

openTx für Taranis Anleitung Deutsch

Bedienungsanleitungen

OpenTx V2.1x und Companion V2.1x

X9E X9D 9XR-Pro 9XR Th9x

Mit vielen Programmier-Beispielen



September 2015 Softwarestand r2940 bis openTx V2.12
Helmut Renz

Achtung, Halt, Aufpassen!

**Das ist keine normale Bedienungsanleitung.
Diese Anleitung beschreibt 3 Softwarestände**

Open9x bis r2940 mit Companion9x bis V1.52

OpenTx V2.00 bis V2.017 mit CompanionTx bis V2.017

OpenTx ab V2.10 mit CompanionTx V2.10 (V2.12)

Es müssen immer die beiden gleichen Softwarestände für openTX und CompanionTx verwendet werden, sonst passt das nicht zusammen.

Open9x: Ist vor allem auf Th9, 9XR, 9XR-Pro noch sehr verbreitet ist, viele haben noch nicht auf openTx V2.017 upgedated und arbeiten mit dem Zadig-Treiber. Sie brauchen die Infos wie sie auf openTX V2.017 updaten können.

OpenTx V2.017: Ist der aktuelle Stand ist, sehr ausgereift, praktisch fehlerfrei. OpenTx und CompanionTx sind weit verbreitet. X9D, 9XR-Pro, Th9, 9XR. Nicht für X9E.

OpenTx V2.10: Ist noch sehr neu, hat eine komplett andere Telemetrieverarbeitung, wird laufend angepasst und erweitert. Kleinere Fehler sind vor allem in CompanionTX noch vorhanden. Wird für X9E gebraucht um alle Funktionen verwenden zu können. (Stand August 2015) X9D, 9XR-Pro

Die Beispiele sind universell zu verwenden.

Jeder soll sich das rauspicken was für ihn interessant ist und zu seinem Sender und Softwarestand passt.

Fast alle Funktionen von openTx sind auf fast allen Sender verfügbar. Einschränkung treten durch den verbauten Prozessor oder Speicher auf.



Der Sender wird mit englischer Menüführung ausgeliefert.

Er kann auf Deutsch umgestellt werden. Dazu muss eine neue Firmware aufgespielt werden.

Das kann man mit dem Programm companion9x machen,

Der Sender ist mit einem Bootloader ausgestattet.

Die Sender-Software OpenTx ist open source und wird laufend erweitert und angepasst.

Inhaltsverzeichnis

Disclaimer.....	12
Haftungsausschluss.....	12
Das deutsche Handbuch besteht aus 7Teilen:.....	12
Dieses Handbuch passt auch für Th9x, 9XR und 9XR-Pro.....	12
Tip zum Ausdrucken:	12
Teil A Der Sender und seine Funktionen.....	13
Der FrSky Sender Taranis	13
Das Projekt Taranis	14
Die Funktionen in Stichworten, ein Überblick:.....	15
Vorstellung von Opentx und Taranis, um was geht es?	17
Blockdarstellung des Senders Frsky 9XD Taranis und X9E.....	18
Funktionsablauf im Sender	19
OpenTx Funktionen Erweiterungen	20
Hardware Modifikationen und Anpassungen:	20
Softwaremodule zusammenstellen:	20
Sender FrSky Taranis Bedienelemente.....	22
Ansicht Rückseite mit Schacht für externes HF Modul im JR-Format	23
Taranis Pinbelegung externer Schacht, Maße im JR-Format	24
Das externe XJT-Modul.....	25
Akkuanschluss	26
Micro-SD-Karte ca 1-2GB	26
Ansagetexte in Deutsch und Töne gibt es hier:	26
USB Mini Buchse.....	26
Lehrer-DSC Buchse 3,5mm Mono	26
Kopfhörer 3,5mm Stereo	26
Sender Akku laden 6 Zellen NiMH (Eneloop-Typ).....	27
Umrüsten auf Lipo-Akku mit 2 Zellen 2S.....	28
Beispiel: Umbau auf einen anderen Akku	29
Knüppelaggregate X9D umstellen von Mode 1 auf Mode 2	32
Mode umstellen	32
Empfängerkombinationen Externes XJT oder DJT Modul	33
X8R-Empfänger mit Smart-Port und SBus jumpern (gilt auch für X6R, X4R).....	34
SBus umwandeln in zusätzliche Servoausgänge oder in ein CPPM Summsignal.....	35
X4R und X4R-SB Telemetrie-Empfänger.....	36
6 Tasten Menüführung lang oder kurz drücken.....	37
Screenshotfunktion: (ab openTx V2.1).....	38
Symbole als Auswahlliste vereinfachen die Eingaben (ab openTx V2.1).....	38
Schalter und Potis Namen und Funktion zugewiesen (ab openTx V2.1)	39
Serielle Port kann mehrere verschiedene Funktionen übernehmen (ab openTx V2.1)	39
Trimmwerte anzeigen lassen (ab openTx V2.1).....	39
Softwarestruktur von OpenTx Frsky Taranis ab V2.017	40
Programmierprinzip OpenTX EVA Prinzip.....	41

Vereinfachtes Funktions- und Programmierschema openTx.....	42
Bezeichner und Bedeutungen.....	43
Eingabewerte in Rot, so wie sie auch in den Menüs auftauchen	43
Eingaben und Werte editieren	44
Grundprinzip der Bedienung ist immer gleich!	44
Editieren und abspeichern.....	44
Die wichtigsten Tastenfunktionen aus dem Hauptbildschirm	44
Werte in einer Checkbox ein/auschalten/freigeben <input checked="" type="checkbox"/> <input type="checkbox"/>	45
Bearbeiten von Zeilen.....	45
Texte eingeben.....	45
Arbeiten mit Auswahlwerten.....	46
Autoselect und Autoswitch von Schalter und Potis.....	46
Flugphasen aktivieren/sperren	46
Eingabe abschließen	46
Screenshotfunktion für LCD-Bildschirm (abV2.0.17)	46
Die Hauptansicht des LCD Display	47
Sender einschalten,.....	47
3 Startbildschirme.....	47
Grundsätzliche Darstellung.....	48
3 verschiedenen Hauptanzeigen	48
Der Kanal Monitor als Servoanzeigen.....	49
Statistik und Debugger Anzeige	50
Telemetriebildschirme verschiedene Darstellungen.....	50
Sender Grundeinstellungen Übersicht (1/9)	51
Die 9 Menüs sind:.....	51
Grundeinstellungen des Senders im Detail (1/9)	52
Die Micro SD-Karte Unterverzeichnisse (2/9)	56
Globale Funktionen ab openTx V2.10 (3/9).....	59
Lehrer / Schüler Einstellungen TR1-TR16 (PPM1-PPM16) (4/9)	60
Das PPM Signal, PPM Impulsrahmen, Einstellungen, im Detail	63
Beispiel: Flugsimulator am PC.....	64
Beispiel: Trainer Funktion aktivieren und testen.....	65
Beispiel: FPV Spotterfunktion.....	66
Beispiel: Kabelloser Lehrer-Schüler Betrieb.....	67
Softwareversion (5/9)	69
Funktionstest aller EingabeTaster (6/9).....	70
Funktionstest aller Analoggeber (7/9)	71
Sender-Akku abgleichen:.....	71
Hardware einstellen (8/9).....	72
6 Stufen-Schalter als Potiersatz	73
Haptikmodul: (Ist in der Taranis Plus schon eingebaut).....	73
Serielle Schnittstelle für S-Port (Smart-Port) Output, Telemetrie-Input, Debug-Mode	74
Analoggeber abgleichen (9/9)	75

Modell Einstellungen.....	77
Die 13 Modell-Menüs:.....	77
Neue Modell erzeugen mit dem LUA Script Modellgenerator	78
Modell auswählen, kopieren, verschieben, anlegen (1/13)	79
Modell Grundeinstellungen Menüs Übersicht (2/13).....	80
Im Internen HF-Modul die Betriebsart einstellen.....	82
Im Externen HF-Modul die Betriebsarten einstellen.....	82
Modell Grundeinstellungen im Detail (2/13)	83
HF Module und Protokolle, Bind, Failsafe,.....	86
Externes HF-Modul	86
Failsafe Mode einstellen.....	87
Failsafe im Detail: XJT-Modul im X16 / D16-Mode für X8R, X6R, X4R, LR12.....	88
Binden des Senders mit internem HF- Moduls an den Empfänger	89
Binden des Senders mit externem HF- Moduls an den Empfänger.....	89
Mehrere Empfänger an ein Modell binden	89
Mit einem Sender gleichzeitig 2 Modelle steuern	89
Reichweitentest durchführen und zu erwartende Werte mit X9E und X6R.....	90
Hubschrauber Grundeinstellungen (3/12)	91
Übersicht Taumelscheibenmischer für 120° Taumelscheibe.....	93
Flugphasen / Flugmode definieren (4/13).....	94
Trimmung bei Flugphasen / Flugmode.....	95
Inputs als Signalvorverarbeitung (5/13).....	96
Beispiel: Dualrate und Exponentialanteil in 3 Stufen umschalten.....	97
Inputs Untermenü und Detailansicht:.....	99
Beispiel: Telemetriewerte anpassen, Geschwindigkeit normieren	100
Beispiel: Dualrate/Expo mit 3-Stufenschalter	101
Beispiel: Dualrate/Expo mit 2-Stufenschalter	102
Mischerfunktionen (6/13)	104
Mischer Hauptbildschirm und Übersichten	104
Bedeutung der Spalten in der Mischer Hauptanzeige 6/13:	104
Zeilen einfügen und ins Untermenü (wie bei Dualrate/Expo).....	105
Mischer Detailansicht, Untermenü, Werte editieren	105
Grundprinzip der Mischerberechnungen: gilt ab OpenTx2.0!.....	109
Beispiele der Mischer Berechnungen ab OpenTx2.0	110
Grafische Darstellung der Bereichsanpassung und Bereichsumrechnung in 2 Schritten	111
Mischerverarbeitung im Detail: EVA-Prinzip Eingabe, Verarbeitung, Ausgabe	112
Grundverständnis der Mischerberechnungen	113
Beispiel: Schalter als Mischer-Quelle mit 2 -Stufen und 3-Stufen Schaltern.....	114
Beispiel: Landeklappen mit 3 Stufen-Schalter fahren	115
Beispiel: Motor Sicherheits-Schalter (Throttle Cut) in 4 Varianten.....	116
Beispiel: Mischer mit Offset und Weight anpassen.....	119
Beispiel: Servos für Delta und V-Leitwerk richtig mischen.....	120
Beispiel: Mit Mischern 2 Motoren auf Gleichlauf einstellen	121
Beispiel: Mischer Bereiche einstellen und berechnen im Detail, Kurven als Variante	122
Servotrim -Mitte, -Wege, -Umkehr, 7/12).....	125

openTx für Taranis Anleitung Deutsch	
Bedeutung der Spalten: In der Statuszeile wird ein Hilfstext angezeigt.....	125
Trimmwerte der Kanäle als Subtrim (Servo-Mittelstellung) übernehmen und abspeichern.....	128
Servo-Min und Servo-Max genauer betrachtet:.....	129
Kurven eingeben (8/13).....	131
Kurven mit 2-17 Stützpunkten.....	131
Kurven editieren	132
Auswahl der Kurventypen für feste oder variable X-Werte	133
Umschalten der Kurvenarten und Kurventypen	133
Globale Variablen GV1-GV9 (9/13)	135
GVAR festen Wert zuweisen.....	135
GVAR Wert von anderer Flugphase übernehmen	135
GVAR veränderbare Werte zuweisen	136
GVAR in den Spezialfunktionen veränderliche Werte zuweisen.....	136
Anwendung von Globalen Variablen GVx	137
Beispiel: DR/Expo und Anwenden von globalen Variablen	138
Beispiel: Anwenden von Globalen Variablen in 4 Schritten.....	138
Beispiel: Globale Variablen vorverarbeiten und Bereich einschränken/anpassen.....	140
Beispiel: Globale Variablen auf Gewichtung und Expo anwenden.....	141
Logische Schalter L1 ... L32 (10/13)	143
Spalte 1 die Bedingungen:.....	143
Spalte 2 und 3 die eigentlichen Vergleichswerte, Variablen oder Festwerte	144
Spalte 4 enthält Freigabeschalter bzw eine weitere UND Verknüpfung	144
Spalte 5 und 6 sind einstellbare Zeitdauern und Verzögerungen	145
Neue Funktionen für Logische Schalter ab opentx2.0.....	146
SRFF Ein SR-FlipFlop mit Bedingungen.....	146
Ersatz der „t“ toggle Funktion: Erfolgt jetzt über einen Log.Schalter.....	146
Puls Einen einmaligen Impuls erzeugen (Edge), wie ein Monoflop	147
Takt Ein einstellbarer Taktgenerator	147
Range Einen Analogwert als Bereich abfragen (kommt erst noch).....	147
Modulo.....	147
Beispiel: Fensterbereiche, Range-Bereich definieren (hier per Gasstellung Throttle)	148
Beispiel: Glühkerzenheizung automatisch aktiveren wenn Gas fast auf Leerlauf.....	149
Übersicht der Variablen für Logische Schalter.....	151
Spezial Funktionen SF (11/13)	152
Beispiel: Telemetrie Grenzwerte setzen und Warntöne erzeugen	154
Beispiel: Variometer umschalten und Telemetrie-Daten alle 0,1s aufzeichnen	154
Beispiel: Ansagen, Werte und Töne in den Spezialfunktionen auslösen.....	154
LUA Custom Scripts Interpreter Sprache (12/13).....	157
Beispiel: LUA auf dem Sender und am PC einrichten	158
Fertige Voreinstellungen, Templates, Th9x, 9XR, 9XRPro(13/13)	162
Telemetrie-einstellungen (13/13)	163
Telemetriedaten parametrisieren für die Anzeige am Sender.....	164
Analoge Eingänge A1 A4 Bereiche anpassen je nach Empfänger	164
Empfangsfeldstärke RSSI des Empfänger.....	165
Sendeantenne-Funktionsüberwachung SWR.....	165
Das Format des seriellen Empfangsprotokoll (UsrData):.....	165

Konfiguration der Balkenanzeigen für Telemetrie:	165
→XJT-HF-Modul im D16 (X16) Modus betreiben!	166
Beispiel: Telemetrie-Grenzwerte setzen und Warnton auslösen	166
Anzeige der Telemetriedaten am Sender je nach Einstellungen	167
Balkenanzeigen mit Schwellwerten, Ansprechschwellen	167
Eingänge A1 und A2 mit Min, Max Max, und LiPo-Zellen.....	167
Höhenmesser, Geschwindigkeit, Temperaturen	168
GPS Daten	168
Telemetrie Alarmer, Warnungen und Ansagen.....	169
Alarmer vom Frsky-Modul (DJT, XJT).....	169
Warnungen.....	169
Variometer einstellen	170
Beispiel: Frsky Variometer am X8R anschließen und Bereiche einstellen	172
Stromsensor / Spannungssensor einstellen	174
FAS-40 und FAS-100 Stromsensoren liefern Strom und Spannung	175
Externe Spannungs- und Stromsensoren an A1 und A2	175
Spannungssensor FLVSS mit Smart-Port Anschluss	176
Übersicht der Telemetriewerte Stand: opentx V2.07	177
Die Neue Telemetrie ab openTx V2.10	179
Überblick:	179
Telemetriedaten	180
Beispiel: Anzeige der Einzel-Zellenspannungen vom FLVSS-Sensor.....	182
Beispiel: Verbrauchte Kapazität in mAh und Leistung in W ermitteln.....	183
Beispiel: Telemetrie unter Companion V2.1 konfigurieren	184
Übersicht aller Telemetrie ID-Bereiche 2Byte und 1Byte.....	185
Etwas mehr ins Detail der Telemetrie:	187
Companion V2.10 Telemetrie Werte definieren und Berechnungen machen	189
OpenTx V2.1x Telemetriewerte in Anzeigewerte umrechnen im Detail.....	190
Ergänzung zu den Analog Eingangswerten A1 bis A4:	191
Verrechnung am A/D Wandler:.....	191
Beispiele 3-6 Zellen Akkuspannungen richtig anzeigen:	191
Parameterübersicht für Telemetrieverrechnung.....	192
OpenTx Update am Sender mit der USB Schnittstelle, SD-Karte	193
Achtung: Ab openTx V2.00 wurde vieles umgestellt und vereinfacht.....	193
Hinweis für CompanionV1.52, openTx r2940 und OpenTx V2.00	193
Teil B Companion Einführung Step by Step	195
Simulation des Sender, Grundeinstellungen, Modell erzeugen	197
Sender-Simulation, Taranis Simulator, Ausgabe-Simulation	199
Softwaresimulation als Kanalsimulator	199
Sendersimulation mit allen Menüs und Anzeigen (immer in Englisch).....	199
Das erste Modell „Programmieren“ Schritt für Schritt.....	200
Das Mischer Fenster am Beispiel Kanal 6.....	201
Die Mischer-Verrechnung geht im Prinzip so	202
Erste Beispiele einfach mal eingeben und abändern.	203

