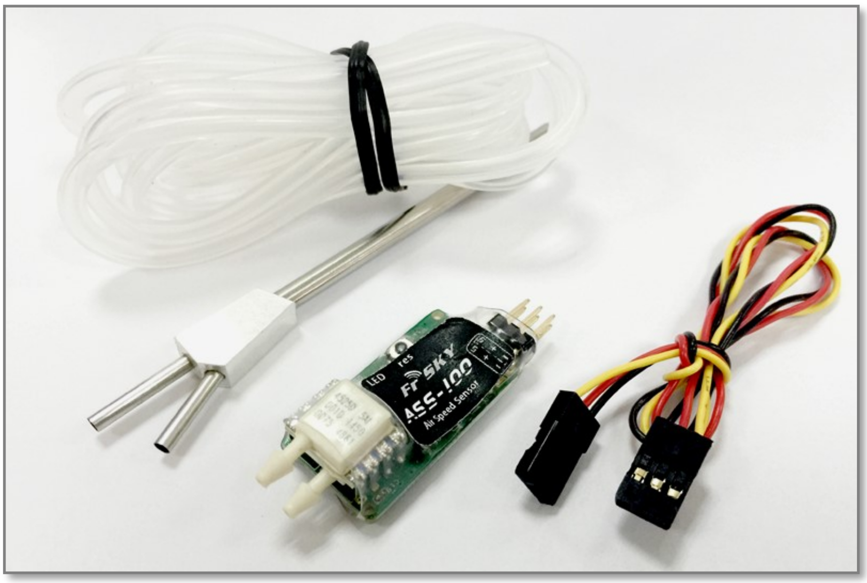
Tmp2	Custom	Id	0410	Instance	5	°C	Precision	0	Ratio	0.0	Offset	0	☐	Auto Offset
RPM	Custom	Id	0500	Instance	5	RPM	Precision	0	Blades	1	Multiplier	1		
Tmp1	Custom	Id	0400	Instance	5	°C	Precision	0	Ratio	0.0	Offset	0	☐	Auto Offset

This sensor can work with different kinds of brushless motors and most speed control systems in market. Two temperature sensors are included with this product. These sensors can used to read temperatures of model accessories, and ambient air temperature.



ASS-70/ASS-100 Frsky Air Speed Sensor for S. Port

This sensor comes in two versions, the ASS-70 has a measurement range up to 270km/h, whereas the ASS-100 will read up to 360km/h. It comes complete with a pitot tube and tubing.

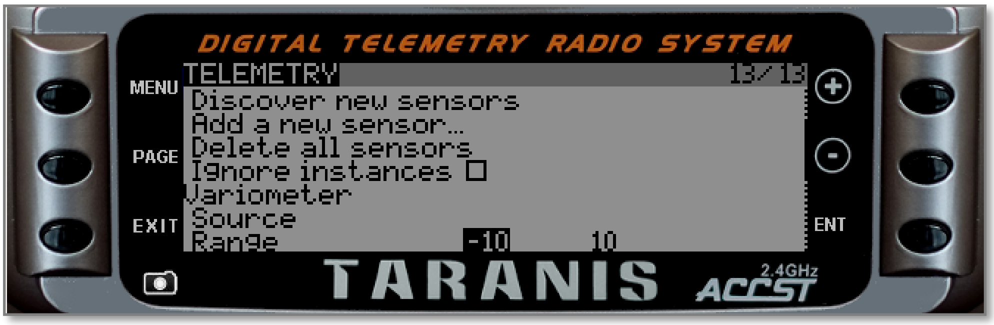


Discovering the Smart Port Sensors

Remember this discovery should be done at an early stage of the model programming to ensure that the sensors are visible in the other screens.

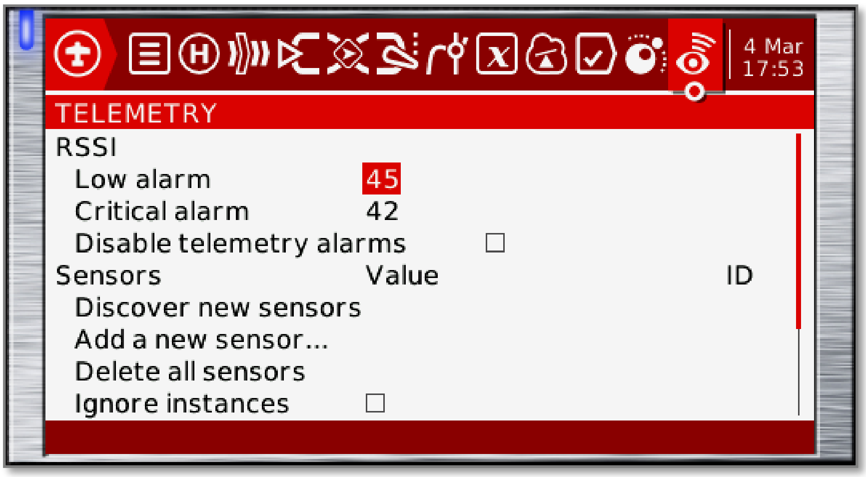
Discovering new sensors must be added from the transmitter. They cannot be discovered using the **Companion**. However, once they have been discovered, the model file can be uploaded to the computer when they will be then accessible through the **Companion**.

To discover new sensors, one needs the receiver, with sensors connected, powered up and bound to the transmitter set to the appropriate model. Go to the telemetry page and scroll down to **Discover new sensors** and press **ENT**. This will find any sensors. Then select **Stop discovery**. Remember to keep the transmitter at least 1m away from the receiver to ensure telemetry signals are received. You can also add and delete sensors from here. Although sensors will appear, the values might not appear straight away. GPS data takes several minutes to start coming through. However, this does not matter for the discovery process, and one does not need to wait for the data to appear. Once the sensors have been detected, the receiver can be powered down, and programming can continue.