Templates und Modellkonfiguration.....	206
Der Modellkonfiguration Wizard	206
Daten von und zum Sender übertragen	207
Im Flashspeicher wird die Sender-Firmware gespeichert.....	207
Im EEprom werden die Modelldaten gespeichert.....	207
Die SD-Karte dient als Modell-Archiv.....	207
Modelle vom PC in den Sender übertragen und zurück.....	208
Taranis mit andere OpenTx-Firmware überschreiben.....	213
Neue Firmware in den Sender flashen Schritt für Schritt	214
Teil C Modelle mit Companion9x programmieren	216
Prinzipdarstellung der Mischerverarbeitung.....	216
Mischerverarbeitung Kanalzuweisung Weganpassung Servo Ruder	216
Übersicht der Mischer Quelle und Ziel.....	217
Mischerprogramme Übersicht Motormodelle	217
Mischerprogramme Übersicht Segler	218
Mischerprogramme Übersicht Delta.....	219
Beispiel: Fertiges Programm für ein Motormodell.....	220
Beispiel: Schritt für Schritt Programmierung eines Modell mit den Mixern	221
Zu dem heiligen Thema: Ein Querruder Servo invertieren anstatt beide positiv	228
Beispiel: Dualrate und Expo als Vorverarbeitung und Signalanpassung der Knüppel.....	231
Beispiel: Einfachster Deltamischer für Nuri mit 2 Servos für QR+HR gemischt	232
Segler mit 4 oder 6 Klappen 2QR 2WK 2BR programmieren.....	235
Das „vermischen“ von Funktionen.....	240
A: Speed- und Thermik -Stellung dazu programmieren.....	241
B: Butterfly zum Landen	242
D: Dynamische Bremsklappen, Wölbklappen, Querruder	248
E: Berechnete Einstellwerte für Weight und Offset:	249
F: Fertiger Segler mit 4 und 6 Klappen	252
Beispiel: 4 Klappen-Segler Butterfly, Wölbklappen, Speed Thermik variabel einstellbar	253
Teil D Viele Beispiele, Tips und Tricks	261
Beispiel: Die grundsätzliche Dinge der Programmierung	261
Beispiel: Logische. Schalter mit logischen Verknüpfungen und Abfragen.....	262
Beispiel: Fahrwerks Door Sequenzer Variante 1.....	263
Beispiel: Fahrwerks Door Sequenzer Variante 2.....	264
Beispiel: V-Leitwerk Ruderwirkung und Mischer	265
Beispiel: Mischer Querruder auf V-Leitwerk mischen.....	266
Beispiel: Mischer Quer auf Seite Mischer Seite auf Quer.....	268
Beispiel: Mischer Seite auf Höhenruder soll immer etwas Tiefe bringen	269
Beispiel: Dynamische Servo Geschwindigkeit mit einem Integral-Mischer	272
Beispiel: Mischer Gas auf Höhe mit Kurve, ohne Kurve	274
Beispiel: Mischer Gas auf Höhe mit Kurve aber variabel zumischen per S1 und GVAR	275
Beispiel: Logische Schalter Funktion SRFF = SR-FlipFlop mit Bedingungen.....	276
Beispiel: Logische Schalter Funktion Puls (Edge) einen einmaligen Impuls erzeugen	277
Beispiel: Logische Schalter Funktion Takt = TIM ist ein Taktgenerator	277
Beispiel: Logische Schalter Funktion Range = einen Analogwert als Bereich abfragen	277
Beispiel: Flip Flop mit SH-Taster für Set und Reset.....	278
Beispiel: Stufenschalter mit SH und globalen Variablen nachbilden.....	279

openTx für Taranis Anleitung Deutsch

Beispiel: Automatisch verschiedene Telemetriewerte ansagen lassen.....	281
Beispiel: Mit SRFF Flip-Flop und SH Log Daten Aufzeichnung Start/Stop	282
Beispiel: Flip Flop mit Toggle Funktion Ein/Aus auf Timer anwenden	283
Beispiel: Flip Flop mit Log. Schalter Lx (CSx) auf Timer anwenden.....	284
Beispiel: Timer Start, Stop, Reset mit Taster SH	285
Beispiel: Potipositionen exakt einstellen.	286
Beispiel: Einmalige Ansage eines Potiwertes nach der Veränderung	287
Beispiel: Schalter, Schaltkanal, einfache Zeitfunktion, Blinken	288
Beispiel: Gaslimiter mit opentx Taranis wie bei einer Graupner MX16.....	291
Beispiel: Pan -Tilt Kamera mit Limiter-Grenzkurve die nicht unter/überschritten wird.....	292
Beispiel: PPMus - Werte berechnen für Flightcontroller und APM Anwendungen	294
Beispiel: APM- Mode mit 6 Stufen bzw mit 2 Schaltern einstellen.....	295
Beispiel: 6-Stufenschalter umrechnen auf andere PPMus-Werte.....	296
Beispiel: PPM-Werte im Vergleich Taranis, Spektrum, Graupner, Futaba, Multiplex	298
Beispiel: Telemetriewerte am Sender einstellen (A1, A2, A3, A4, RSSI, Vario)	299
Beispiel: Vario einstellen und Höhenansagen aufrufen.....	302
Beispiel: Servotester mit einstellbaren Wegen und Zeiten in 2 Varianten	303
Beispiel: Langsame Servo-Bewegungen mit Slow up Slow down im Mischer	305
Beispiel: Flightmode mit Fade-In Fade-Out langsam Ein-und Ausblenden.....	307
Beispiel: GVAR in den einzelnen Flugphasen anwenden und variabel verändern	308
Beispiel: Kanalverriegelung, Veränderung des Kanalwert sperren.....	311
Beispiel: Variable Querruder-Differenzierung, im Flug mit GVAR 0-50%	312
Beispiel: Variable Querruderwege, im Flug mit GVAR einstellbar 50-100%	312
Beispiel: Querruder mit asymmetrischer Anlenkung, Spoiler variabel Speed-Flugphase.....	317
Beispiel: Eigene Ansagetexte als *.wav Dateien erzeugen	320
Beispiel: Schalterstellungen und/oder Schalterwert ansagen	322
Beispiel: Eigene Sprachdateien mit Programm Balabolka erzeugen (Freeware)	325
Beispiel: Modellname einmal bei Aufruf ansagen lassen.....	328
Beispiel: Splashscreen für Taranis anpassen und erzeugen.....	330
Beispiel: Splashscreen (Start-Screen) einfügen oder ersetzen.....	331
Beispiel: Textdateien als Checkliste auf das LCD-Display bringen.....	332
Beispiel: Input Signalvorverarbeitung, statt Mischer mit einem freien Hilfskanal	333
Beispiel: Taranis Sender am Flugsimulator anstatt eines Joystick verwenden.....	334

Teil E Companion V2.xx und openTx V2.xx 336

CompanionTx V2.00x Start und Senderprofil anlegen	338
Der neuen Modell Wizard ab companion V2.00	341
F4, F5, F6 Simulation von Telemetriewerten, Trainer und Debugger für LUA.....	348
Der neue Bootloader ab openTx V2.00	353
Neues Verfahren für das OpenTX Firmwareupdate auf Taranis ab OpenTx V2.00	355
Den Bootloader selbst updaten:	356
Hilfe: Falsche OpenTx Software geflasht, LCD dunkel, nichts geht mehr, was tun?	357
Merkhilfe Kanal und Schalterbelegung	358

Firmwareupdate der Sensoren, Empfänger, HF-Module 359

Variante 1: Mit dem PC die Smart-Port Sensoren, Empfänger, HF-Module updaten.....	359
Mit dem PC die X-Empfänger und das externe HF-Modul updaten (ETSI V1.8.1)	363
Variante 2: Mit dem Sender S-Port-Sensoren, Empfänger, HF-Module updaten	368
Mit dem PC SPORT-ID Adresse ändern, Telemetrie Übertragungsrate anpassen.....	370

Teil F Der FrSky Pultsender X9E..... 372

X9E Ein erster Überblick.....	372
X9E Ansichten und Einblicke	373

Teil G Die Sender 9XR Pro, 9XR, Th9x mit OpenTx V2.0x..... 382

Den Sender 9XR Pro mit openTx V2.0x flashen.....	382
--	-----

openTx für Taranis Anleitung Deutsch	
Die Sender 9XR, Th9x mit OpenTx V2.0x bedienen und flashen	383
Bezeichner und Bedeutungen	384
Eingaben/Werte editieren	385
Die wichtigsten Tastenfunktionen aus dem Hauptbildschirm	385
Editieren und abspeichern.....	386
Navigation durch Zeilen und Eingaben in mehreren Spalten	386
Zeilen in denen Eingaben nur an einer Stelle/Spalte möglich sind	386
Werte in einer Checkbox ein/ausschalten <input checked="" type="checkbox"/> <input type="checkbox"/>	386
Eingabe abschließen	387
Bearbeiten von Zeilen.....	387
Texte eingeben.....	388
Arbeiten mit Auswahlwerten.....	388
Flashen von Th9x und 9XR mit AVR ISP-Schnittstellen.....	389
Notwendige Umbauten vorab:.....	389
Flashen Teil 1 von 4 Einlöten der 6 Leitungen für das Flashen des Senders	390
Funktionen der Schalter und Taster am Sender	393
Flashen Teil 2 von 4	394
Der AVR ISP-Programmer: mySmartUSB Light.....	395
Arbeiten mit CompanionTx.....	397
Einstellungen für das Brennprogramm.....	397
Grundeinstellungen für CompanionTx.....	398
Flashen Teil 3 von 4	399
Flashen Teil 4 von 4	400
Ablauf:	401
Th9x Schaltplan der Stromversorgung für Akku laden	402
Link-Sammlung der Modifikationen.....	405
Die Programmierer und das Team von openTx.....	407
Instructions for building and programming	408
Building from source	408
From author of the software:	408
EU-Konformität, CE-Kennzeichnung, FCC-Prüfprotokolle,.....	410
Konformitätserklärungen für EN 300328 V1.7. 1 FrSky-Baugruppen bis 31.12. 2014.....	411
Konformitätserklärungen für EN 300328 V1.8. 1 FrSky-Baugruppen ab 01.01.2015.....	412
CE für Externe HF-Module	413
CE für die Empfänger.....	414
Hinweis zu der neuen EU Sendenorm ETSI V1.8.1.....	417

Disclaimer

THIS SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PURPOSE. YOU WEAR THE ENTIRE RISK OF QUALITY AND PERFORMANCE OF THE PROGRAM. TAKING TO YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION IN THE EVENT THAT THE PROGRAM PROVE DEFECTIVE.

Haftungsausschluss

Die Software ist wie sie ist und ohne Garantien irgendwelcher Art, weder ausdrücklich noch sinngemäß, einschließlich der Gewährleistung für die Marktgängigkeit oder Eignung für einen bestimmten Zweck. Der User übernimmt das vollständige Risiko des Gebrauchs der Software. Unter keinen Umständen ist eine Person, ein Unternehmen oder eine Organisation, die an der Entwicklung dieser Software beteiligt ist, für irgendwelche Schäden haftbar, die aus dem Gebrauch, dem Missbrauch oder dem Unvermögen, die Software anzuwenden, entstehen.

Das deutsche Handbuch besteht aus 7Teilen:

- A) Den Sender FrSky Taranis und alle seine Funktionen.
Die Softwaregrundlage zu openTX ist open9x für die Sender TH9x und 9XR
Sie enthält Erweiterungen und Anpassungen die in den Foren 9xforums, openTx, er9x
Diskutiert werden. Die Screens sind in Deutsch teilweise in Englisch bei bestimmten Absätze
und Bezeichner, da diese in Companion9x oder den Foren immer wieder auftauchen.
- B) Das Programm CompanionTx zum Programmieren, Simmulieren, Flashen des Senders und
Modelledateien ins EEprom übertragen.
- C) Ausführliche Mischer- und Programierbeispiele Step by step
- D) Zahlreiche Programmier-Beispiele aus vielen Bereichen
- E) Companion V2.00, OpentxV2.00 installieren und Bootloader einrichten, Startbild ersetzen
- F) Pultsender X9E
- G) Sender Th9x und 9XR mit ISP-Schnittstelle programmieren

Wer Anpassungen machen will soll das tun und dann auch bitte wieder veröffentlichen.
Das bisherige deutsche Handbuch für open9x für die Sender TH9x und 9XR gibt es weiterhin
unter: <http://code.google.com/p/open9x/>

Dieses Handbuch passt auch für Th9x, 9XR und 9XR-Pro

Auch diese Sender können auf opentx V2.00 geflasht werden. Durch deren begrenzten Speicher
und den 8bit-Prozessor werden aber nicht alle neuen Funktionen voll unterstützt.
Das Sender- und Modellupdate erfolgt wie bisher (siehe Handbuch open9x).
Der Sender **X9R-Pro**, mit AT SAM 32bit Prozessor, hat fast die gleiche Leistungsfähigkeit wie
die Frsky Taranis und damit auch alle neuen Funktionen wenn man auf openTx V2.0 updatet.

Tip zum Ausdrucken:

A4 doppelseitig mit Funktion A5-Broschüre, dann hat man ein kleines, praktisches Ringbüchlein
oder Heft und man kann mal ein paar Blätter austauschen oder ergänzen.

Teil A Der Sender und seine Funktionen

Der FrSky Sender Taranis

Wenn Sie einen eigenen Sender entwickeln könnten, was würden Sie alles integrieren? Genau diese Frage stellte sich die Firma FrSky und den Kunden.

Das Ergebnis heißt Taranis! Frsky hat erfolgreich einen Hightech-Sender zu einem niedrigen Preis vorgestellt, der die meisten High-End Markensender am Markt übertrifft.

Nun könnte man sich Fragen wo hat FrSky gespart um den Preis niedrig zu halten? Qualitäts-Kompromisse um den Preis niedrig zu halten ist nicht die Art von FrSky.

Bei FrSky fehlt vielleicht eine schicke bunte Werbekampagne und ein riesiges Marketing-Budget, es wird aber nicht an der Hardware geknausert!

Das wichtigste für jeden Sender ist das Aufrechterhalten einer felsenfesten Verbindung zum Empfänger. Frsky ist bekannt für die Verwendung des ACCST Frequenz-Sprungverfahren. Dabei wird das ganze 2,4Ghz Band im Sprungverfahren benutzt und sehr schnell die Frequenzen gewechselt (47 Kanäle in 9ms) um eine hervorragende Zuverlässigkeit und Reichweite zu erzielen. Vieles kann die Verbindung vom Sender zum Empfänger beeinflussen. Deshalb haben alle FrSky-Empfänger eine RSSI-Signalauswertung (receiver signal strength indication) integriert die per Telemetrie zum Sender übertragen wird. Der Taranis Sender zeigt dauernd die Empfangsqualität (RSSI-Signal) des Modell am Sender-Display an und erzeugt Alarmmeldungen bevor das Empfängersignal kritische Werte erreicht. Das kann Abstürze verhindern und macht das Hobby sicherer.

Zusätzlich zum RSSI-Signal hat Taranis weitere Sicherheitseinrichtungen integriert.

Receiver Lock bzw. Modell Match bindet den Empfänger fest an das Modell das im Sender ausgewählt ist. Somit ist kein Fliegen mit einem falsch ausgewählten Modell möglich.

Taranis hat 3 Failsafe-Methoden.

1- Hold halten der letzten gültigen Werte,

2- voreingestellte Einstellungen anfahren (Gas auf 30%, Flaps unten, Querruder neutral usw.)

3- keine Ausgangssignal und damit einen Flight Controller starten

(mit Homing-Funktion) Durch die empfindliche und einstellbare RSSI-Funktion werden sie fast nie den Failsafe-Mode auslösen.

Sprach-Ansagen wie ein Copilot, der Sender kann Alarmer auslösen und Sprach-Ansagen machen die am Lautsprecher oder Kopfhörer ausgegeben werden

Zeitansagen, Spannungswarnungen, Fahrwerk, Vario-Signale, Höhengaben usw. können alle durch Sprach-und Sound-Files auf der SD-Karte ausgelöst werden.

Die Software openTx für den Taranis-Sender ist eine Entwicklung von Modellfliegern und Programmierern aus dem RC Bereich und open-source, also frei verfügbar. Die Programmierer der Sender-Software **openTx** und der **Companion9x**-Software, die es für Linux, Window und Macintosh gibt, sind sehr empfänglich für Anregungen und Wünsche der Benutzer. Es gibt keine Beschränkungen oder Einschränkungen.

Mit 60 Modellspeichern, 64 freie Mischer, 9 Flugmode, Sequenzern, frei programmierbare Servogeschwindigkeiten und Verzögerungen, alle Arten von programmierbaren Schaltern, Funktionen, Kurven und Triggerereignisse, freie Zuordnung von Eingänge, Ausgängen und Kanälen. Alles kann mit allem verrechnet und logisch verknüpft werden.

Diese vielen Möglichkeiten und die komplexen Programmiermöglichkeiten könnten zum Alptraum werden, aber durch die open-source Gemeinde gibt es ein Programm, **companion9x**, mit dem wir alles bequem am Computer (für Windows, Ubuntu, Linux, Mac) testen, programmieren und

openTx für Taranis Anleitung Deutsch

simulieren können bevor wir es per USB-Kabel in den Modellspeicher des Sender oder auf die micro-SD-Karte auf der Rückseite übertragen.

Dazu gibt es noch fertige Voreinstellungen (Wizzard) die einem viel Programmier- und Einstellungsarbeit abnehmen.

Falls Sie noch andere Sender und Empfänger haben können sie auch diese Sende-Protokolle mit einem passenden Modul im JR-Format an der Rückseite integrieren.

In den Modellgrundeinstellungen werden dann diese Protokolle für das externe HF-Modul einfach ausgewählt und das interne FrSky HF-Modul kann abgeschaltet werden.

Damit kann man Module von Futaba, Spektrum, JR, Graupner, Assan und andere verwenden.

Für die UHF-Freunde kann man 12V direkt zum UHF-Modul durchschalten und braucht keine extra Verkabelung oder extra Akkupack.

Das Projekt Taranis

Der Frsky Taranis Sender ist eine neue Art der Zusammenarbeit.

Zum erstenmal hat ein namhafter Hersteller der R/C Industrie mit den Entwicklern der open-source R/C Gruppe eng zusammengearbeitet um Hardware und Software so zu entwickeln und zu verbessern, dass ein open-source Sender entsteht, der sehr preiswert ist, aber mehr bietet als alle großen Marken-Hersteller.

Das bedeutet, es gibt beim Taranis-System keine Einschränkungen und Beschränkungen in den Funktionen so wie bei den Marken-Herstellern und Ihren Marketing-Entscheidungen, die den vollen Funktionsumfang nur in Ihren Hochpreis-Sender anbieten.

Das Taranis-System mit ihrer offenen Hardware und Software-Struktur wird auch in Zukunft weiterentwickelt und angepasst. Neue Anforderungen und Entwicklungen können mit dem open-source Prinzip sehr schnell umgesetzt und für verschiedenste Benutzer angepasst werden.

Das System openTx für Taranis ist eine Weiterentwicklung aus open9x für die Sender Th9x, 9XR und andere offene Hardware-Systeme.

Open9x gibt es schon seit mehr als 5 Jahren, ist sehr ausgereift und wurde immer wieder an unterschiedliche Sender, Prozessoren und Hardware angepasst und erweitert.

Mit **OpenTx** wurde das System an die Hardware von FrSky mit einem 32 Bit Prozessor angepasst und nochmal erheblich erweitert.

Damit steht von der Hardware und von der Software ein System zur Verfügung das absolut an der Spitze der R/C-Technik steht.

Das Taranis-System von FrSky ist aber auch darum sehr preiswert, weil bewusst viele Standardkomponenten verwendet wurden.

Das Gehäuse stand von einem erprobten anderen Sender zur Verfügung, die Elektronik, Prozessor und Platinen-Layout sind Anpassungen der open-source ersky9x- Entwicklung, die Software ist open-source

Telemetrie, Sende- und Empfängermodule sind von FrSky.

Das sehr schnelle PXX- Protokoll sind Weiterentwicklungen von FrSky und das sichere AFHSS ACCST ist schon lange in Betrieb und ausgereift.