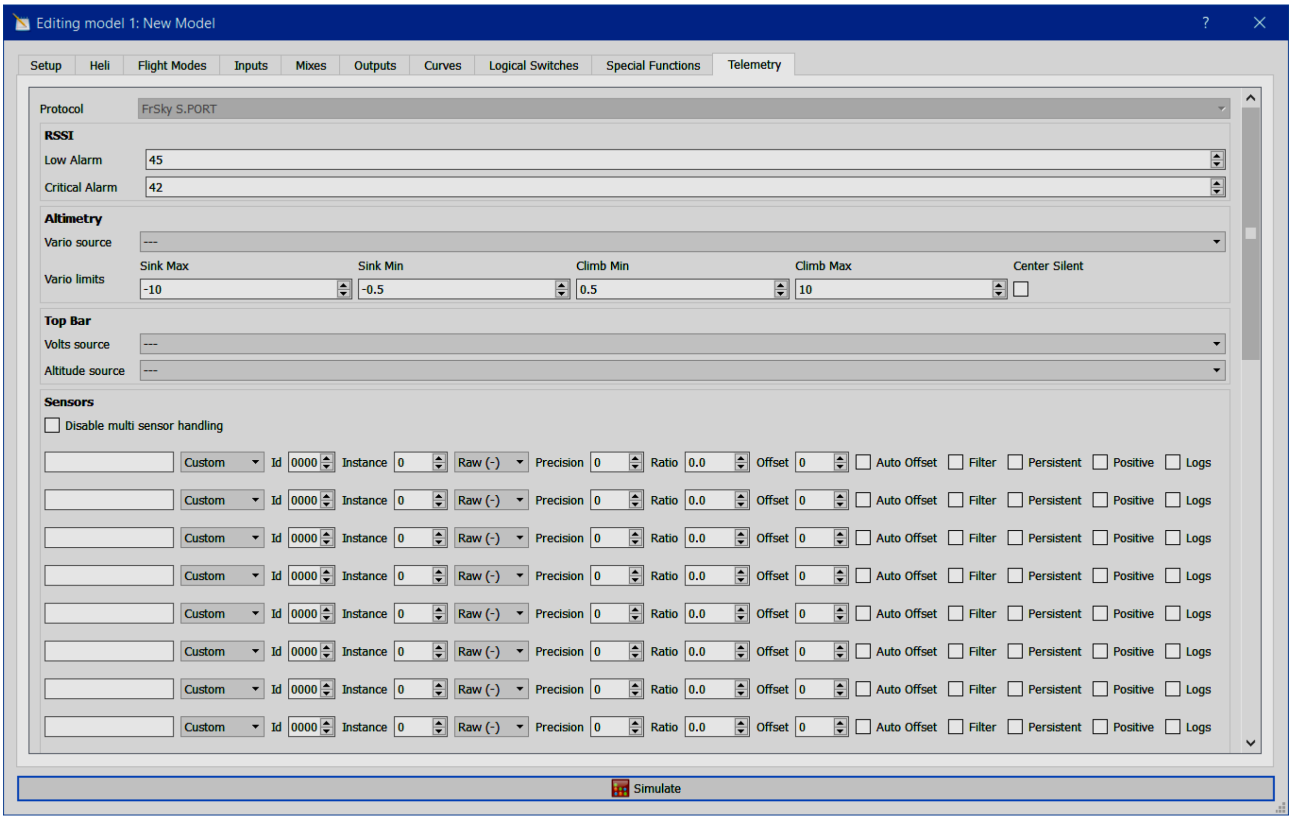


Add a new sensor allows the used to add a calculated sensor. This is covered in more detail further on in this section.

Ignore Instances is the same as **Disable multi sensor handling** on the **Companion** screen. This should not normally be ticked. It is to stop third party sensors that flood the sensor screen filling all the slots.



The Telemetry Screen



Protocol

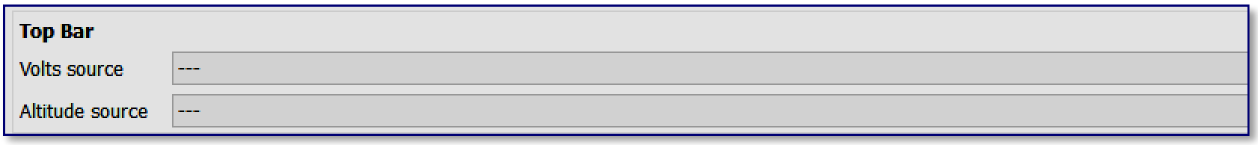
In this screenshot, the Protocol box is currently greyed out, and cannot be changed from FrSky S Port as the transmitter is set to a FrSky radio. Other radios will allow different protocols to be used. These appear in a drop-down box.

RSSI

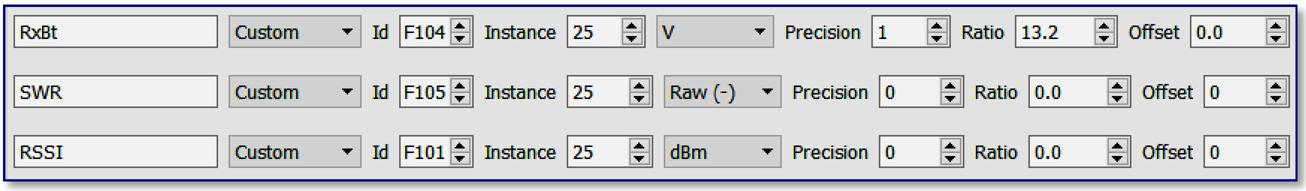
RSSI stands for Received Signal Strength Indicator. This is a value transmitted by the receiver in the model to a receiver in the transmitter that indicates how strong the signal is that's being received by the model. When it drops below a minimum value, this indicates that one is in danger of flying out of range. RSSI warnings are automatically built into the OpenTX software, and are triggered by the two settings automatically set in the telemetry window. These can be changed by the user, but the default values set, 45 and 42 should be reduced with extreme caution. However, increasing the values slightly would give a greater safety margin. It should be noted this is a built in telemetry function of the "X" series receivers and requires no additional sensor hardware.

Altimetry

This option requires one of the vario sensors to be connected and discovered first.



These options refer to the top bar of the telemetry screen where you are able to show a voltage source and an altitude source, depending on what sensors have already been discovered.



Without any sensor connected, the above three “sensors” are always detected as they are built into each “X” series receiver.

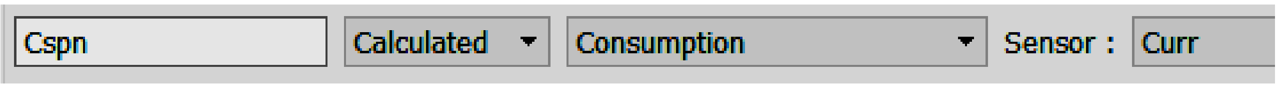
RxBt	The voltage of the receiver battery.
SWR	SWR stands for "Standing Wave Ratio" This is a measure of transmitter antenna quality. Reading below 51 is normal. Note that SWR does not work on some earlier Taranis and Taranis Plus radios.
RSSI	See earlier notes about this.
Disable multi sensor handling	This should not normally be ticked. It is to stop third party sensors that flood the sensor screen filling all the slots.
Units	This drop down box offers a range of units. Conversion is carried out automatically if the unit is acceptable alternative for the default unit. i.e. **mph** can be changed to **km/h**, but not **volts**, although the units shown appear to have changed.
Precision	The number of decimal places required.
Ratio	This is to adjust sensor outputs to ensure to give a realistic range.
Offset	Offset adds a constant to the value.
Auto Offset	Stores the first received value after a reset as zero.
Persistent	The value gets stored at power off and recalled next time. Can be useful for mAh or fuel totalizer, for example.
Positive	Gives only positive values.
Logs	Value will be set to the data log if ticked.

Calculated Telemetry

OpenTX not only gives the ability to see and use telemetry values transmitted directly from the receiver, but also to calculate new telemetry fields. Up to 32 physical and virtual sensors can be used for each model. When used in other **OpenTX** screens, most of the sensor values can have three options, the actual sensor value, the minimum sensor value (the sensor name denoted by a - sign in front) and the maximum sensor value (the sensor name denoted by a + sign in front). Thus if a current sensor is fitted, one can see the maximum current drawn during a flight. Each new virtual sensor can be given a name of up to 4 letters, and **OpenTX** differentiates between capital letters and lower case. Thus it is possible to have two virtual sensors names such as **Batt** and **batt**. If, during editing, a calculated sensor name is changed, other screens using that name will also change automatically. This does not normally happen with **OpenTX**.

Examples of Calculated Telemetry

Flight Battery Consumption



This requires the current sensor **Curr**, and the field is calculated. A very useful derived value for giving warnings of flight battery consumption.

Battery Power, Electric Flight

Here the current is multiplied by the voltage (using the voltage sensor built into the 40amp current sensor) to give the power in watts. Useful for seeing the maximum power drawn during a flight, and together with the logs can see how long this power was maintained, as it will reduce as the battery voltage reduces.

Lowest Lipo Cell Voltage Reading (using the voltage FLVSS sensor)

This gives the voltage reading for the cell with the lowest voltage. Again this is very useful as an early warning of battery failure, depending on the voltage levels set in the **Logical Switches** section. Typically, if set appropriately, this can alert of a failing battery when full power is applied. Obviously the speed controller should be programmed to reduce voltage at a lower value than is set in the **Logical Switches.**

Individual Lipo Cell data (using the voltage FLVSS sensor)

Vol1	Calculated	Cell	Cells Sensor	Cels	Cell 1
Vol2	Calculated	Cell	Cells Sensor	Cels	Cell 2
Vol3	Calculated	Cell	Cells Sensor	Cels	Cell 3
VolD	Calculated	Cell	Cells Sensor	Cels	Delta

Here **Vol1**, **Vol2** and **Vol3** are the individual cell voltages. **VolD** gives the delta value (the difference in voltage between the highest cell and the lowest) for the battery. Note only three cells have been shown here, if it is a 4S battery, **VolD** will give the delta value between the four cells in the battery.

Measuring Distance using the GPS

Dist	Calculated	Dist	GPS Sensor :	GPS	Alt. Sensor :	---	m

The calculated sensor is created using the formula **Dist** (Distance) with the **GPS** as the source. This will give a 2D distance.

Dist	Calculated	Dist	GPS Sensor :	GPS	Alt. Sensor :	GAlt	m

If a 3D distance is required, first edit the **Altitude** field to turn on **Auto Offset**, then include the **Galt** source.