Die sehr hochwertigen Knüppelaggregate sind von einem namhaften Hersteller der auch die großen Markenhersteller beliefert und dort nur in den Hochpreisprodukten verbaut wird.

Alles in allem eine High-Tech-Produkt in einem schlichten, aber funktionalen Gehäuse ohne unnötige Design-Gimmicks und Schnick-Schank. Die inneren Werte zählen.

Die Funktionen in Stichworten, ein Überblick:

- Volle Telemetrie RSSI Signalauswertung mit Vor-Alarm wenn die Schwellenspannung sinkt
- Selbsttest der Sender-Antenne, überwacht dass auch HF abgestrahlt wird.
- 16 Kanäle im internen HF Modul, weitere 16 Kanäle mit zusätzlichem HF-Modul (max 32)
- 60 interne Modellspeicher und weitere auf micro SD-Karte
- 64 freie Mischer
- Knüppel Mode 1 - Mode 4 oder beliebig belegbar
- 9 Flugmode Flugzustände
- 32 Kurven mit 2-17 Punkten mit und ohne verrunden der Kurven per Spline
- 32 Logische Schalter Schalter (bzw. Prog.Schalter, Custom switch)
- 64 Programmierbare Spezial-Funktionen
- 9 globale Variablen GVAR pro Flugphase
- Sprachansagen, Sound und Alarmer, Variotöne integriert
- USB Schnittstelle, micro- SD-Karte, serielle Schnittstelle für Erweiterungen
- USB für Firmware Update, Sound, Read, Write Modelle und Einstellungen auf SD-Karte
- USB Standard PC Joystick Interface für PC Flugsimulator, kein PPM to USB Interface nötig
- Kreuzknüppel, 4-fach Kugelgelagert, hochwertige Potis, einstellbare Rasterung, sanfter Lauf
- Zahlreiche Eingänge (4 Sticks, 4 Trimmungen, 2 seitl. Geber, 2 Potis, 8 Schalter)
- Abgleichbare Sticks und Potis
- DSC-Lehrer/Schüler Buchse mit bis zu 16 PPM-Kanälen TR1-TR16 Input und Output
- Grosses LCD Display 212x64 Pixel, 16 Graustufen, hintergrundbeleuchtet
- Echtzeit Datenlogger für alle Telemetriedaten auf SD-Karte
- Empfänger mit Modellmatch (mit FrSky Empfänger und PXX Protokoll)
- 2 Timer in verschiedenen Betriebsarten, UP, Down, % von Knüppel, Modell-Flugzeit
- Trimm Auflösung einstellbar von grob bis superfein, exponentiell
- Erweiterte Wege von 100% auf 150%
- Erweiterte Trimmung von 25% auf 100%
- Standard JR 3,5mm Trainerbuchse, DSC-Buchse für PPM-Signal Ausgang oder Eingang
- Frei programmierbare Trainerfunktion oder FPV-Spotterfunktion
- 32 bit Prozessor ARM Cortex M3 120Mhz
- Companion9x, Programm (Windows, Mac, Linux) Companion9x zum programmieren, simulieren, updaten, lesen und speichern von Modellen und Einstellungen
- 8 Sprachen durch Update der Firmware (Auslieferung in Englisch), bzw. beliebig anpassbar
- Sound Mischer für Töne Ansagen, Alarmer, Vario, Warnungen, Hintergrundmusik
- Hoppingsequenz aus 250 Kanäle statistische gleichmäßig verteilt 47 Kanäle in 9ms mit 300kHz Bandbreite
- Internes Telemetrie-HF-Modul für die Übertragung von bis 16 Kanälen. Mit einer Refreshrate von 9ms für Kanal 1-8 und 18ms für Kanal 9-16. Unterstützt das vorhandene D8-Protokolle (alle Empfänger von Typ D und VxR-II) das neue D16 Protokoll und den Long Range Mode LR12.
- Long Range System mit 12 Kanälen, sendet ca 3 mal weiter als normale 2,4GHz Systeme

openTx für Taranis Anleitung Deutsch

- Die Zuordnung der Kanäle intern und extern ist frei. D.h. man kann mit einem zusätzlichen externen XJT Modul ein redundantes System aufbauen und 2 mal die gleichen 16 Kanäle übertragen oder aber bis zu 32 Kanäle oder alles dazwischen.
- Das internen XJT HF-Modul hat eine Modell-Match Funktion und 3 Failsave Mode: letzte Position halten, voreingestellte Positionen anfahren, alle Positionen auf Mitte.
- Der externe Modulschacht (keine 6V) ist im JR-Format und kann abhängig vom Modul weitere 16 Kanäle im PXX Protokoll ausgeben oder PPM Signale für div andere Module oder serielle Signale Daten für DSM2 Module von Spektrum
- Telemetrie mit bis zu 5 frei einstellbaren Screens und frei konfigurierbaren Sprachansagen, unterstützt vorhanden Empfänger und Sensoren genauso wie die neuen S-Port Sensoren.
- Metrisches Einheitensystem. Integrierte Variometertöne (Daten vom Vario-Sensor im Modell) Datenloggerfunktion auf Micro SD-Karte
- Open-Source Firmware für schnelle Anpassungen, Erweiterungen von Spezial-Funktionen und Verbesserungen. Entwickler-Homepage: <http://www.openrcforums.com/>
- Eingebaute Ladeschaltung für NiMH Eneloop Akku 6 Zellen 2000mAh (alt 800mAh)
- Versorgung mit normalem 12 V DC-Netzteil

Sender-Set beinhaltet:

- Aluminum Koffer
- Taranis Sender
- Netzteil für senderinternes Akku Ladegerät 220VAC / 12VDC 0,5A
- Tragegurt
- Empfänger X8R - 16 Kanal, S-Bus, Smart Port Receiver
- NiMH Akku 6 Zellen, Eneloop-Type 2000mAh



Vorstellung von Opentx und Taranis, um was geht es?

Taranis ist ein Sender von FrSky der mit der open-source Software openTx als Betriebssystem läuft.

openTx ist eine Weiterentwicklung von **open9x**.

Ursprünglich ist open9x als Sender Software für einen ganz bestimmten Typ an Hardware entstanden. IMAX/FLYSKY/TURNIGY/EURGLE/AIRJUMP3/...9x und wie sie sonst noch alle heißen, ist eine Microcontroller Fernsteuerung aus China. Es ist aber immer der gleiche Sender der unter verschiedenen Labels verkauft wird.

Dieser Sender hat ein monochromes LCD Display mit 128x64 Pixel, 2 Kreuzknüppel, 3 Potis, 6 Umschalter, einen 3 Wege-Schalter und 4 Trimmräder.

Er arbeitet mit einem ATmega 64 Microcontroller mit 64K Flash und 2 K EEPROM

Das interessante an dem Sender ist sein Preis. Dieser kostet nur ca. 40-60€

Ein Programmierer namens Thomas Husterer, genannt THUS, hatte irgendwann mal eine zündende Idee als ihm klar wurde das man diesen Sender auch selber programmieren könnte und die Schaltpläne öffentlich zugänglich waren. Jeder Sender besteht aus den gleichen Grundkomponenten Kreuzknüppel, Trimmer, Schalter Display und einem einfachen Microcontroller.

Dann entschloss er sich die Original Sender Firmware durch seine eigene, selbstgeschriebene Software zu ersetzen und dies zu veröffentlichen.

Seither gibt es mehrere Projekte für den Sender TH9x als open source: th9x, er9x, ersky9x, open9x, gruvin9x und ein paar weitere.

OpenTx gibt es in diversen Menü-Sprachen, auch komplett in Deutsch, wobei die meisten Bezeichnungen eingedeutscht sind, teilweise aber auch bewusst weiterhin in Englisch gehalten sind.

Daraus ist nun openTx entstanden und an die neuen Hardwaremöglichkeiten eines modernen 32bit Prozessors angepasst und erweitert worden.

FrSky hat diese sehr ausgereifte und umfangreiche Sender-Software offiziell übernommen.

Zur eigentlichen Software im Sender gehört auch die Programmier-und Simulations-Software **Companion9x** für den PC

Ich empfehle dringend das Programm Companion9X zu benutzen, das vereinfacht vieles!

Weitere Hilfen, Infos, Templates, Mods, Hardware, Software findet man hier:

Das zugehörige Forum ist : <http://9xforums.com/forum/>

Die Software Infos : <http://www.open-tx.org/>

OpenTx findet man unter: <http://www.open-tx.org/downloads.html>

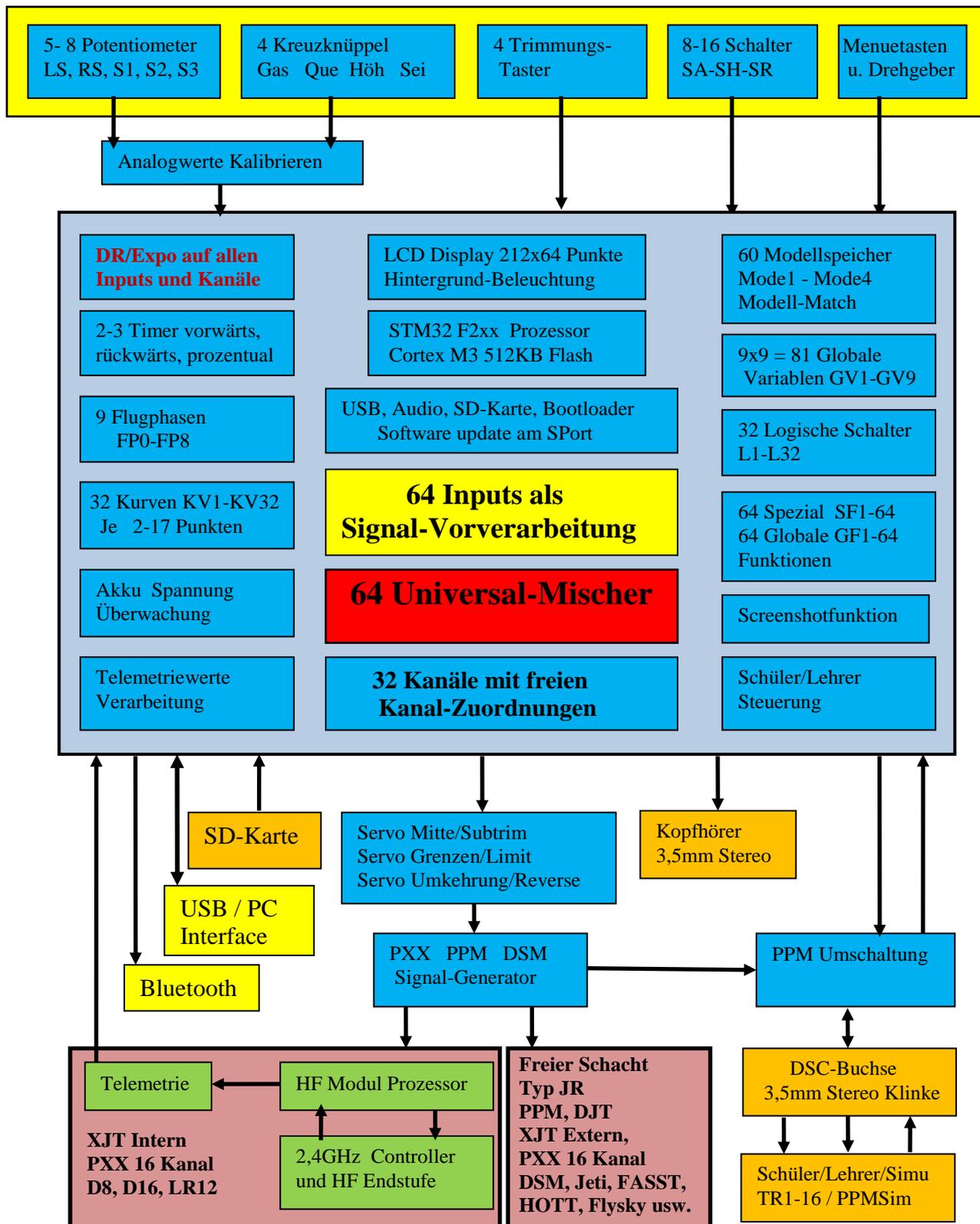
Companion9x findet man unter: <http://www.open-tx.org/2014/06/26/companion-2.0.5/>

Eine der besten Seiten über Frsky Baugruppen:

http://www.eflightwiki.com/eflightwiki/index.php?title=FrSky_Telemetry

Einen Überblick über die Taranis Hardware gibt das Blockschaltbild

Blockdarstellung des Senders Frsky 9XD Taranis und X9E



Funktionsablauf im Sender

Der Sender besteht aus 4 Haupt-**Eingabe** Baugruppen

1. 4 Sticks/Kreuz-/Steuerknüppel:
Englisch: **Rud**(der), **Thr**(ottle), **Ele**(vator), **Ail**(erons) (RTEA)
Deutsch: **Sei** (Seitenruder), **Gas**, **Höh** (Höhenruder), **Qeu** (Querruder) (SGHQ)
2. 4 (5) Potentiometers: **LS**, **RS**, **S1**, **S2**, (**S3**)
3. Trimmaster für die Steuerknüppel **TrmR**, **TrmT**, **TrmE**, **TrmA**
Deutsch (**TrmS**, **TrmG**, **TrmH**, **TrmQ**)
4. Schalter **SA** **SH**

Alle Analogeingänge (Steuerknüppel und Potis) werden kalibriert.

Die 4 Kreuzknüppel und auch alle Inputs und Mischer können durch Dualrate, Expo und Kurven verändert werden, bevor sie in den Mischern verarbeitet werden.

Eine Signalvorverarbeitung kann auf alle Inputs erfolgen.

Die Mischer sind das zentrale Element der Software. Sie steuern alles. Hier werden die Eingänge verarbeitet, gewichtet, Schalter, Kurve, Zeiten, Flugphasen zugeordnet und dann den 32 Ausgängen/Kanälen (**CH1 .. CH16 CH17 .. CH32**) zugeordnet.

Nachdem die Eingänge verarbeitet und den Ausgänge zugeordnet sind, werden mit den Limits/Subtrim die mechanischen Grenzen für die Servobewegung am Modell begrenzt, mit Subtrim die Mitte und mit Invers die Drehrichtung eingestellt.

Zum Schluss werden dann die Ausgangs-Kanäle mit dem Signal-Generator in einen seriellen Datenstrom oder PPM-Signal gewandelt und dem internen XJT HF-Modul und/oder einem externen HF-Modul zugeführt und an das Modell übertragen.

Es gibt noch weitere Arten von Eingangssignale: **PPM1-PPM16 (TR1-TR16)** Eingangssignale an der DSC-Buchse, Trainer/Schüler Eingang, empfangene Telemetriedaten

Dann gibt es noch für die Weiterverarbeitung, für Aktionen und Reaktionen:

32 Logische / progr. Schalter **L1-L32** als virtuelle Schalter

64 Spezial Funktionen **SF1-SF64** mit vorgefertigten Funktionen und Abläufen

64 Globale Funktionen **GF1-GF64** gleich wie die Spezialfunktion aber für alle Modelle

32 Kurven (**KV1-KV32**) mit 3-17 Stützpunkte in X und Y frei definierbar

9 Globale Variablen (**GV1-GV9**) für jede Flugphasen mit unterschiedlichen Werten

16 PPM Eingänge am DSC Trainer-Port (**TR1-TR16**)

Mehr Details dazu in den einzelnen Kapiteln, den Mischern und den Modelleinstellungen

Über das USB / PC Interface kann man:

Modelle hin und her übertragen werden, auf die SD-Karte zugreifen

Den Sender als PC-Joystick für einen Flugsimulator verwenden

Den Sender mit neuer Software geflasht werden

Ab opentxV2.0 ist dazu ein eigener Bootloader installiert.

OpenTx Funktionen Erweiterungen

Da **open9x/openTx** für Atmega64 mit begrenztem Speicher von 64K Flash und 2K EEPROM geschrieben wurde und um div Hardware- und Software-Optionen ergänzt werden kann, gibt es eine Vielzahl von Kombinationen aus Hardware-Erweiterungen und Software-Möglichkeiten die man zusammenstellen kann.

Im Programm **Companion9x** kann man die verschiedenen Sender auswählen und sieht dann die verschiedenen Softwareoptionen die man dort sehr einfach und komfortabel zusammenstellen kann. Hier ein Überblick über die bis jetzt vorhandenen Funktionen, die unter Companion9x angezeigt und ausgewählt werden.

Hardware Modifikationen und Anpassungen:

(im Taranis Sender ist davon fast alles schon enthalten!)

1. **audio** – damit wird anstatt dem eingebauten schrecklichen Summer ein kleiner Lautsprecher angesteuert. Das ist mit einer einfachen Hardwaremodifikation möglich. Der Lautsprecher spielt dann div. Melodien. Mit dieser Option wird das dann gesteuert.
2. **haptic** – mit dieser einfachen Hardware-Erweiterung wird ein kleiner Vibratormotor angesteuert der dann parallel zum Summer/Lautsprecher vibriert.
3. **frsky** – damit wird das FrSKY HF-Modul für Telemetrie mit dem Sender verbunden. Das ist etwas aufwändiger einzubauen, aber es ermöglicht die FrSky Telemetriedaten direkt am Display darzustellen ohne zusätzliche Telemetriebox. Alle Telemetriesysteme benötigen Hardwaremodifikationen am Sender.
4. **PXX** – Ein neues serielles Übertragungsprotokoll der Fa. FrSKY
5. **jeti** – verbindet ein Jeti-Telemetriemodul mit dem Sender
6. **ardupilot** – empfängt Daten vom Modul ArduPilot
7. **voice** – für Sprachansagen mit einem Synthesizermodul und SD-Karte
8. **DSM2** – steuert ein DSM2 Modul von Spektrum
9. **SP22** – Smartie Parts 2.2 Ist eine Huckepack/Adapterplatine für einfaches Programmieren/Flashen und für die Hintergrundbeleuchtung

Softwaremodule zusammenstellen:

(im Taranis Sender ist davon fast alles schon enthalten!)