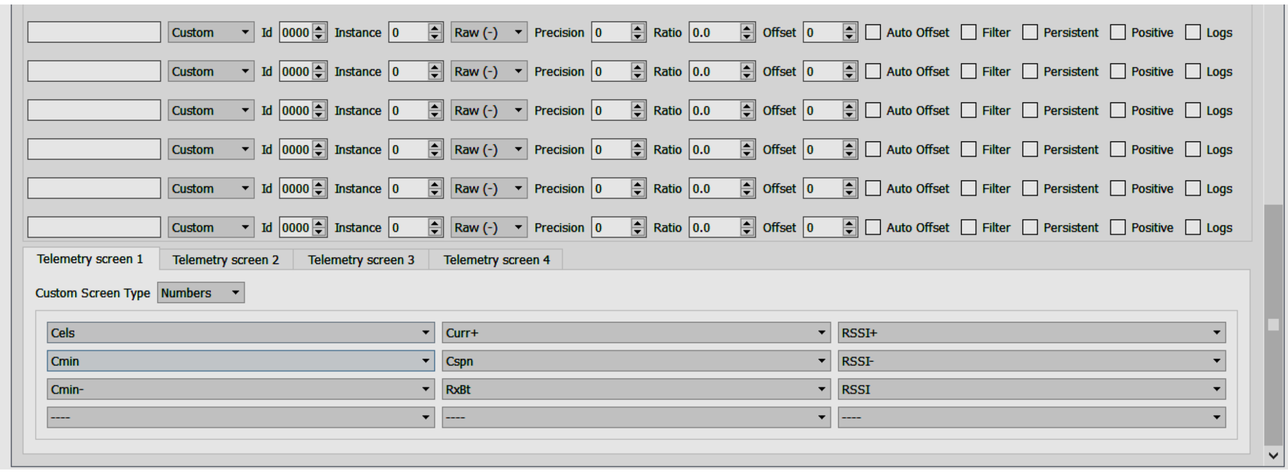
Analogue Inputs A1/A2 on “X” Series Receivers

The analogue inputs on the “X” series receivers are legacy (ie. Historic). The only use they are put to now is to feed back as telemetry the voltage of the receiver. Essentially this is reported back as simply a number in the range 0-255. The ratio parameter can the be used to turn them into something useful to display. Hence the discovered sensor RxBt (the receiver battery voltage) which is actally A1 has the ratio 13.2

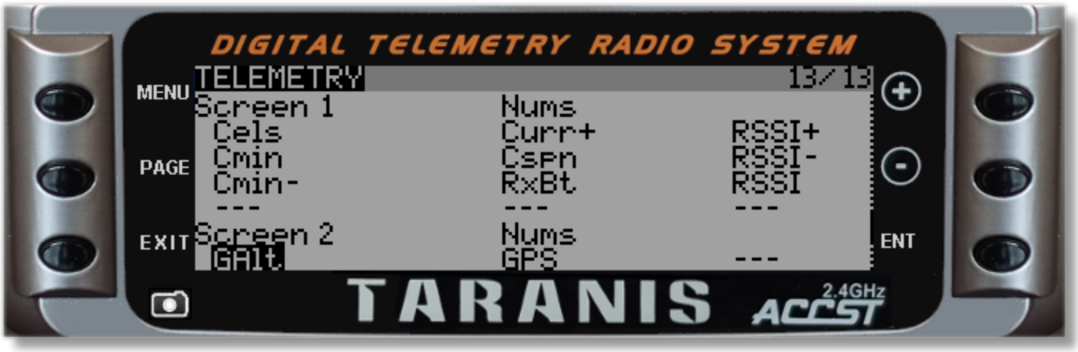
The X4 receivers, unlike the rest, do have the other analogue input, A2, which can be used for telemetry purposes. Again they do need a ratio parameter putting in. See Section 10, How-To Part 2 for an example of how this is used.

Setting up Telemetry Displays (Taranis)

The Horus and Taranis (Q X7, X9D etc.) are different here. On the Horus, the telemetry screens are set up on the transmitter as widgets. See the Horus radio section for details of adding telemetry screens. This section deals with telemetry displays only on the Taranis. Up to 4 telemetry screens can be set up on the radio. Each screen can use numbers, bars or use Lua scripts to display the data. Setting up a telemetry screen can be found at the bottom of the main **OpenTX Companion** telemetry page for Taranis radios.



First select the telemetry screen, then the custom screen type, then any of a whole range of parameters can be entered into the screens. This is probably the last operation to set up once a model has been programmed, as it can contain a range of parameters including logical switches, and telemetry values.



These screens are really very much down to individual preference, though for electric flight users, having a screen record of such things as maximum current drawn, pack voltages and maximum power can be very helpful.

Data Logging

Data is constantly being sent back to the transmitter from the receiver and any sensors connected to it. This data can be stored on the transmitters SD card, and downloaded to a computer using a USB cable and Bootloader mode on the transmitter, or by removing the SD Card and plugging that into a computer.

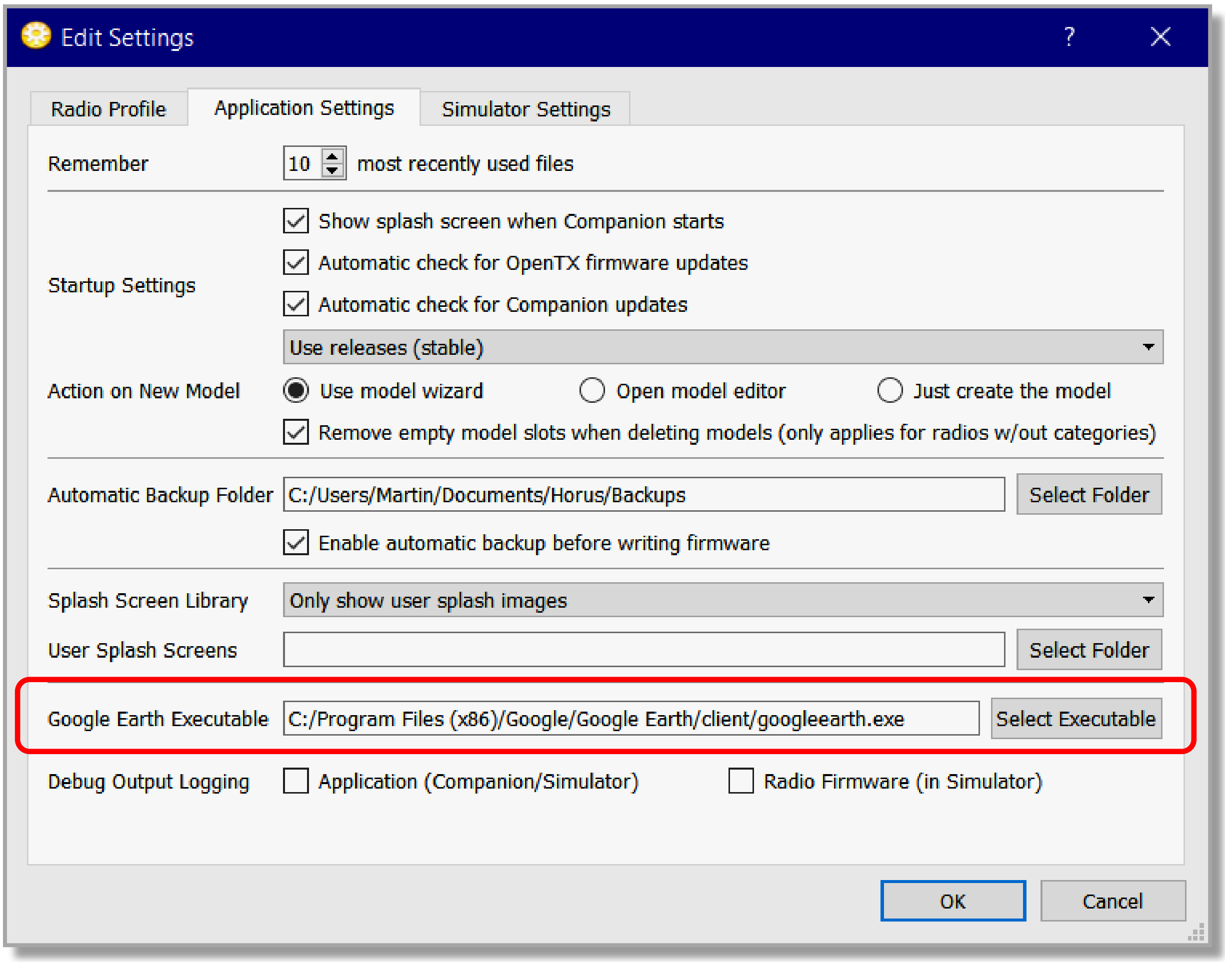
To enable data logging:

1. Ensure that the appropriate **Logs** box is ticked on the **Telemetry Screen**. It is easy to tick every box, but some data does not need to be logged. E.g. with the GPS sensor, it will report back on the date. Logging this field is mostly unnecessary, especially as the saved log file always included the date and time of the flight.

2. Several **Special Functions** need to be set up to enable logging. There are various ways of doing this:

This method simply uses switch F to start and stop logging. The parameter in **SF11** sets the frequency of logging, in this case every 2 seconds. **SF12** and **SF13** simply give verbal warnings of the data logging state. On a 10 minute flight, that still amounts to 600 lines of data!
Other methods could include using a **Logical Switch** to determine when the countdown clock starts for instance, or using the speed parameter if a GPS or speed sensor is fitted, and enabling logging when the speed is greater than, say, 5mph.

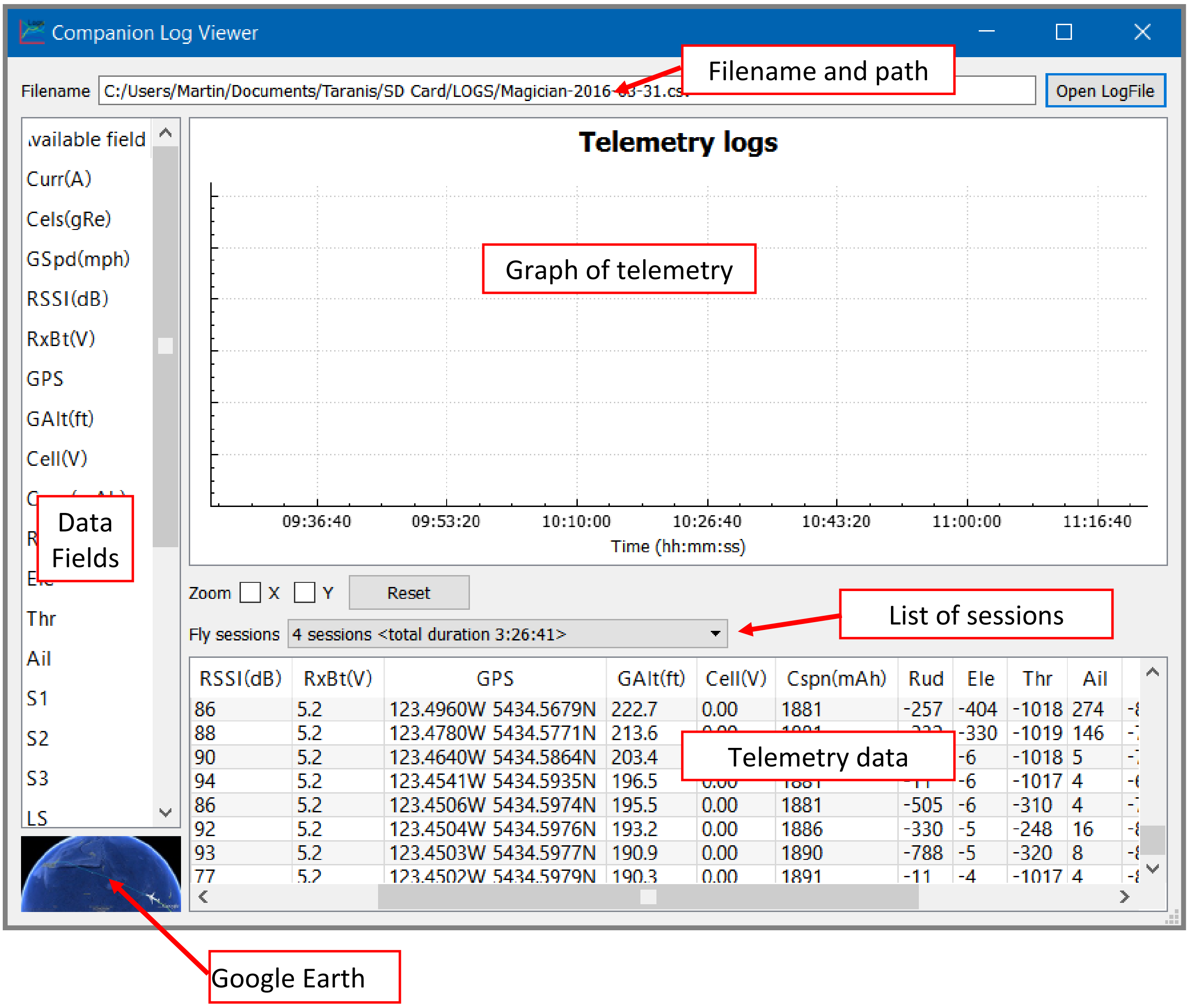
3. To use Google Earth with the GPS sensor, then it is necessary first load Google Earth onto your computer, and then enter the correct path where Google Earth is stored from the **Edit Settings**, **Application** menu of the **Companion**.