1. **heli** – für Helikopter, die benötigten Grundfunktionen auswählen
2. **nosplash** – kein Startbildschirm anzeigen
3. **nofp** – keine Flugphasen verwenden
4. **nocurves** – keine Kurven verwenden
5. **ppmca** – Darstellung der Signalmitte (1500) in μs im Limitmenü statt +/- 100%
6. **ppm μs** – Darstellung **aller** Kanal-Impulsbreite in μs anstatt in %. Im Hauptmenü und im Servomonitor 980 μs bis 2020 μs , im Limitsmenü -512 (= -100%) + 512 (= +100%)
7. **potscroll** – Potentiometer für das Scrolling durch die Menüs aktivieren
8. **autoswitch** – Schalter können im Setup Menu beim Betätigen automatisch erkannt werden, ein Betätigen macht sie kenntlich normal und als “!” invers
9. **nographic** – keine grafischen Check-Box
10. **nobold** – keine fette Darstellung von aktiven Elementen
11. **pgbar** – ein kleiner Balken zeigt an wenn Daten abgespeichert werden
12. **imperial** – Anzeigewerte in Zoll statt Metrisch
13. **gvars** – globale Variablen verwenden/aktivieren

Eine aktuelle Liste mit allen möglichen Optionen findet man unter
OpenTx Projekt Wiki : <http://code.google.com/p/open9x/wiki/CompilationOptions>

In der Beschreibung steht dann (**if the option xxxx is chosen**) für Funktionen die nur dann vorhanden sind wenn diese Option auch ausgewählt wurde.

Die Beschreibung von **open9x/openTx** ist für das normale Standard Board mit ATmega 64 Prozessor Es gibt aber auch folgende Boards mit angepasstem, erweitertem Funktionsumfang der Software

1. **STD** TH9 das normale Standard Board mit ATmega 64
2. **STD128** TH9 128 Board wie Standard aber mit ATmega 128 (doppelt so viel Speicher)
3. **Gruvin9X** Board mit Atmega 2560 und sehr vielen Erweiterungen
3. **ERSKY9X** Board mit ARM cortex M3 32bit und sehr vielen Erweiterungen
4. **9XR** Neuer Sender, fast baugleich zu STD, von Hobbyking
5. **9XR 128** mit Atmega 128 prozessor
6. **9XR-Pro** mit 32 Bit Prozessor und neuer Hauptplatine mit vielen Funktionen
7. **TARANIS** X9D FrSky-Sender mit opentx-Software
8. **TARANIS Plus** X9DPlus Hardware-Erweiterungen und Verbesserungen
6. **X9E** Pultsender von Frsky mit openTx, ab Mitte 2015
7. **Horus** Neuentwicklung von Frsky mit Farbdisply, kommt Ende 2015

Die open9x/openTx – Software läuft ohne Änderungen auf den Sendern, Turnigy TH9x und 9XR, da sie fast baugleich sind.

Bei openTx für Taranis ist das anders.

Hier sind praktisch alle Hardware-Erweiterungen auf der Platine schon enthalten und damit auch fast alle Softwareoptionen schon integriert!

Es gibt nur noch 2-3 zusätzliche Funktionen, bzw ein paar Funktionen zum abwählen
ppmµs – Darstellung **aller** Kanal-Impulsbreite in µs anstatt in %. Im Hauptmenü und im Servomonitor 980µs bis 2020µs, im Limitsmenü -512 (= -100%) + 512 (= +100%)

LUA - für die Script-Programmiersprache Für die Sender X9D, X9D Plus, X9E

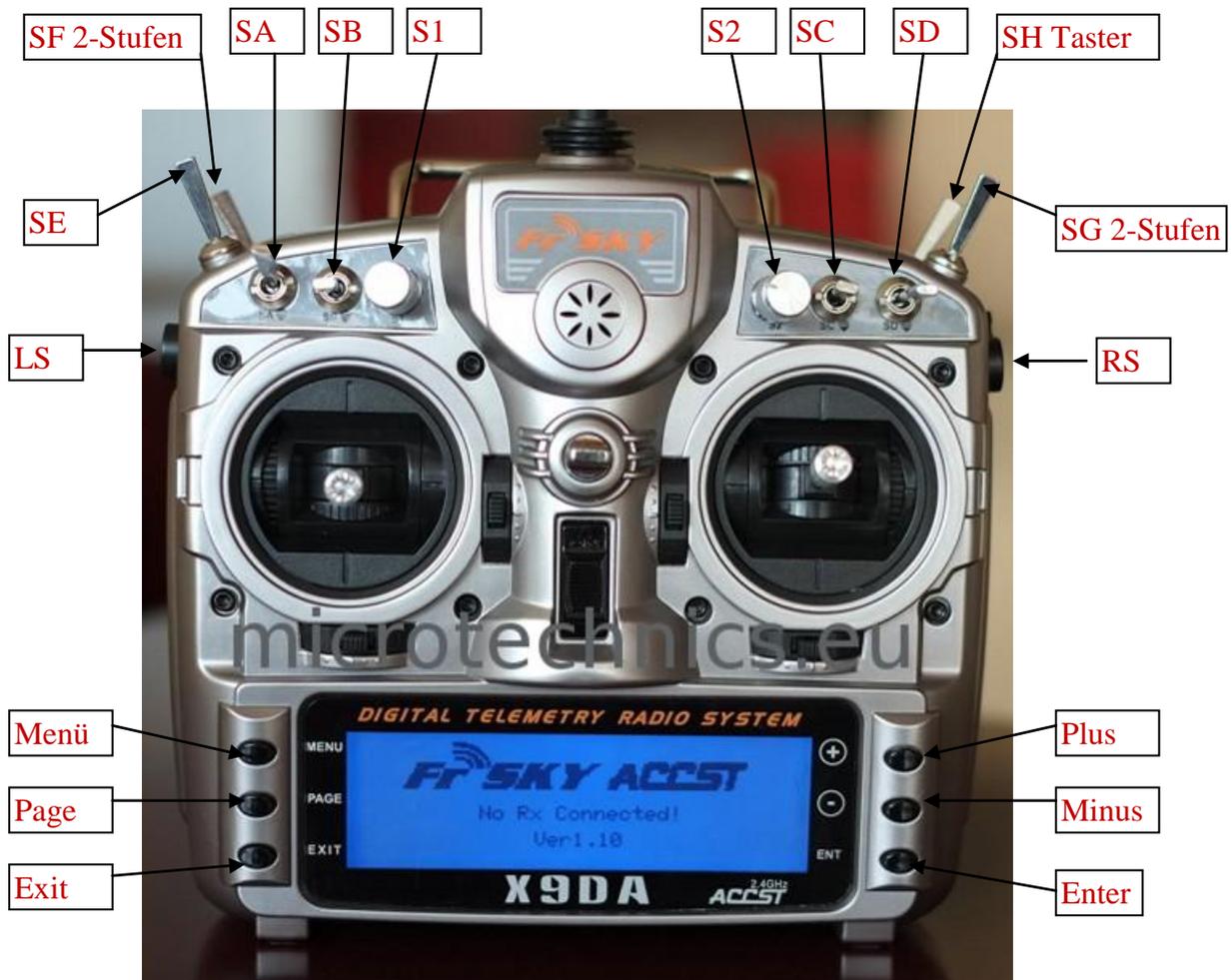
Haptik - für das Vibratormodul

Das liegt daran dass die Software nochmal erheblich erweitert wurde,
ein 32bit Prozessor und 512 kB Flashspeicher zur Verfügung steht
das Handling und die Anzeigen im Display etwas anders aufgebaut sind und
die Hardware im Sender sich erheblich von den einfachen Th9x-Sendern unterscheiden.
OpenTx für Taranis wird ständig weiterentwickelt und mit Funktionen ergänzt.

Dieses Handbuch beschreibt den Sender FrSky Taranis, Taranis Plus und seine

openTx für Taranis Anleitung Deutsch
Softwarefunktionen, aber auch für die Sender Th9, 9XR, 9XR-Pro.

Sender FrSky Taranis Bedienelemente



Jeder 3-Stufenschalter kann auch einfach gegen einen 2-Stufenschalter ausgetauscht werden
(machem sind das zu viele 3-Stufenschalter)

Ansicht Rückseite mit Schacht für externes HF Modul im JR-Format

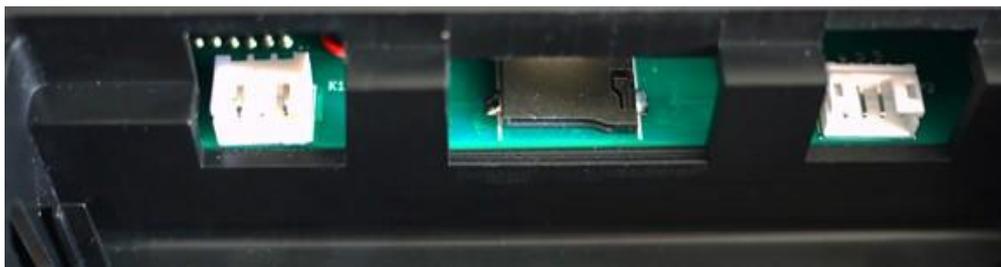


Verschiedene zusätzliche HF-Module möglich

Akkuanschluss JST-XH

micro SD-Karte

Serielle Schnittstelle JST-PH



Trainer DSC-Buchse

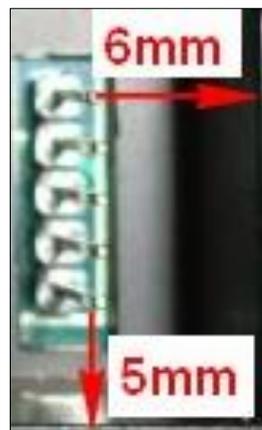
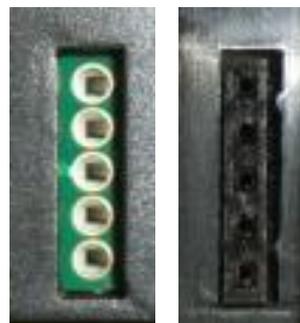
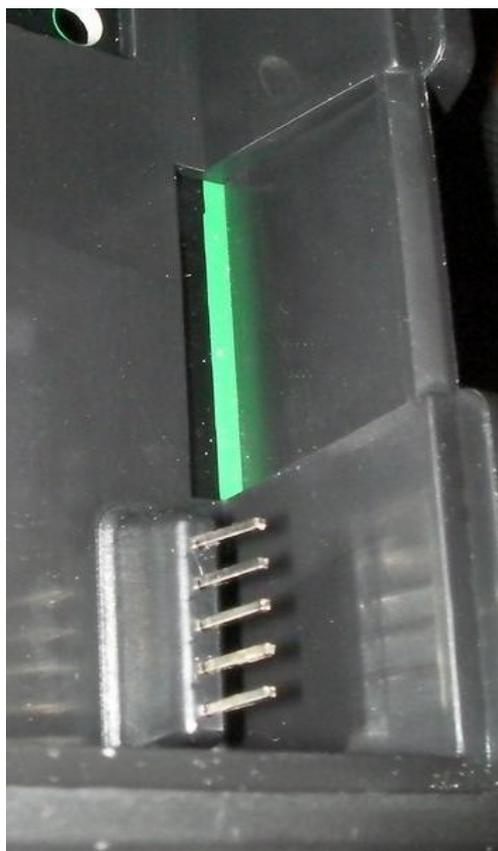
USB-Anschluss

Kopfhörer



Taranis Pinbelegung externer Schacht, Maße im JR-Format

Buchsen von 2 versch. Externen Modulen im Vergleich gut schlecht



- CPMM** Output: für passende HF-Module **CPPM, PXX, DSM2**
- Heart_Beat** Input: weitere Möglichkeiten für Inputs **S-Bus, CPPM (Lehrer S-Bus)**
- VMAIN** Akkuspannung unreguliert 6-9V (nach F1, D5, Hauptschalter)
- GND** Signal-Masse
- SPORT** Output: **S-Port-Signal** für Update am S-Port, XJT, Empfänger, Sensoren

Einfache externe Module haben nur Buchsenleisten, dort passen die Stifte oft nicht sauber rein, das Module klemmt, lässt sich nicht leicht einsetzen, deshalb kann es sein dass man die Stifte ausrichten muss (6mm und 5mm).

Hochwertige Module haben runde Zentrier-Buchsen, gehen leicht rein. Falls es extrem klemmt kann man auch die interne Platine ausrichten. Nicht mit Gewalt reindrücken!

Externes HF-Modul Grundeinstellungen für übliche Summensignale (CPPM-Werte)

Frsky /Graupner/Futaba/Multiplex: 4-8 Kanäle 22,5ms 300us + (pos Startflanke)

Spektrum/Orange DSM2/DSMx: 4-8 Kanäle 22,5ms 400us - (neg Startflanke)

Test und Alternativ: 4-8 Kanäle 18 - 27,0ms 100 - 400us + oder -

Graupner PPM 18 für 9 Kanäle
9 Kanal 22-24ms 300us +

Graupner PPM24 für 12 Kanäle (andere PPM min und max)
12 Kanal 28-30ms 300us +

Das externes XJT-Modul

Falls das interne XJT-Modul nicht mehr ausreicht kann man ein zusätzliches XJT-Modul stecken. Das kann dann nochmal bis zu 16 Kanäle übertragen.

Als externes XJT-Modul erkennt es an seiner **CPPM-Buchse** automatisch wie es angesteuert wird, vom PXX-Protokoll oder von CPPM-Signalen
Bei einem passenden CPPM-Signal erkennt es auch die Anzahl der Kanäle automatisch.

Es kann angesteuert werden:

1. Von der Taranis:

Im Fr-Sky D8 PXX-Digitalmode für alle D-Empfänger max 8 Kanal alle 18ms
Im Fr-Syk D16 PXX- Digitalmode für alle X-Empfänger max 16 Kanal alle 9ms
Mit einem CPPM - Signal für 4-16 Kanälen mit entsprechendem Protokoll
(das macht allerdings keinen Sinn)

2. Von einem Fremdsender:

Mit einem CPPM - Signal von 4-16 Kanälen mit entsprechendem PPM Zeitrahmen
Siehe dazu Erklärungen zum Aufbau eines PPM-Signals.
z.B. Turnigy Th9x, 9XR, 9XR Pro, alle Sender mit einem CPPM-Signal an das Hf-Modul
auch Graupner Modulsender im Mode PPM18 für 9 Kanäle PPM24 für 12 Kanäle

3. Von einem anderen Empfänger oder von einem Arduino Board

Wenn dieser ein CPPM-Signal (=Summensignal) ausgibt, als Relayfunktion,
oder auch für eine Joysticksteuerung.

Binden:

Es kann von der Taranis aus direkt gebunden werden, wie das interne XJT Modul auch.
wenn es im D8 oder D16 / X16 Mode angesteuert wird.

Oder:

Per Binde-Knopf am Modul wenn es per CPPM-Signal angesteuert wird.
Dann muss allerdings per Dipschalter der Zielempfänger-Typ D8, D16, eingestellt sein.



Das XJT-Modul hat auch auf der Rückseite zusätzlich die 2 doppelreihigen Pfostenbuchsen in 2mm Raster, passend für moderne JR-Sender und Graupner MX24s

Akkuanschluss

6 Zellen NiMH 1,2V = 7,2V 2000mAh mit geringer Selbstentladung (Eneloop-Typ)
Akkufach: 108x31x23mm Akku: JST-XH Stecker
Es ist Platz für 8 Zellen Mignon

Stromverbrauch ca 150-180mA (ohne Sound)
Eingebaute Akku-Ladeschaltung für NiMH
Somit nicht zum Laden von Lipo's geeignet!

Mit Netzteil AC 220V DC 12V 500mA
Ladebuchse Hohlstecker 5,5x2,5mm



Steckerbelegung



Micro-SD-Karte ca 1-2GB

Micro SD-Karte formatiert mit FAT12, FAT16 oder FAT32 (1GB reicht völlig aus)
mit mindestens 8-10 Unterverzeichnissen, meist gibt es noch mehr. Videos, Dokus usw.

→ **Dazu gibt es eine eigene Seite siehe Sender Grundeinstellungen 2/8**

Ansagetexte in Deutsch und Töne gibt es hier:

<http://85.18.253.250/voices/opentx-taranis/de/> als katrin.zip

Diese Texte kann man auch ganz einfach selber machen, anpassen, erweitern, siehe Teil C
Achtung: Nur 7-8 Zeichen als Dateiname zulässig, eventl. umbenennen!

USB Mini Buchse

Die Taranis meldet sich am PC mit 2 Wechseldaten-Laufwerken an meist E: F: oder F: G:
E: die SD-Karte mit allen obigen Unterverzeichnissen und allen weiteren Dateien
F: das EEPROM da stehen 1 oder 2 *.bin Dateien drinnen, Finger weg! nicht anfassen!

Lehrer-DSC Buchse 3,5mm Mono

Für PPM Signale als Eingang oder Ausgang
Lehrer / Schüler und Simulator-Anschluss
Steckerbelegung: Spitze ist Signal, Ring ist Masse
(ein Stereostecker geht auch!)



Kopfhörer 3,5mm Stereo

Texte, Warnungen, Ansagen, Telemetrie, Vario, Klänge und Töne
die per Telemetrie, Funktionsschalter oder Zustände aufgerufen werden.

Sender Akku laden 6 Zellen NiMH (Eneloop-Typ)

Der Sender hat ein eingebautes Ladegerät für NiMH

Der beiliegende Akku hat 6 Zellen, 2000mAh, Stromverbrauch 210-230mA (ohne Sound)

Das **beiliegende Steckernetzteil 220V AC** liefert **12V DC Festspannung und 400mA**

Man kann auch ein anderes Steckernetzteil nehmen, das eine geregelte 12V Festspannung liefert. Oder am Zigarettenanzünder der 12V Autobatterie laden.

An der rechten Seite ist die Ladebuchse, ein Hohlstecker mit 5,5x2,5mm

Belegungen Plus = Innen Minus =Außen



Akkustecker: JST-XH am 6 Zellen NiMH-Akkusatz, Nennspannung 7,2V

Rechts unten ist auch die grüne Lade-LED.

Beim Ladestart blinkt sie kurz, dann schaltet sie auf Dauerlicht

Solange die LED leuchtet wird geladen, ist sie aus, ist der Akku voll.

Man darf auf gar keinen Fall ein Akkuladegerät an der Ladebuchse anstecken, den das versucht den Akku zu ermitteln und erhöht dabei die Spannung auf bis zu 45V, damit wird die Elektronik des Senders zerstört!

Man darf auf gar keinen Fall einen Lipo- oder LiFe-Akku über das eingebaute Ladegerät des Senders laden!

Ein moderner NiMH Akkusatz hat eine sehr geringe Selbstentladung z.B. Sanyo Eneloop (der Begriff Eneloop ist geschützt, darum verwendet jeder Akkuhersteller eine andere Bezeichnung für diese Art der Zellen mit sehr geringer Selbstentladung).

Sender mit 2,4GHz haben nur noch einen sehr geringen Stromverbrauch, ca. 100-200mA, so dass ein Akkusatz mit 2100mAh locker 10Std hält.