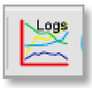
Notes:

The log files stored on the SD card will have the filename of the model, and the date recorded. Each log file may contain a number of logging instances undertaken during that day. The log file is also readable in Excel or similar spreadsheets.

It should be noted that the GPS sensor requires a couple of minutes to activate, this could be important if the power is switched on just before a flight as would be the case with an electric model with an arming plug.



This **Telemetry Log Screen** is accessed from the **OpenTX Companion**, either from the **File** menu or from the screen menu icon.

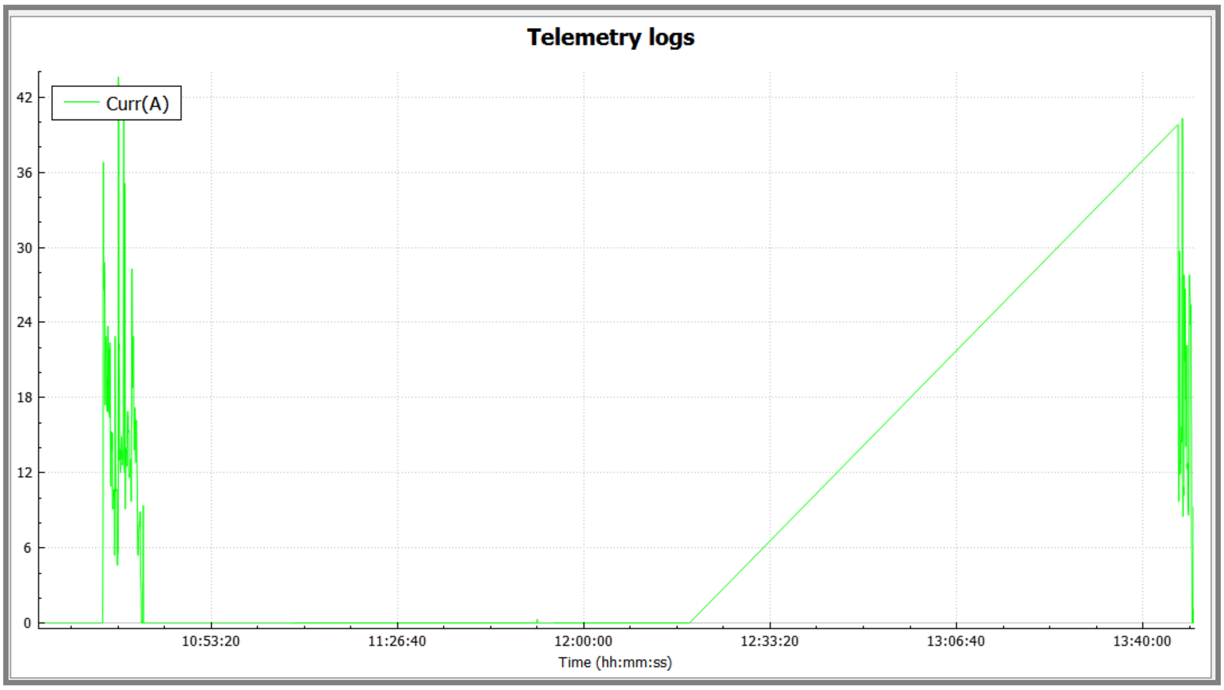


Notes

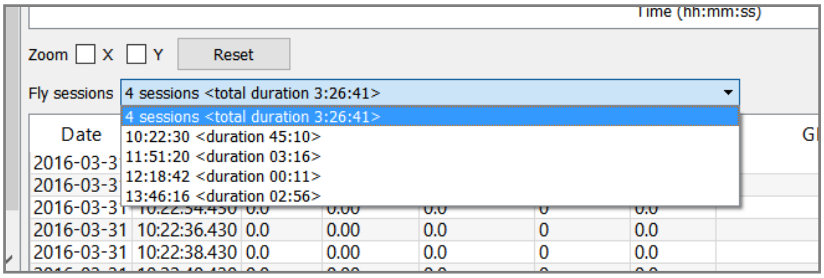
- The Google Earth icon in the corner of the screen will only appear if it has been correctly enabled in in the **Companion** settings.
- The data logging also logs the joystick, slider and switch positions on the transmitter. Even if no sensors are fitted it will also log the **RSSI** and receiver battery voltage. This latter ability is very useful with a new model to fully test that the receiver aerials are functioning well in all flight conditions.
- **OpenTX** 2.2 will not be able to transfer GPS data to Google Earth if that data was created on **OpenTX** versions before 2.1 due to the way the GPS data is stored.