Außerdem reicht eine Akku-Nennspannung von ca. 7,2V völlig aus, da der Prozessor mit 3,3V versorgt wird (6 Zellen NiMH Nennspannung $6 \cdot 1,2 = 7,2V$)

NiMH Akkus sind vollgeladen mit ca. 1,27V/Zelle ($6 \cdot 1,27 = 7,62V$) und leer mit ca. 1,1V/Zelle ($6 \cdot 1,1 = 6,6V$) Bei einem 6 Zellen NiMH Akkusatz stellt man deshalb die Warnschelle für Akku leer am Sender auf ca. 6,8V ein. Sender Systemeinstellungen 1/6, Akku leer unter: 6,6V

**Der X9E Sender hat 8 Zellen NiMH Typ Eneloop mit angepasstem internem Ladegerät
Voll geladen $8 \cdot 1,27V = 10,2V$ Fast leer $8 \cdot 1,1V = 8,8V$ Akku leer auf 8,8V einstellen**

Andere Zellenspannungen:

Lipoly Zellenspannung:

Nennspannung = 3,7V 2S= 7,4V

Ladeschlussspannung = 4,2V 2S= 8,4V

Entladeschlussspannung = 3,0V 2S= 6,0V

LiFe Zellenspannung:

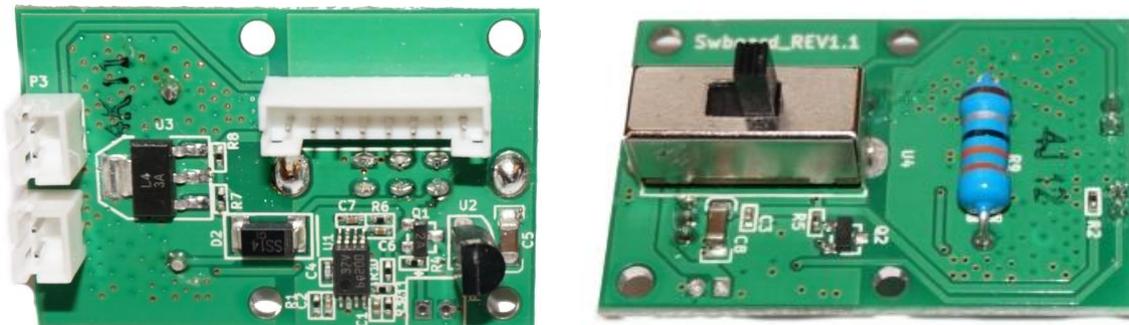
etwa 3,2-3,3 V 2S= 6,4 - 6,6V

etwa 3,6-3,65 V 2S= 7,2-7,3V

etwa 2,5 V. 2S= 5V

Hinweis zu den Versionen A01 und B01 der Taranis:

Die Taranis hat einen eingebauten **Ladecontroller auf der Hauptschalter- Platine**. Dieser ist ausgelegt für 6 Zellen NiMH, hat Delta Peak und hat auch eine Zeitüberwachung. Je nach Taranis-Version ist sie anders bestückt **A01 für 800mAh** ab **B01 für 2000mAh**. Wenn man den Akku hochrüstet von 800mAh auf 2000mAh sollte man auch die Ladeplatine tauschen. Die gibt es recht günstig ca 6€ bei microtechnics, Belgien.



Oder man muss dann mehrfach den Ladezyklus neu starten, 2-3 mal
Oder man klemmt die interne Ladeschaltung ab, und lädt mit einem externen Ladegerät direkt.
Dann aber auch die Ladebuchse kennzeichnen, damit klar ist was gemacht wurde!

Umrüsten auf Lipo-Akku mit 2 Zellen 2S

Es gibt Lipo und LiFe Akkusätze für Sender mit 3 oder 4 Anschlusssteckern: Ladeanschluss, Balanceranschluss und Senderanschluss. Damit kann man den Akku mit einem externen modernen Lipo-Ladegerät laden ohne den Akku-Stecker im Sender abzuziehen zu müssen.

ABER: !! Polarität von diesen Akkusätzen beachten, eventl. drehen!

Powerschalter am Sender immer auf AUS! und nicht mehr an der seitlichen Ladebuchse laden, am Besten dann die Ladebuchse intern ausstecken!

Aufpassen muss man am Anschluss des Akkus am Sender. Wenn man einen Original-Stecker JST-XH verwendet und den Akku richtig anlötet kann nichts passieren. Also markieren, dreimal überlegen und kontrollieren, sonst gib der Sender Rauchzeichen, das wars dann!

Buchsenbelegung JST-XH am Sender:
Rot = Plus = Links an der Buchse
Schwarz = Minus = rechts an der Buchse



Beispiel: Umbau auf einen anderen Akku

Der Stromverbrauch bei 6 Zellen, 2000mAh NiMH beträgt 210-230mA (ohne Sound)

6 Zellen NiMH Akku vollgeladen, dann ist die Akkuspannung (ohne Last) ca 7,7V

Akku Kalibrierung auf 7,6 -7,7V einstellen

6 Zellen Akku leer bei ca 6,6V Alarmschwelle einstellen auf ca 6,7V

Akku Anzeigebalken am Sender 6,6-8,0V einstellen

Alarm einstellen auf 6,7V

Sender läuft auch noch bei 5,8V, dann aber 240mA Stromverbrauch

In den Akkuschacht passen auch 8 Zellen Mignon, z.B Eneloop 2100mAh,
dann den Schaumstoff im Deckel entfernen.

Aber das bringt nichts da die interne Ladeschaltung für 6 Zellen ausgelegt ist!

Auch 2-3 Zellen Lipoly/LiFe sind möglich

12V ist keine Problem, max Obergrenze ist 15V für den Schaltregler!

Wer das interne Ladegerät nicht will/braucht muss an der Ladebuchse umlöten,
so dass der Akku direkt drann hängt, dann aber mit externem gutem Ladegerät laden!

Verpolungsschutz macht das externe Ladegerät, deshalb keine Verpolungsschutz-Diode,
denn das externe Ladegerät muss ja auch rückwärts die Akkuspannung messen können.

Interne NiMH-Ladeschaltug umgehen und Akku direkt laden

Das bringt nur was wenn man den Akku immer extern laden will

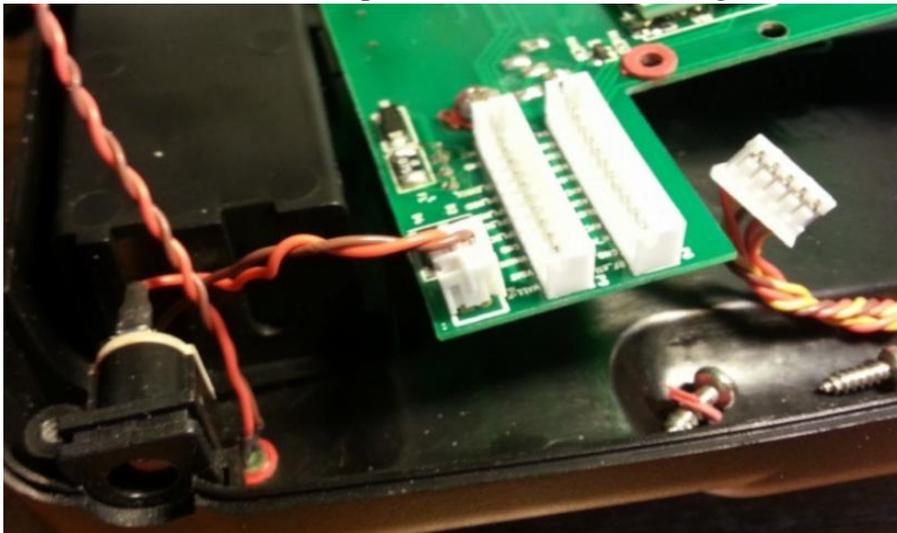
Vorsicht bei diesem Umbau, auf eigenes Risiko!

Selbst bei Power OFF liegt noch Spannung an der internen Ladeelektronik

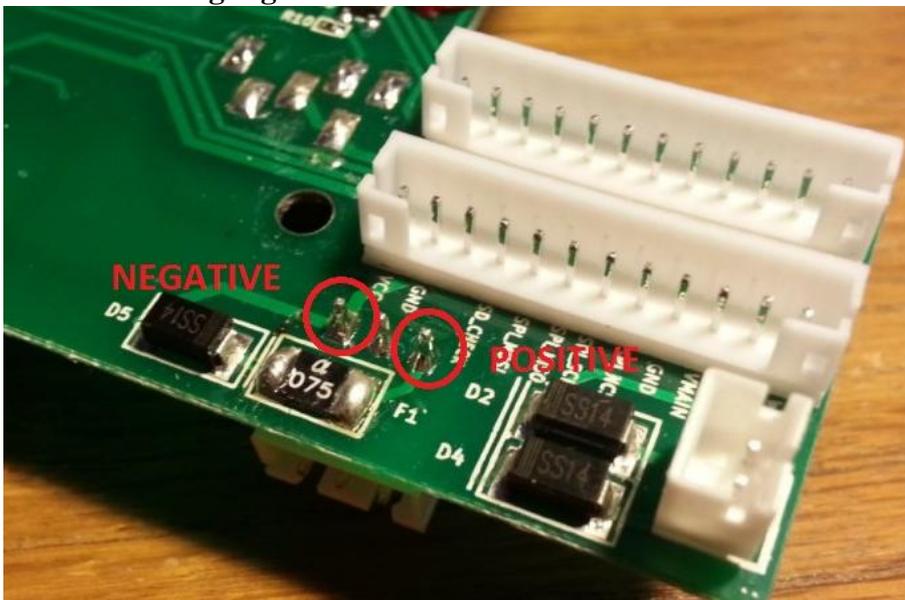
Nur für 6 Zellen NiMH Akku möglich, keine 8 Zellen NiMH, keine Lipo !

Laden nur mit hochwertigem programmierbarem Ladegerät!

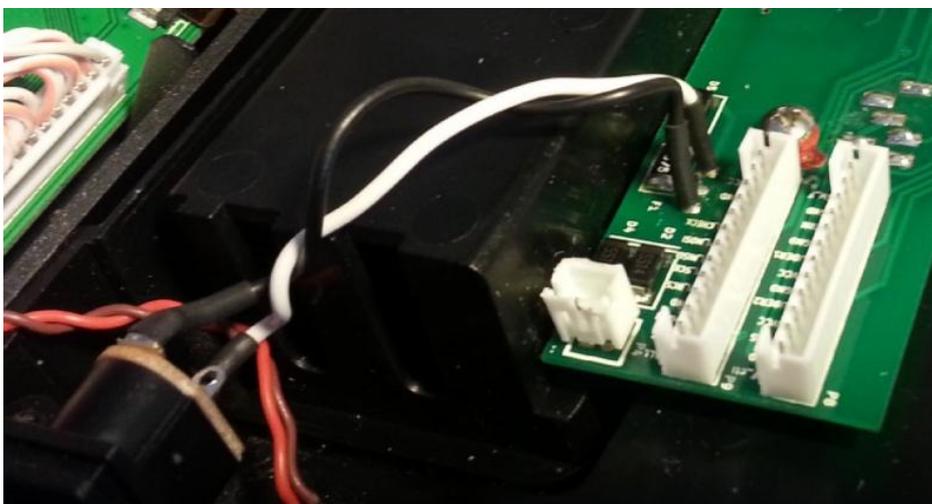
So sieht das Original aus: Ladebuchse und Stecker auf der Platine



Das ist die Belegung des Akkus selbst

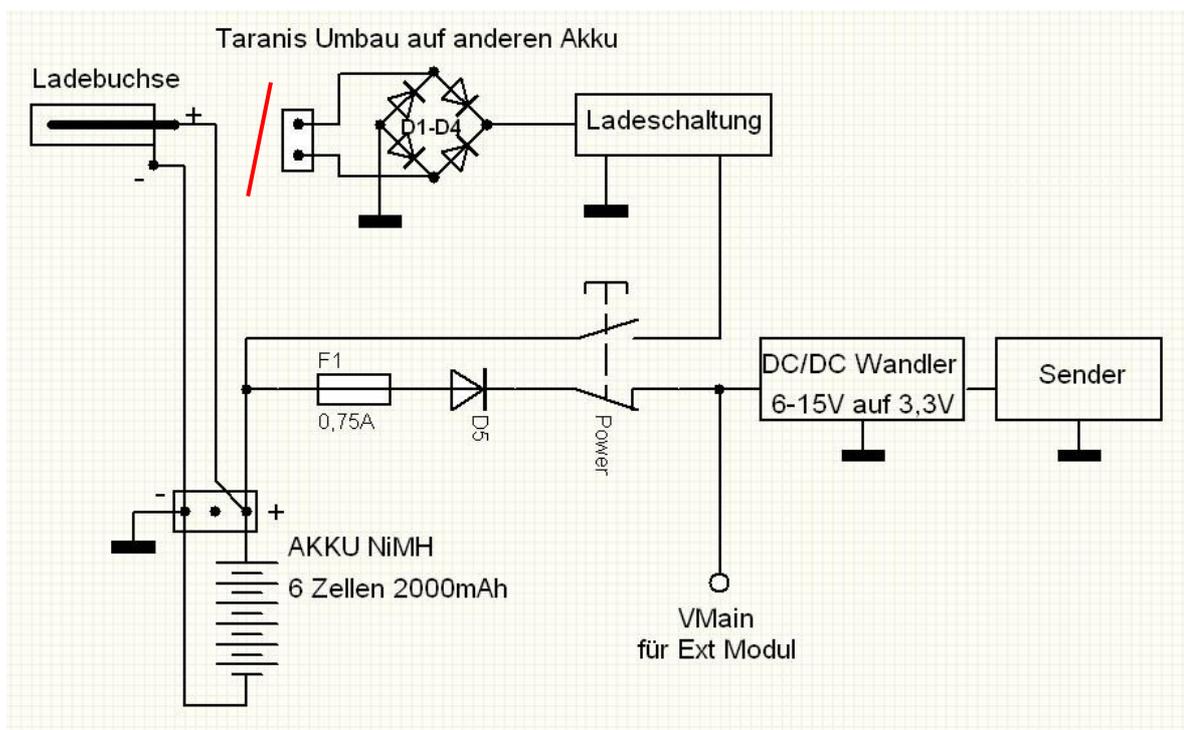


Und so nach Umbau: Ladebuchse direkt auf die Pins des Akkus gelötet





Prinzip des Umbaus: Von der Ladebuchse direkt auf die Pins des Akkusteckers



**Geladen wird immer mit Power Off (wie bei allen anderen Sendern auch)
Externes Ladegerät richtig einstellen: Akkutyp, Ladestrom, Zellenzahl!**

Knüppelaggregate X9D umstellen von Mode 1 auf Mode 2 (Mode 1, Mode 3 = Gas rechts, Mode 2, Mode 4 = Gas links)

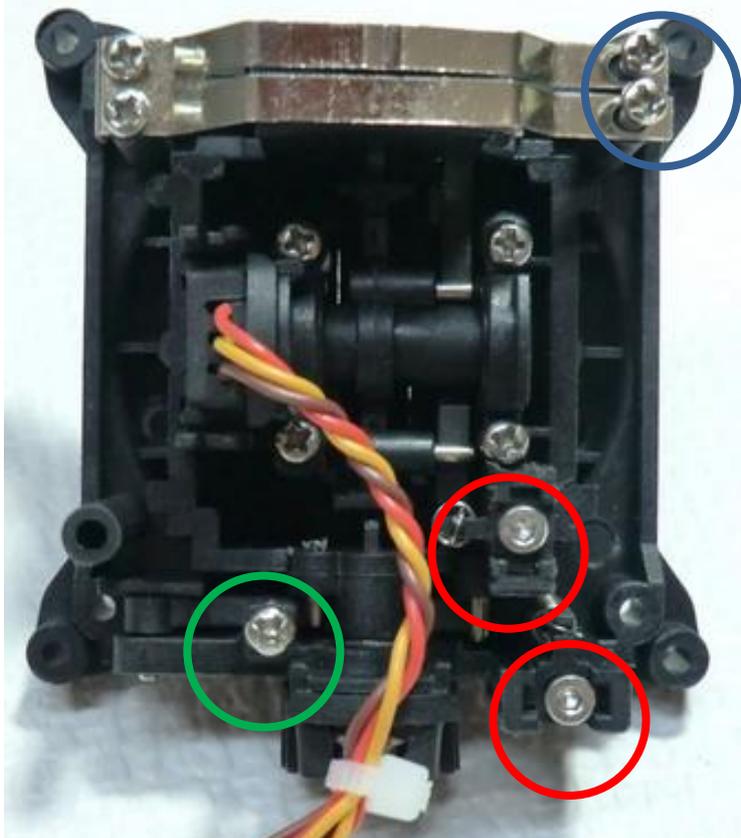
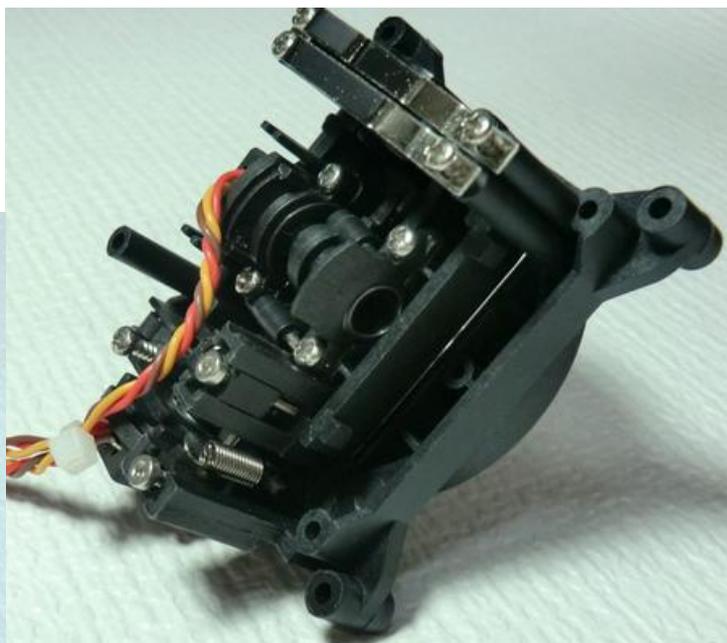
Dazu muss das Gehäuse geöffnet werden. 6 Schrauben auf der Rückseite.

Für den Umbau muss man keine Federn oder Hebel ausbauen und am anderen Aggregat wieder irgendwie reinbauen. Das geht ganz einfach.

Nur die entsprechende Schrauben lösen bzw anziehen, und schon hat man die Gas-Funktionen von rechts (Mode 1) auf links (Mode 2) umgebaut.