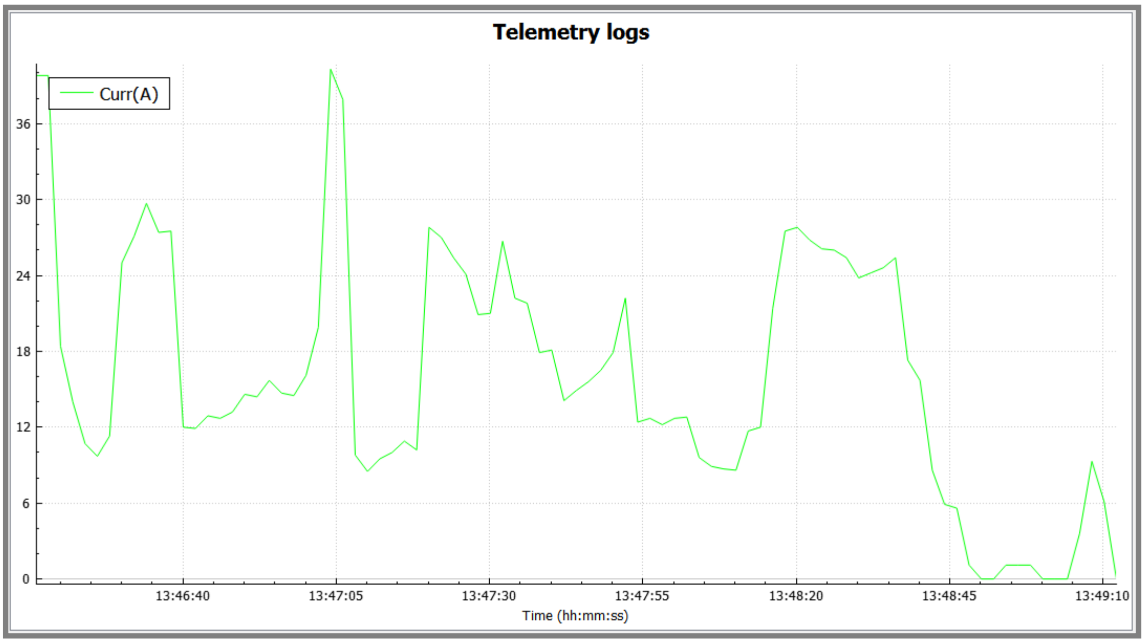
There are a number of points to note with the data logger.

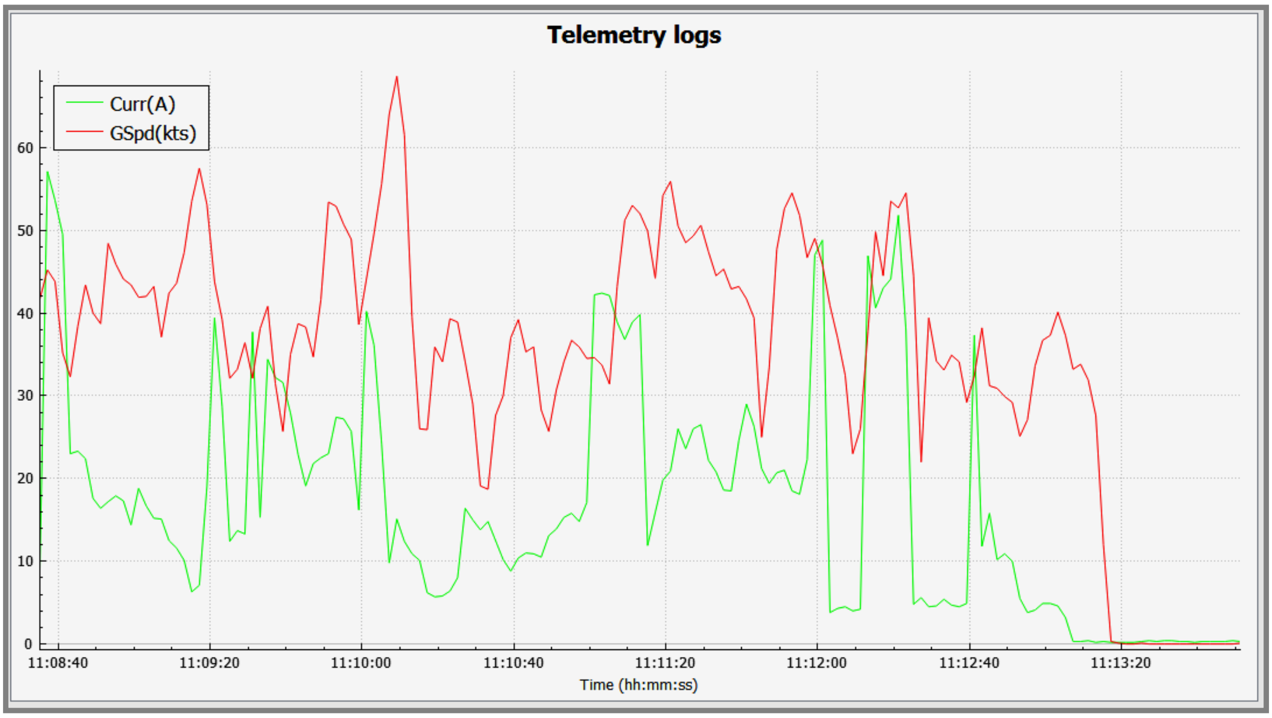
The logger will tend to log all the data. Then you will get a screen like this:



instead, load a single session instead of the whole file which lasted nearly 3 and a half hours.