Die Rastfunktion oder die Knüppeldämpfung kann man auch getrennt für jede Achse einzeln einstellen.

Sehr hochwertiges Knüppelaggregat:



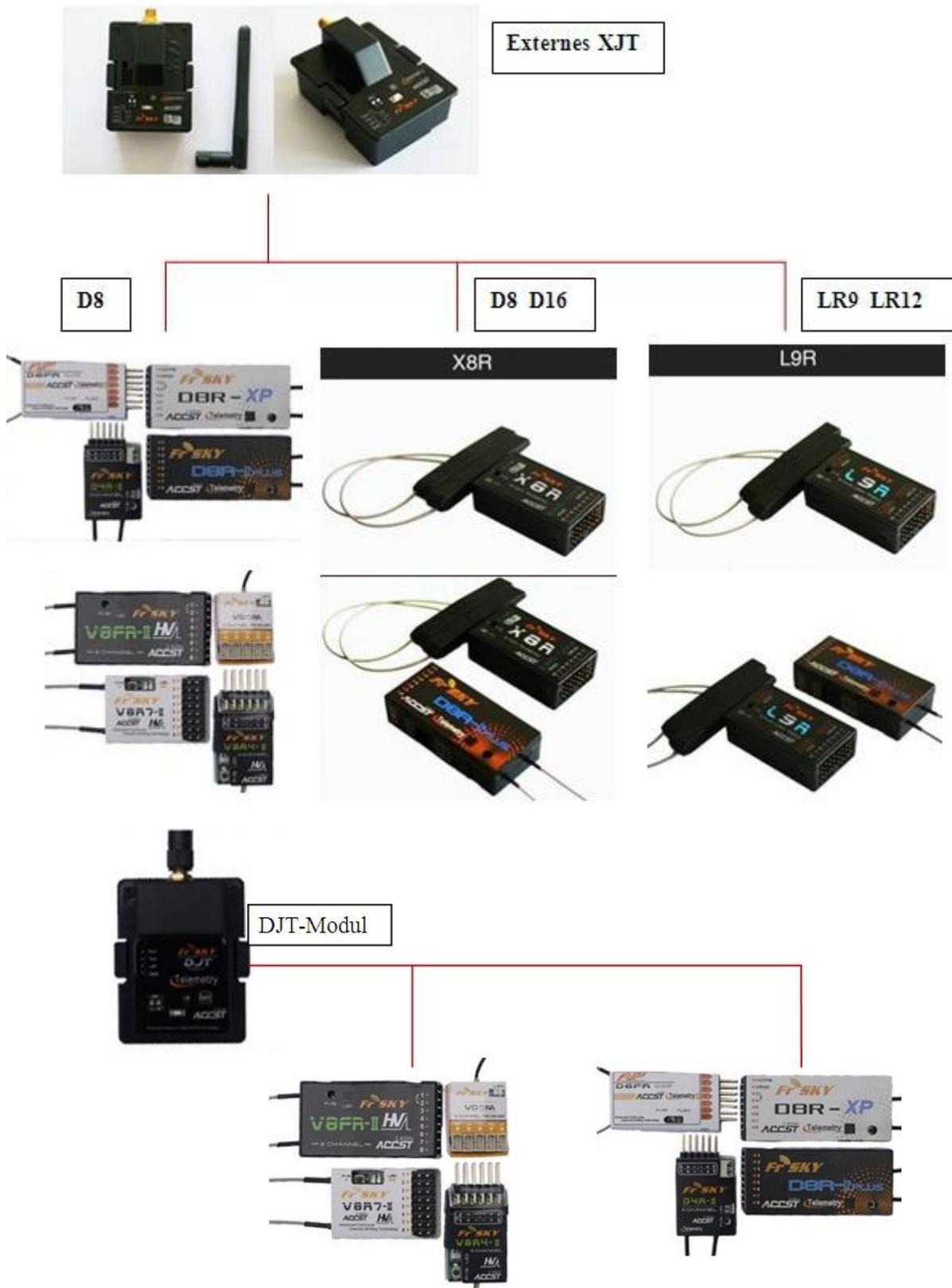
Mode umstellen
Y-Achse frei beweglich machen

Blau = Bremse und Rasterung
einstellen in der Y-Achse

Rot = Federkraft für X und Y
einstellen, untere ist für Y

Grün = Y-Achse Feder abheben für
freie Y-Achse, Mode umstellen

Empfängerkombinationen Externes XJT oder DJT Modul



Das XJT-Modul (intern oder extern) kann 3 Betriebsarten, D8-, D16-, LR -Mode

XJT im D8 Mode für die D und V-II Empfänger

XJT im D16 für die X-Empfänger

→V8 Empfänger nur in D8 Modus, kein Failsafe, kein Modellmatch, Binden mit Jumper

X8R-Empfänger mit Smart-Port und SBus jumpern (gilt auch für X6R, X4R)

Am Empfänger X8R gibt es 2 serielle Anschlüsse, bitte nicht verwechseln!

S. Port bzw Smart-Port, **hinten** bei den 2 Antennen, um die **FrSky-Telemetrie anzuschließen**
 S-Bus, **vorne** bei den Servosteckern um S-Bus Baugruppen anzuschließen

Der SBus ist eine spezielle serielle Schnittstelle. Diese Schnittstelle wurde von Futaba entwickelt um Servos anschließen zu können, die im Modell weit verteilt sind. Dazu werden allerdings spezielle Servos und Signalverteiler (HUB) benötigt. Auch div Flugcontroller (NAZA) können den SBus verwenden. Damit wird nur 1 Servokabel als Anschluss benötigt.



Smart-Port für Telemetrie

RSSI und SBus

Jumper Beispiel Mode 4

Green LED	Red LED	Status
ON	Flashing	Binding
Flashing	OFF	Normal
OFF	Flashing	Signal Lost
Flash Twice	OFF	Failsafe Set

Receiver Mode and Binding Operation:

Mode of X8R	Telemetry	Channel Output	Receiver Mode select & Bind Operation	
			Jumped before Bind (signal pins)	F/S Button
Mode 1(D8)	√	CH1~CH8	CH7&CH8	connect the battery to any available channel output (no need to hold the F/S button on X8R)
Mode 2(D16)	×	CH1~CH8	CH3&CH4	connect the battery to any available channel output while holding the F/S button on X8R
Mode 3(D16)	×	CH9~CH16	CH1&CH2	
Mode 4(D16)	√	CH9~CH16	CH1&CH2, CH3&CH4	
Mode 5(D16)	√	CH1~CH8	No Jumper	

Der X8R kann so gejumpert werden (Mode = Betriebsart) dass er:

- Im **D8-Mode** für ein DJT-HF-Modul **oder** im **D16-Mode** für ein XJT-HF-Modul arbeitet
- Mit Telemetrie **oder** ohne Telemetrie-Übertragung arbeitet
- Kanal 1-8 **oder** Kanal 9-16 an den Servostecker rauskommen

Ohne Jumper ist der X8R im Mode 5, D16-Mode, mit Telemetrie, Servo-Kanal 1-8 vorbelegt. Die Jumper müssen vor dem Binden gesteckt sein und können danach wieder entfernt werden.

Der SBus gibt aber immer alle 16 Kanäle (1-16) raus!

Das **RSSI-Signal**, Empfänger Signal Stärke, kommt immer raus. Es ist keine Analogsignal, sondern ein PWM-Signal mit 3,3V und 100kHz, Pulsweitenmodulation 1-99%, das mit einem RC-Tiefpass (10kOhm 10uF) auch zu einem Analogsignal gewandelt werden kann.

Der X8R gibt leider kein CPPM Summensignal raus! (es gibt aber einen SBus to CPPM Wandler)

Die Beste Seite über Frsky-Baugruppen:

http://www.eflightwiki.com/eflightwiki/index.php?title=FrSky_Telemetry

SBus umwandeln in zusätzliche Servoausgänge oder in ein CPPM Summensignal

1. Möglichkeit: 16 Servos direkt anschließen

2 Stk X8R Empfänger verwenden, entsprechend vor dem Binden jumpern.

Jeder hat den gleichen Modell Match, nur einer darf Telemetrie übertragen

1. X8R auf Kanal 1-8, mit Telemetrie, 2. X8R auf Kanal 9-16, ohne Telemetrie

2. Möglichkeit: 16 Servos direkt anschließen

2 Stk SBus to PWM Decoder verwenden. Ein S-Bus to PWM Decoder kann 4 Servos ansteuern. Dazu muss ihm allerdings eingrogrammiert werden auf welche Kanäle des SBus er reagieren soll. Das macht der Servo Channel Changer, dort wird der PWM-Decoder als Servo angesteckt.

S-Bus Kanal 9-12 soll auf Kanal 1-4 des 1. S-Bus to PWM Decoders

S-Bus Kanal 13-16 soll auf Kanal 1-4 des 2. S-Bus to PWM Decoders

→ Jeden Kanal einzeln Schritt für Schritt einprogrammieren!

Am X8R-Empfänger werden dann beide PWM-Decoder mit einem V-Kabel am S-Bus angesteckt.

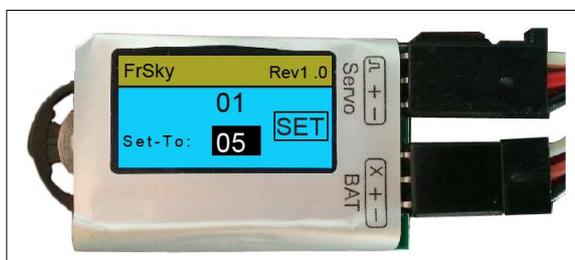
Siehe auch hier: <http://www.youtube.com/watch?v=bZ50z41hnmQ>

<http://www.youtube.com/watch?v=8T00Z1aQ6GM>



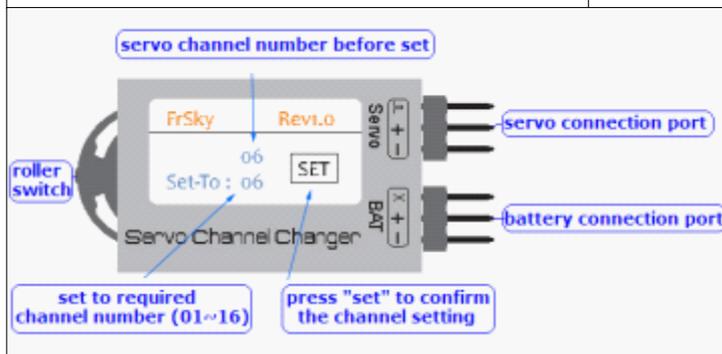
SBus to PWM Decoder für 4 Servos

Nur für Digitalservos, da 9ms Framezeit, alle 4 Kanäle kommen gleichzeitig somit Stromspitzen!



PWM Decoder

5V Akku



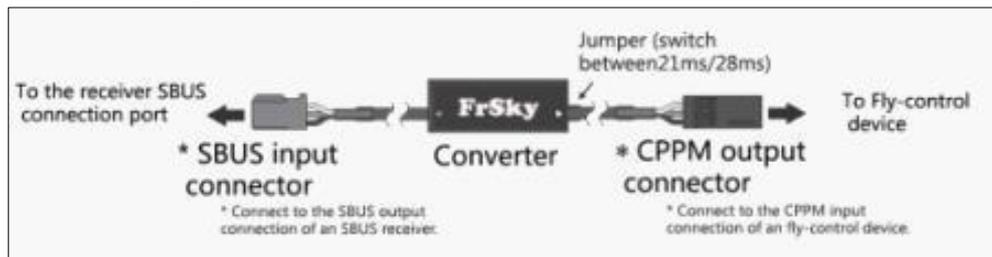
Servo-Channel Changer zum Programmieren

3. Möglichkeit: 8 Kanal CPPM Summensignal

SBus nach CPPM Decoder. Damit wird der SBus in ein CPPM Summensignal umgewandelt.

Aber nur Kanal 1-8 erscheint als Summensignal. Den X8R-Empfänger dann auf Kanal 9-16 jumpern

Das Timing kann von 21ms auf 28ms geändert werden.



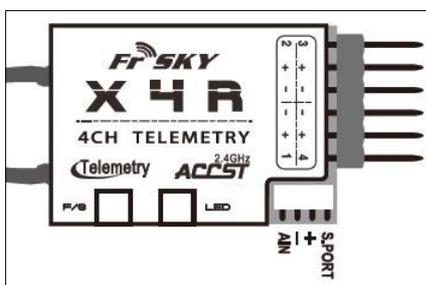
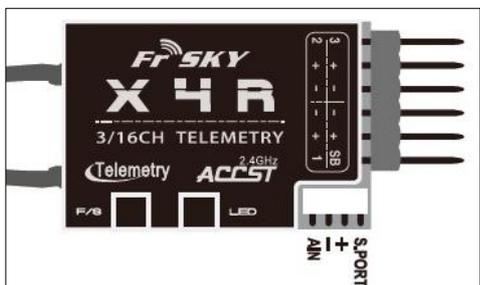
X4R und X4R-SB Telemetrie-Empfänger

Diesen sehr kleinen Empfänger gibt es in 2 Versionen.

X4R mit 4 Servo- Anschlüssen aber **ohne S-Bus** (weißes Gehäuse)

X4R-SB mit 3 Servo-Anschlüssen und **mit S-Bus** (schwarzes Gehäuse)

Beide haben den Anlogeingang AD2 (3,3V) und den FR-Sky Telemetrieanschluss **S-Port**
Mit neuestem Software-Update kann er auch 8 Kanal-CPPM mit 27ms ausgeben. (an K1)



Intern auf der Platine sind beide Empfänger aber exakt gleich aufgebaut!

Mal ist am CH4 Anschluss das 4.Servosignal rausgeführt, mal der S-Bus. (Servo-Bus)

Das ist jeweils nur über **eine** 0-Ohm Brücke auf Ober- **oder** Unterseite kodiert, R34 und R35.

0-Ohm Brücke **R34** legt auf den CH4 Anschluss das 4. Servosignal,

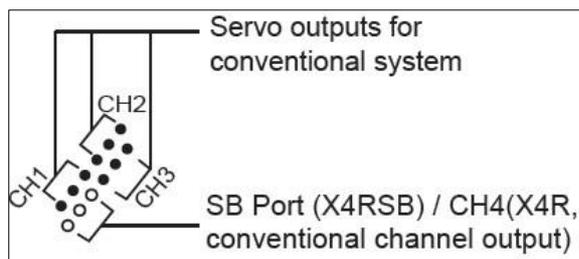
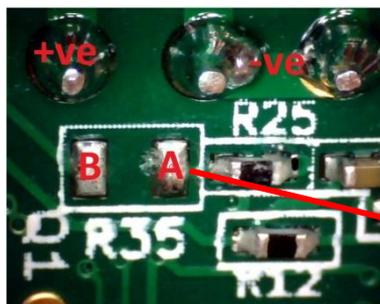
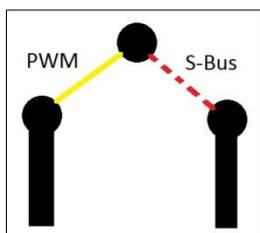
0-Ohm Brücke **R35** legt auf den CH4 Anschluss den S-Bus.

Somit kann man sich den Empfänger also selber anpassen!

Wer erst mal keinen S-Bus braucht sollte sich den weißen X4R- Empfänger besorgen.

Damit hat er gleich mal die 4 Servo-Anschlüsse (nur Brücke R34 ist belegt, R35 ist frei)

Man kann dann später an **Pin A** das **S-Bus Signal** mit einem Servokabel rauslegen.



6 Tasten Menüführung lang oder kurz drücken

Zusammenfassung der Tastenbedienung

Infozeile oben: Senderakku, Empfängerakku, SD-Karte, USB-Verbindung, Laustärke, Uhrzeit



Umschalten der Grundbildschirme:

[PAGE] 3 Grundbildschirme und Kanal-Monitor

[PAGE Long] 5 Telemetriebildschirme umschalten

Umschalten in Hauptmenüs und Untermenüs

[MENU Long] In die Sendereinstellungen 1/7

[MENU] In die Modelleinstellungen gehen 1/13

[PAGE] in den Menüs 1 Seite vorwärts

[PAGE Long] in den Menüs 1 Seiten zurück

[EXIT] Eine Eingabe, Zeile, Untermenü, zurück

[EXIT Long] Zurück in den Grundbildschirm

[ENT Long] Statistik anzeigen

mit **[+]** Debug aufrufen bzw.

im Kanalmonitor **[+]** **(1-16) (17-32)**

Eingaben machen:

[ENT Long] in die Untermenüs

Cursor **[+]** nach oben bzw links

Cursor **[-]** nach unten bzw rechts

[ENT] Eingabe, dann blinken

mit **[+]** **[-]** Werte eingeben

[ENT Long] Auswahlmenü erscheint

mit **[+]** **[-]** Edit/Kopieren/Verschieben

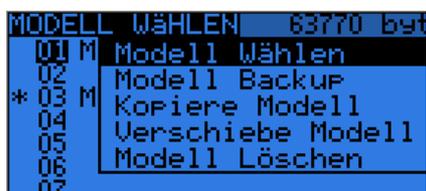


Oder: **[ENT Long]** zeigt situationsabhängige

Auswahlmenüs an oder schaltet Eingaben um

von **Zahlen nach Variablen** und

vereinfacht so das Handling erheblich.



Beim Eingeben von Werten kann man durch **gleichzeitiges** Drücken von 2 Tasten Werte ändern

[+] **[-]** Wert invertieren **[-]** **[ENT]** Wert +100

[EXIT] **[PAGE]** Wert -100 **[MENÜ]** **[PAGE]** Wert 0

Screenshotfunktion: (ab openTx V2.1)

Auf der SD-Karte das Verzeichnis: **/Screenshots** einrichten!

(Mit **[Exit] + [Enter]** zusammen drücken ?)

oder in den Spezialfunktionen per Schalter, Funktionsaufruf **Screenshot**

dann wird der aktuelle LCD-Screen auf der SD-Karte abgespeichert

Symbole als Auswahlliste vereinfachen die Eingaben (ab openTx V2.1)

Überall wo Quellen oder Schalter ausgewählt werden, z.B. Inputs, Mischer, kann man per **[ENTER Long]** den Auswahlliste schon vorselektieren, damit findet man die passende Quelle viel schneller.

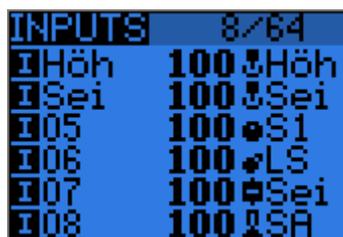


Je nach akt. Möglichkeiten erscheint eine unterschiedliche Auswahlliste



Auch schnelles Invertieren (!) eines Schalters oder Signals

SA → !SA → SA



In den Inputs und Mischern erkennt man dann auch sofort was gemeint ist und wo das Signal herkommt. Knüpfel, Potis, Schieber, Trimmung, Schalter, usw

Schalter und Potis Namen und Funktion zugewiesen (ab openTx V2.1)

Die Funktionen der Poti, Stufenschalter, 2-Pos, 3-Pos, Toggle können frei definiert werden, damit kann man beliebige Schalter verbauen.



Serielle Port kann mehrere verschiedene Funktionen übernehmen (ab openTx V2.1)

Im Akkuschacht ist der serielle Port (siehe Hardware) In den Sender Grundeinstellungen kann er jetzt als Eingang, Ausgang für div Funktionen verwendet werden.

8 Daten, 1 Stop, No Parity, keine Flußkontrolle. 3 Draht

- S-Port Mirror** 57600 Baud Ausgang Telemetrie vom S-Port hier ausgeben/durchreichen
- Telemetrie** 9600 Baud Eingang Telemetriedaten eines externen D-Moduls hier einspeisen
- S-Bus-Schüler** Ausgang Schülersender Werte im S-Bus Format ausgeben
- Debug-Modus** 115200Baud Ausgang Für Entwickler, Fehleranalyse



Trimmwerte anzeigen lassen (ab openTx V2.1)

Enschalten der Funktion bei den **MODELL-EINSTELLUNGEN** 2/13

Auswahl für die Anzeige: Nein, Ja, Kurz (nur bei Veränderungen)

Damit sieht man die hochauflösenden Trimmwerte in us



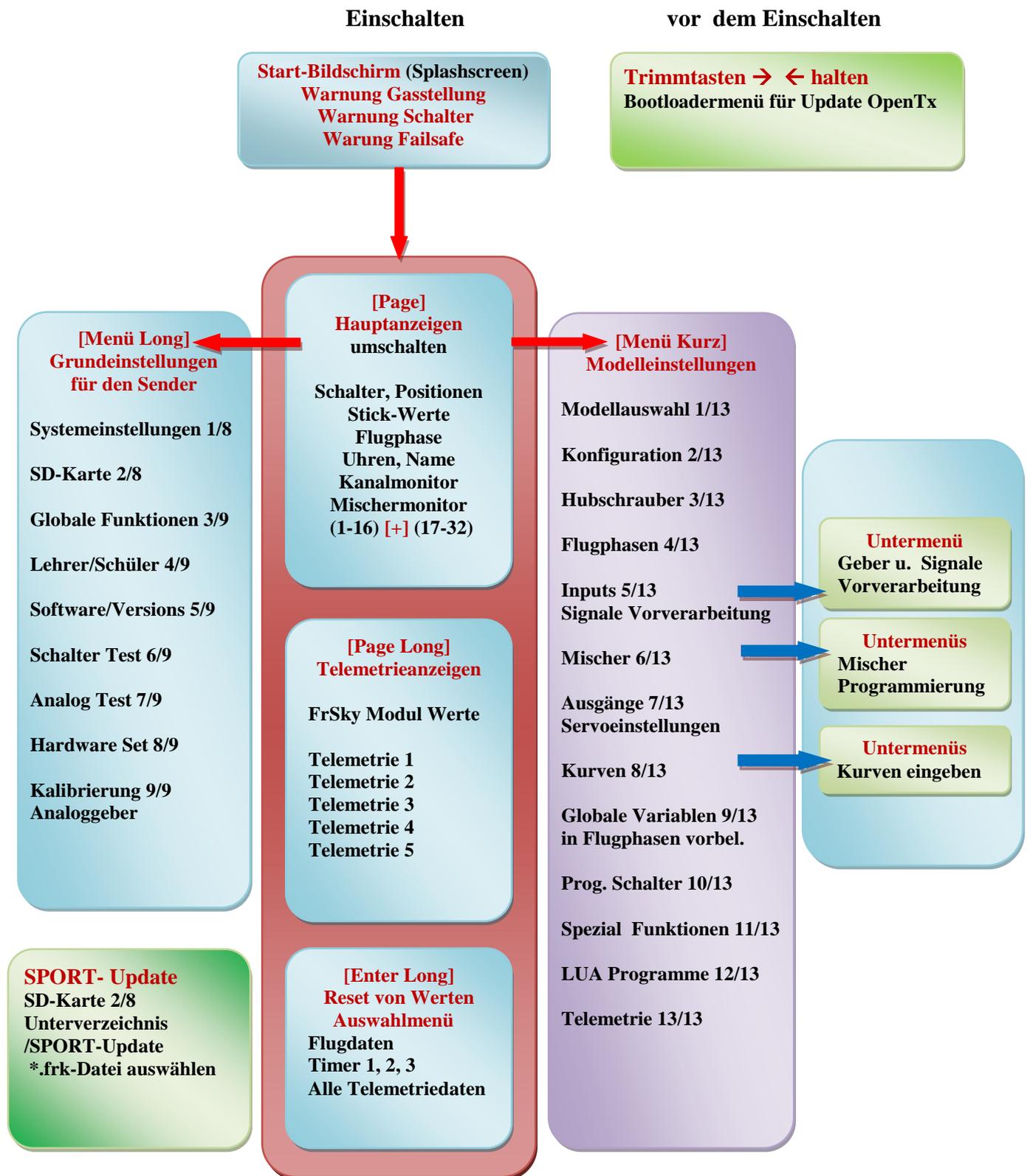
Das sind nicht %-Werte sondern die Trimmsschritte in us

Normaler Trimbereich +/-25% = +/-125 Schritte = 125us

Erweiterer Trimbereich +/-100% = +/-500 Schritte = 500us

Auflösung: Extra fein= 1us Fein = 2us Mittel = 4us Grob= 8 us

Softwarestruktur von OpenTx Frsky Taranis ab V2.017



Programmierprinzip OpenTX EVA Prinzip

EVA-Prinzip: Eingaben, Verarbeitung, Ausgaben

Egal wie umfangreich die Opentx schon ist oder noch wird, das Grundprinzip ist immer gleich: Es gibt keine Einschränkungen, alles ist mit allem direkt möglich. Für die Programmierung müssen wir uns immer nur 3 Fragen stellen. Das gilt für alle Eingaben egal ob Mischerzeilen, programmierbare Schalter, Spezial Funktionen, Geber, Servos, Kanäle, Telemetrie, Flugphasen,

Welche Funktion/Aktion/Reaktion will ich erzeugen, dazu brauche ich 3 Dinge:

- 1. Eingaben, Quelle: Wo kommt das Signal her, welche Signalquelle brauche ich**
- 2. Verarbeitung: Was will ich mit dem Signal tun, wie muss es verrechnet werden**
- 3. Ausgaben, Ziel: Wo soll das Signal was/wie bewirken, Kanäle, Servos, LS, SF**

Eingaben = Inputs Signale Vorverarbeitung berechnen und anpassen

Geber: Kreuzknüppel, Potis, Schalter, Kanäle, Telemetriewerte

Verrechnung: Quelle, Weight, Offset, Kurven, Funktionen, Trimmungen, Schalter, Dualrate,

Immer nur **eine** Inputzeilen aktiv, bzw per Schalter umschaltbar

Geber $\rightarrow ((\text{Quelle} * \text{Weight}) + \text{Offset}) + \text{Trim} \rightarrow \text{Kurve} \rightarrow \text{Mischer}$

Verarbeitung Kanäle = Mischer berechnen und anpassen

Quelle, Weight, Offset, GVARs, Kurven, Kanäle, Funktionen, Differenzierungen, Schalter, Flugphasen....

Mehrere Mischerzeilen können aktiv sein mit:

Addieren, Replace, Multiplizieren

Input $\rightarrow ((\text{Quelle} * \text{Weight}) + \text{Offset}) + \text{Trim} \rightarrow \text{Kurve} \rightarrow \text{Servos}$

Ausgabe Servos = Wege und Grenzen anpassen für die Mechanik

Subtrim, Min, Max, Inv, Kurve, PPM-Mitte, Sym,

Servos $\rightarrow \text{Kurve} \rightarrow (\text{Subtrim}, \text{Min}, \text{Max}, \text{Inv}) \rightarrow \text{Sender Baugruppe}$

2,4Ghz Sender Baugruppe

2 HF Module möglich

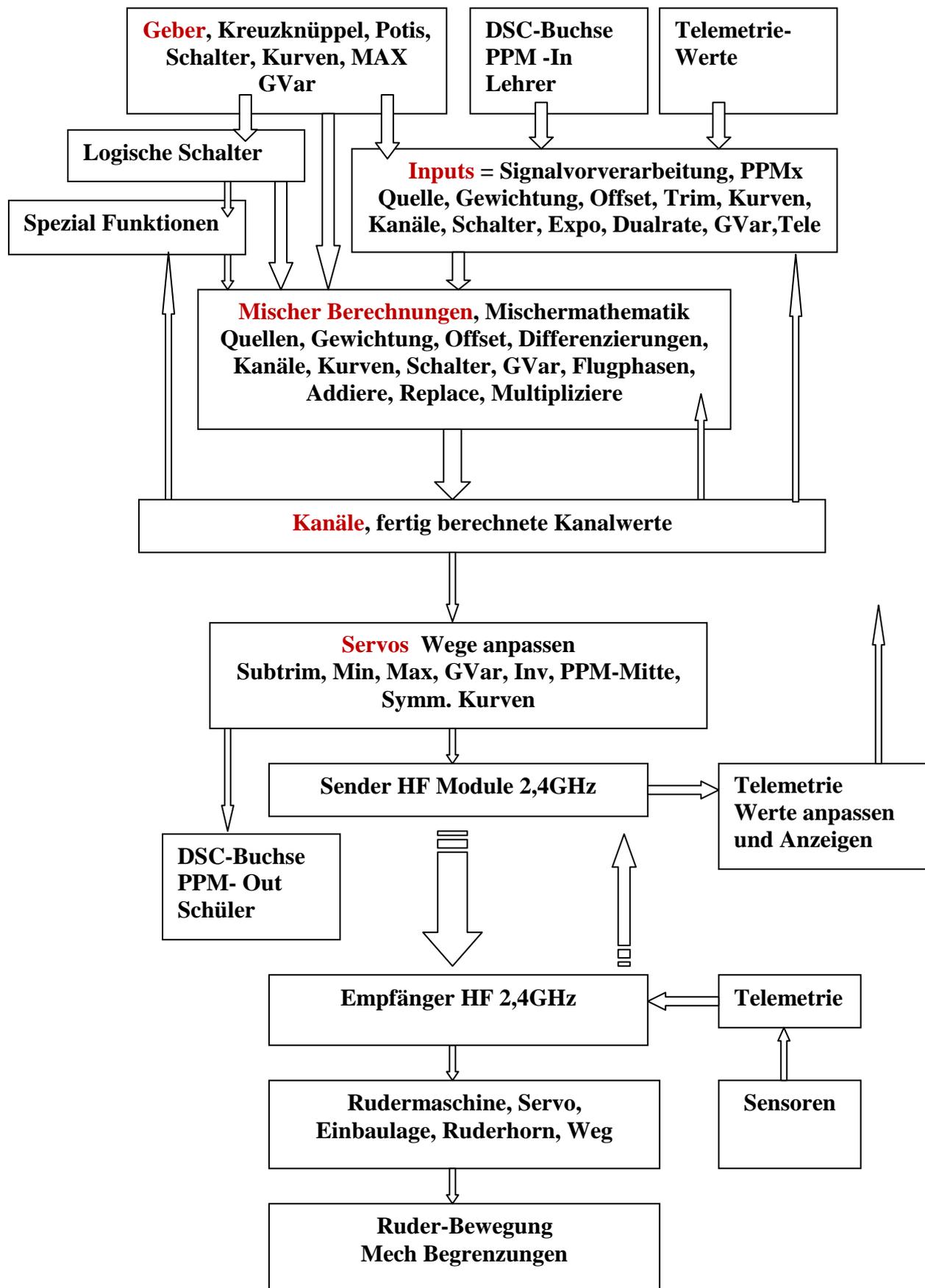
Kanäle Daten senden, Telemetrie empfangen

PPM1-PPM16

DSC-Buchse

Output

Vereinfachtes Funktions- und Programmierschema openTx



Eingaben und Werte editieren

Es gibt am Sender 6 Tasten um durch die Menüs zu navigieren und zu editieren
3 Tasten links für das Navigieren durch ganze Bildschirme **MENU**, **PAGE** und **EXIT**,
3 Tasten rechts für Eingaben und navigieren durch Zeilen und Spalten **Plus**, **Minus**, **Enter**

Sie werden hier in der Anleitung immer in eckigen Klammern gesetzt z.B. **[MENU]**

Manche Funktionen werden durch einen längeren Tastendruck (ca. 0,5Sec) aufgerufen dann steht da z.B. **[MENU Long]** andere durch einen normalen, kurzen Tastendruck dann steht nur **[MENU]**

Hier hat sich gegenüber open9x für TH9x Sendern vieles vereinfacht, da mit **[ENTER LONG]** situationsabhängig verschiedene Auswahlmenüs erscheinen.

Grundprinzip der Bedienung ist immer gleich!

Mit den 2 Cursor Tasten **[+]/[-]** in eine Zeile, Spalte gehen, das wird invers dargestellt.
mit **[ENTER]** in den Eingabemodus wechseln, das blinkt dann
mit **[+]/[-]** Werte ändern oder auswählen,
mit **[ENTER]** Wert übernehmen und **[+]/[-]** zur nächsten Eingabezeile/Spalte
oder **[EXIT]** Eingabemodus verlassen.
mit **[ENT Long]** umschalten von **Zahlen** nach **globale Variablen**

Editieren und abspeichern

Grundsätzlich gilt, dass geänderte Werte sofort wirksam und abgespeichert werden!

Man kann also den Sender ausschalten und alles ist schon gespeichert

Alle Werte werden im internen EEPROM des Microcontroller abgespeichert.
Trotzdem kann es dabei zu einer kurzen Verzögerung kommen den das abspeichern dauert ein paar Millisekunden. Man sollte also mit dem Ausschalten des Senders ca. eine Sekunde warten.

Es gibt keine "UNDO" Zurück-Funktion, jede Veränderung ist sofort gültig

Die wichtigsten Tastenfunktionen aus dem Hauptbildschirm

[MENÜ] wechselt in das Menü für alle Modelleinstellungen.

[MENÜ LONG] wechselt in die Grundeinstellungen des Senders

[PAGE LONG] wechselt in die Darstellung der Telemetrie-Anzeigen

[ENTER LONG] wechselt in die Statistik und Debug Anzeigen des Senders

Ist man in der Modelleinstellung oder in den Grundeinstellungen des Senders kann man mit **[PAGE]** / **[PAGE LONG]** durch die Seiten vorwärts / rückwärts blättern.

Werte in einer Checkbox ein/ausschalten/freigeben ☑ ☐

Mit den 2 Cursorastern [+], [-] steuert man durch die Zeilen und Spalten, dabei werden die Eingabe-Positionen invers dargestellt.

In einer Checkbox wird mit Druck auf [ENTER] die Funktion sofort ein-oder ausgeschaltet.

Das gilt auch für Werte die man nur umschaltet (Toggle-Funktion)

z.B. Maste/Slave Auswahl bei der Schüler/Lehrer Auswahl

Bearbeiten von Zeilen

In den OpenTx Menüs sind manche Eingabe mit Zeilen zu ergänzen/einfügen/löschen z.B. bei den Modelllisten, Mischern, DR/Expo, Schaltern usw.

In all diesen Fällen ist das Vorgehen immer gleich

Editieren, Einfügen, Löschen, Kopieren, Verschieben von Zeilen

Mit den Cursorastern [+] und [-] auf die Zeile gehen und mit [ENTER LONG] erscheinen situationsabhängig unterschiedliche Auswahlmenüs

das mit [+] und [-] und [ENTER] bearbeitet wird.



Im der Modellauswahlliste (1/12) ist immer das Modell mit dem Stern „*“ aktiv.

Texte eingeben

In manchen Seiten/Bereichen muss man Texte eingeben, (Modellname, Name der Flugphase usw.)

1. Mit [+] und [-] den Buchstaben auswählen
2. Mit [ENTER] wird der Buchstabe übernommen und zur nächsten Position gesprungen.
3. Mit [+] und [-] das nächste Zeichen ändern, Ziffern, Sonderzeichen, usw.
4. Mit [ENTER LONG] wird von Groß auf Kleinbuchstaben gewechselt und umgekehrt und dann gleich zur die nächsten Position gesprungen.
5. Beenden mit einfachem [EXIT]

Wie man sieht ist [ENTER LONG] eine ganz wichtige Tastenfunktion!

damit wird ein Menü aufgerufen,

eine Auswahl gemacht,

eine Umschaltung von Zahlen nach globale Variablen

oder von Großbuchstaben nach Kleinbuchstaben umgeschaltet.

oder Failsafe -Werte abspeichern

Also durchaus mal länger drücken wenn man unsicher ist!

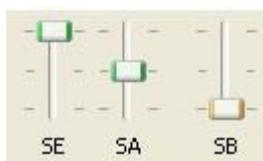
Arbeiten mit Auswahlwerten

In OpenTx gibt es auch die Möglichkeit Schalterstellungen, Potis, Sticks usw. direkt abzufragen. z.B. Schalterstellungen beim Einschalten, Mittenposition der Potis durch kurzes Piepsen, Auswahl der Flugphasen die in Mischer oder Dualrate/Expo aktiv sein sollen.

Autoselect und Autoswitch von Schalter und Potis

Sehr Praktisch: Anstatt Schalter, Potis, Sticks aus der Tabelle auszuwählen kann man auch einfach nur den Schalter betätigen oder das Poti drehen, dann erkennt die Software automatisch die Auswahl.

Darstellung Schalterstellungen in Companion, Simulation und am Sender



SE ↑ Down- Stellung, der Schalter am Sender zeigt von mir weg

SA — die Mittelstellung ist klar

SB ↓ Up- Stellung, der Schalter am Sender zeigt zu mir her

Flugphasen aktivieren/sperrn

In den Menüs gibt es auch Zeichenketten z.B. (01²345678) für die Flugphasenummern FP0-FP8 oder (RETA1234) bzw. (SHGQ¹234) für die Mittenpositionen von Sticks und Potis. Jedes Zeichen korrespondiert dabei mit einem Element für das es steht.

Ist ein Element aktiv wird es invers dargestellt, nicht aktiv als normale Darstellung.