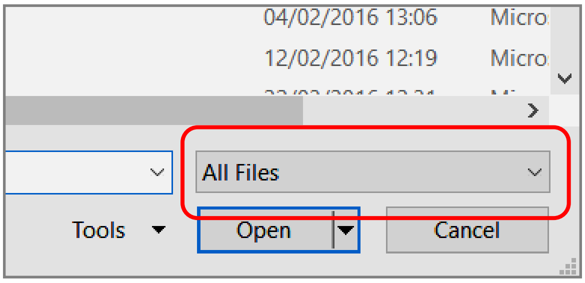


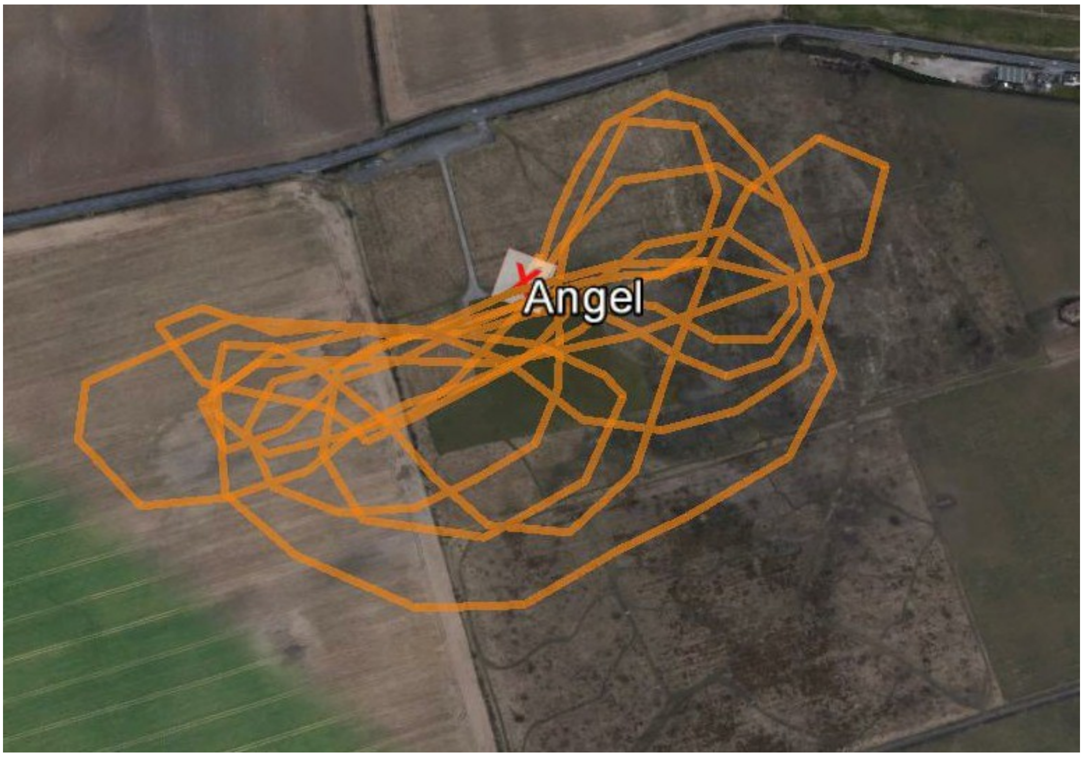


- Several data fields can be viewed together by holding down the CTRL key on the keyboard whilst selecting another data field.
- By highlighting just some of the lines of data it is possible to remove the parts of the log at the start and end of the flight where the plane is on the ground. The part in blue has been highlighted:

Date	Time	Curr(A)	RSSI(dB)	RxBt(V)	GSpd(kts)	Cels(gRe)	GPS	GAlt(ft)	Date	Cmin(V)	Cspn(mA
2016-04-10	11:13:45.280	0.3	93	5.6	0.0	19.21	123.4127W 5434.5969N	146.6	16-04-10 11:19:31	3.80	2226
2016-04-10	11:13:47.280	0.3	90	5.6	0.0	19.21	123.4127W 5434.5969N	147.6	16-04-10 11:19:31	3.80	2226
2016-04-10	11:13:49.280	0.4	81	5.6	0.0	19.21	123.4127W 5434.5969N	148.6	16-04-10 11:19:31	3.83	2226
2016-04-10	11:13:51.280	0.3	84	5.6	0.1	19.23	123.4128W 5434.5969N	149.6	16-04-10 11:19:31	3.82	2226
2016-04-10	11:13:53.280	0.3	82	5.6	0.1	19.26	123.4128W 5434.5969N	149.9	16-04-10 11:19:31	3.83	2226
2016-04-10	11:13:55.280	0.3	0	5.6	0.1	19.26	123.4128W 5434.5969N	149.9		3.83	2227
2016-04-10	11:13:57.290	0.3	0	5.6	0.1	19.26	123.4128W 5434.5969N	149.9		3.83	2227
2016-04-10	11:13:59.290	0.3	0	5.6	0.1	19.26	123.4128W 5434.5969N	149.9		3.83	2227

- If necessary, data logging files can be edited in Microsoft Excel. They are in CSV format and should be saved the same way. When loading, it is necessary to select **All Files** in the Excel file load window as shown opposite in order to see the logs.





Provided one has GPS logging data, clicking on the Google Earth button will automatically plot the flight. The name of the plane is shown, in this case Angel, and the flight path recorded. Ample proof for the club Safety Officer that I did not fly over the road with my plane! By altering the Google Earth perspective, different views can be obtained.