Das kann man einstellen, indem man mit den Cursorsn [+]/[-] die Position anwählt, dann wird diese Position wieder invers blinkend dargestellt. Ein kurzer Druck auf [ENTER] und man kann diese Position jeweils aktivieren oder deaktivieren.

Verlassen des Editiermodus durch [EXIT] oder gleich durch [+] oder [-] weitergehen.

Eingabe abschließen

Alle Änderungen werden sofort in den Einstellungen dargestellt, sofort abgespeichert und wirken sich am Sender sofort aus.

Wertänderungen werden mit [EXIT] oder [ENTER] abgeschlossen. Es gibt keine Undo-Funktionen, man kann also nicht einfach wieder zu den vorherigen Werten zurück.

[EXIT] kurz geht immer **eine** Eingabe, **eine** Zeile, **ein** Untermenü zurück

[EXIT LONG] geht **ganz** zurück in die Hauptanzeige

Screenshotfunktion für LCD-Bildschirm (abV2.0.17)

In den **Spezialfunktionen per Schalter** auslösen oder per [ENTER]+ [EXIT] zusammen Wird auf der SD-Karte gespeichert, Verzeichnis: /Screenshots anlegen!

Die Hauptansicht des LCD Display

Sender einschalten, Splash Screen, dann Gas, Schalter, Failsafe Warnung (falls aktiviert)



Dieser Start-Screen kann durch einen eigenen Splashscreen ersetzt werden.
Format 212x64 Punkte, S/W



Falls Gas, Schalter Failsafe Warnung aktiviert wurde erscheinen noch 3 Fenster

- Für die Gasstellungswarnung
- Für die Schalterüberwachung
- Für die Failsafeüberwachung

3 Startbildschirme Anzeige mit [Page] umschalten und Kanal Monitor (1-16) [+] (17-32)

Sender Spannung, Empfänger Spannung, Alle Schalter, Knüppel, Trimmungen



8 Stellungen der physikalische Schalter und Zustand der 32 logischen Schalter LS1-32



Timer-Zeiten Absolut, Persistent, Modellzeit



Kanalmonitor Kanal 1-16

Kanal Monitor	
GAS	-1.4
QUER1	0.6
H HE	10.4
SEITE	0.0
QUER2	100.0
FAHRWE	0.0
CH7	0.0
CH8	0.0

Kanal 17-32

Kanal Monitor	
CH9	0.0
CH10	0.0
CH11	0.0
CH12	0.0
CH13	0.0
CH14	0.0
CH15	0.0
CH16	0.0
CH17	0.0
CH18	0.0
CH19	0.0
CH20	0.0
CH21	0.0
CH22	0.0
CH23	0.0
CH24	0.0
CH25	0.0
CH26	0.0
CH27	0.0
CH28	0.0
CH29	0.0
CH30	0.0
CH31	0.0
CH32	0.0

Die Kanal-Namen Gas, Quer1, Höhe, Seite usw. kommen von den Servoeinstellungen

Grundsätzliche Darstellung

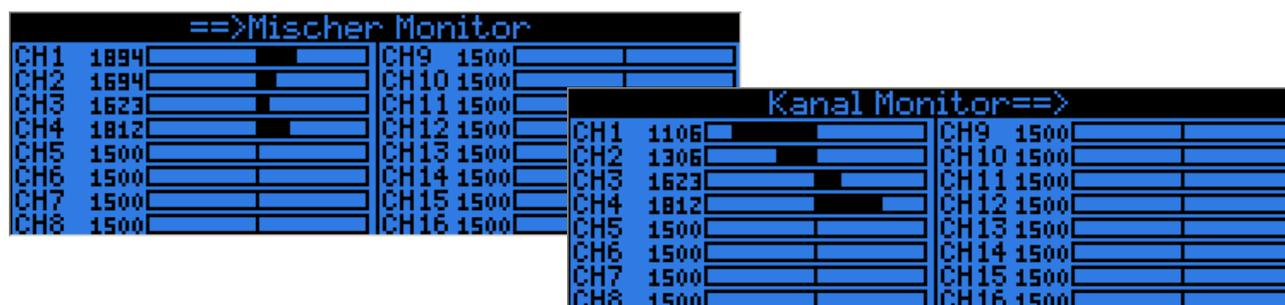
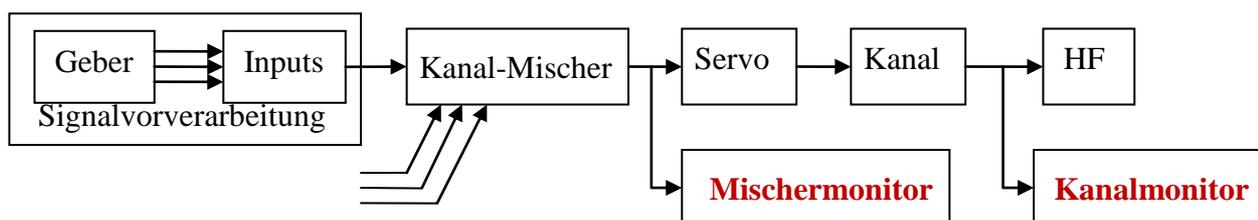
Die Hauptansicht ist in 2 Teile eingeteilt, ganz oben ist die **Statuszeile**: dort werden Spannungen angezeigt, SD-Karte aktiv, DSC und Trainer-mode, USB, LOG aktiv, Audio Lautstärke, Zeiten und es erscheinen Hilfetexte bei der Eingabe

3 verschiedenen Hauptanzeigen

- Modellname z.B. Twister (ein Segler)
- Name der gerade aktiven Flugphase (hier "Normal")
- Stellungen der 4 Sticks, der 4 Potis und der 4 Trimmungen
- Timer 1 (10:00) und seine Betriebsart (prozentuelle Zeit TH%).
- Timer 2 und seine Betriebsart (hier ABS, absolut, dauern ein, vorwärts)
- Stellungen aller Schalter und Zustände der 32 Logischen (Progr.) Schalter
- Die numerischen Werte von jeweils 8 Ausgangs Kanälen

Kanalmonitor und Mischermonitor (ab openTx V2.0.17)

Der Kanalmonitor ist der „Normalfall“, es gibt aber auch einen Mischermonitor



Der Mischermonitor zeigt das Ergebnis der „Vermischung“, also der Mathematik an.

Man beachte: Der Mischer-Balken zeigt +/- 200% an!

Das ist vor allem bei vielen Mischerzeilen vor Vorteil und zeigt das unverfälschte Ergebnis der Mischerberechnungen an.

Der Kanalmonitor berücksichtigt auch die Servoeinstellungen also die „reale Welt“

mit Servowegen, Begrenzungen und Richtungen. Der Kanal-Balken zeigt +/-100% oder +/-150% an

Mit **[ENT]** kann man umschalten zwischen Kanalmonitor und Mischermonitor

Der Kanal Monitor als Servoanzeigen

Anzeige aller 32 Kanäle Kanal 1-16, Umschaltung mit [+] Kanal 17-32



Kanal Monitor			
GAS	-1.4	CH9	0.0
QUER1	0.6	CH10	0.0
H HE	10.4	CH11	0.0
SEITE	0.0	CH12	0.0
QUER2	100.0	CH13	0.0
FAHRWE	0.0	CH14	0.0
CH7	0.0	CH15	0.0
CH8	0.0	CH16	0.0

Kanäle können im Servo-Menü auch Namen zu geordnet werden, die werden dann hier angezeigt.

In der Hauptansicht wird mit **[ENTER LONG]** ein Auswahlmenü aufgerufen um Zeiten, Flugdaten und Telemetriedaten zu löschen.



Statistik und Debugger Anzeige



Vom Hauptbildschirm kommt man mit **[ENTER LONG]** auch in die Statistik-Anzeige

TOT - Total, Gesamlaufzeit des Modells, kann man mit **ENTER** auf 00:00 setzen

TM1 - Timer1, **TM2** - Timer2,

THR - absolute Zeit der Gasstellung GSs

TH% - prozentuell Zeit der Gasstellung GS%

Die senkrechten Balken geben die Gasstellungen und die Zeiten wieder

Mit **+** gibt es noch eine Debug Modus, Speicherbelegungen, Framezeiten usw.

Telemetriebildschirme verschiedene Darstellungen

Es gibt 5 Telemetriebildschirme die man frei konfigurieren kann.

In die Telemetrieanzeigen kommt man von der Hauptanzeige mit **[PAGE LONG]**

und kann dann mit **[PAGE]** oder **[+] [-]** die Telemetrieseiten durchblättern.

Die Beschreibung der Telemetrieanzeigen erfolgt in einem gesonderten Kapitel und ist sehr umfangreich. Die Messwerte können als **Zahlwerte mit Einheiten** oder als **Balkenanzeige** konfiguriert werden. Das hier ist nur mal ein kleiner Auszug der vielen Möglichkeiten.



Sender Grundeinstellungen Übersicht (1/9)

Von der Hauptanzeige kommt man mit **[MENÜ LONG]** in das Menü für die Sender Grundeinstellungen mit 8 Seiten

Diese sind unabhängig vom ausgewählten Modell universelle grundlegende Einstellungen

Die 9 Menüs sind:

1. Sender Grundfunktionen einstellen
2. SD-Karte mit Unterverzeichnissen
3. Globale Funktionen GF1-GF64
4. Lehrer/Schüler Einstellungen
5. Versionsinfo und Softwarestand
6. Testfunktionen der Schalter und Taster
7. Testfunktion der Analogwerte
8. Hardware Einstellungen, Stufenschalter, Serielle Schnittstelle
9. Abgleichen/Justieren aller Analogwerte

Grundeinstellungen des Senders im Detail (1/9)

OpenTx Taranis V2.017

SENDER-GRUNDEINSTELLUNGEN		1/8
Datum:	2015-01-28	
Uhrzeit:	16:36:45	
Akku Spgs-Bereich	6.0-8.0	
----Töne-----		
Modus	NoKey	
Lautstärke		
Beep-Lautst.		
Beep-Länge		
Beep-Freq. +/-	+150Hz	
Wave-Lautst.		
Hintergr-Lautst.		
----Variometer----		
Lautstärke		
Niedrigster Ton	700Hz	
Höchster Ton	1700Hz	
Wiederholrate	500ms	
----Haptik-----		
Modus	NoKey	
Dauer		
Stärke		
LCD-Kontrast	25	
----Alarm wenn---		
Akkuspg kleiner	6.5v	
Inaktivität nach	10m	
Speicher voll	<input checked="" type="checkbox"/>	
Alle Töne ganz aus?	<input checked="" type="checkbox"/>	
----LCD-Beleuchtg----		
Modus	Beide	
Dauer	95s	
Helligkeit	100	
Alarm	<input type="checkbox"/>	
Startbild Ein für	4s	
GPS-Zeitzone	0	
GPS-Koordinaten-Format	GMS	
Landescode	Europa	
Sprach-Ansa9e	Deutsch	
Einheiten	Metrisch	
Schaltermitte Verz.	150ms	
Kanalvoreinstellung	GQHS	
Modus	→⊙ ↓⊙ ⊙↓ ⊙→	
<input checked="" type="checkbox"/>	Que Gas Höh Sei	

Datum und **Uhrzeit** für die eingebaute Echtzeit-Uhr eingeben

Akku-Spannungsbereich für die Ladezustandsanzeige im Batterie-Symbol eingeben

1. Töne Sound

- 1.1 **Mode:** Betriebsart für den Piepser, Summer
 - Quiet** Ganz aus, kommt nie! nichts, aber auch keine Warnungen falls z.B. die Akkuspannung zu tief ist! (Vorsicht bei Li-Po!)
 - Alarm** Nur bei Alarmmeldungen (Akkuspannung niedrig, Sender aus)
 - NoKeys** Nicht wenn Tasten gedrückt werden
 - All** Immer ein, das nervt!

2. Sound Mixer für div Töne, Ansagen, Vario und Hintergrundmusik .

- Volumen:** Gesamtlautstärke am Ausgang des Sound Mixer
- Beep Volumen** Lautstärke der Warntöne
- Beep Länge** Dauer des Beepsignals
- Beep Freq. +/-** Beep-Tonänderung wenn Eingabegrenzen erreicht
- Wave Volumen** Lautstärke der Ansagetexte
- Bgrd Volumen** Lautstärke der Hintergrundmusik
- Dauer**
- Tonhöhe** für Tonänderung bei Auf + und Ab - gute Werte 150 Hz

Varioeinstellungen Töne in Abhängigkeit der Steing- und Sinkraten, siehe Telemetrie Vario

- Vario **Lautstärke** angepasste Lautstärke
- Vario **min Frequenz** niedrigste Frequenz bei größtem Sinken
- Vario **max Frequenz** größte Frequenz bei größten Steigen
- Vario **Pulse** Pausenzeiten

Haptikeinstellungen Taranis kann auch Haktik (Vibratormotor) wenn opentx mit Haptik gewählt.

- Modus** Stumm, Alarm, NoKey, Alle
- Dauer**
- Stärke**

3. Kontrast des LCD Display einstellen Werte ca 5-45, gute Werte 15-25

4. Alarme wenn

4.1 Akku Spg kleiner : Akkuspannung zu niedrig.

Wenn die Spannung unter den eingestellten Wert fällt kommt ein Summeralarm. Wenn das richtig eingestellt ist läuft der Sender weiter, aber es ist eine Warnung dass es Zeit wird zum Landen und den Akku zu laden. (Für 6 Zellen NiMH auf 6,9V einstellen)

→Dazu muss vorher der Akku richtig abgeglichen sein, im Menü 5/6

4.2 Inactivity alarm: Wenn der Sender längere Zeit nicht bedient wird kommt nach Ablauf der Zeit ein Summeralarm. Die Voreinstellung ist 10 min.

Werte können von 1 bis 250 min eingegeben werden.

Ein Wert von 0 schaltet diese Funktion ab. Durch bewegen der Knüppel wird diese Überwachung wieder resetet und der Alarm geht weg. Gut falls man vergessen hat den Sender auszuschalten. Wird der Sender über USB versorgt ist dies Funktion auch aus.

- 4.3 **Memory Low:** Wenn ON kommt eine Warnung falls der Modellspeicher im EEPROM fast voll und nur noch 200 Byte frei sind.
Der Sender sendet nicht bis diese Alarmmeldung wieder weg ist. Dies dient der Sicherheit.
- 4.4 **Sound Off:** Das ist die „letzte Chance“ falls der Summer ganz ausgeschaltet ist. Wenn diese Funktion ON ist und der Summer mit „0“ (Quiet) außer Funktion ist kommt beim Einschalten im Start-Up ein Warnmeldung dass kein Piepser oder Summer kommt.
5. ~~Timer Events~~ **Zeitablaufsteuerung** → Ist ab Opentx 2.00 bei den Modellen
~~**Persistent:** Dauerhaft, Modell Zeit, Modellaufzeit, Modellgesamtzeit, wird aufsummiert~~
~~**Jede Minute:** Piepst jede Minute wenn der Timer läuft~~
~~**Countdown:** Piepst alle 30, 20, 10, 3, 2 und 1 Sekunde bevor die Zeit abgelaufen ist.~~
~~Auswahl: **Kein, Pieps, Stimme**~~
6. **Backlight** Hintergrundbeleuchtung steuern
Alarm: Blinkt immer wenn ein Alarm-Piepser kommt.
Mode: Betriebsart Hintergrundbeleuchtung wenn:
ON - immer Ein
OFF - immer Aus
Keys - Ein wenn eine Taste gedrückt
Stks - Ein wenn ein Stick (Knüppel) bewegt wird
Both - Beides ein Taste und Stks
Color- Für Taranis Plus kann die Beleuchtung in der Farbe um geschaltet werden
Dauer- Hintergrundbeleuchtung AUS nach x Sekunden . Bereich 0 bis 500s.
7. **Splash screen:** Startbildschirm Anzeige und Dauer (BMP-Bild-Format: 212x64Pixel 2Bit)
Kann übrigens durch Drücken einer Taste übersprungen werden.
8. **Time zone:** 1-12 Std Zeitanpassung um die GPS UTC Zeit auf Ortszeit zu korrigieren
(+1Std für Deutschland)
9. **GPS Zeitzone +/-Std:** Zeitversatz in Stunden, +1Std bei mitteleuropäischer Zeit
- 9.1 **Uhr mit GPS stellen:** Die Echtzeituhr wird mit der GPS-Zeit automatisch gestellt (ab V2.1)
10. **GPS coord:** Koordinatenanzeige NMEA oder **GMS** Format für GPS Koordinaten
z.B. 48°N 53' 11,235'' Grad Min Sec Nördliche Breite
11. **Per GPS die Uhr stellen:** damit wird automatisch die Echtzeituhr auf die GPS-Zeit gesetzt.
12. **Ländercode:** Europa (Amerika wg Einschränkung des 2,4GHz-Bereichs in den USA)
13. **Sprachansagen** in Deutsch, damit auf das passende SD-Karten Verzeichnis zugreifen.

14. **Metrisch** für Berechnungen und Anzeigen (statt Imperial = Zoll-Werte)
15. **FAI Mode** bei best. Wettbewerben dürfen keine Telemetriewerte übertragen werden wenn das einaml aktiviert wurde bleiben die dann auch dauerhaft aus!
Bekommt nan nur durch neu flashen wieder weg
16. **Switch mid delay Schalter-Mittelstellung** kurz ausblenden, zum schnellen Umschalten, damit in der Mittelstellung nichts ausgelöst wird, Kontaktunterbrechung ausblenden, um 150ms verzögert, damit keine Aktionen ausgelöst werden wenn man nur schnell durchschaltet von up nach down.
17. **Mode:** Knüppelbelegung am Sender, Darstellung als Grafik **Mode 1 - Mode 4**
18. **Rx Channels ord:** Das ist die Kanalanordnung für die Festlegung der Reihenfolge bei Anwendung von **neuen** Modellen. Damit die Mischer bei einem **neuen** Modell die Kanäle vorab schon richtig zuordnen können.
 - **RETA** bedeutet Kanal Rud = 1, Ele = 2, Thr = 3, Ail = 4.
 - **AETR** bedeutet Kanal Ail = 1, Ele = 2, Thr = 3, Rud = 4.und so weiter... bis **TAER** (16 Möglichkeiten)
Im deutschen Menü z.B. Mode 4 **GQHS** für Kanal 1-4 Belegung
(**Gas**) Gas=CH 1, (**Que**) Querruder=CH 2, (**Höe**) Höhenruder=CH 3, (**Sei**)Seitenruder=CH4
19. **MODE 1, MODE 2, MODE3, MODE4.** Die Knüppelbelegung wird grafisch dargestellt

Mode 2 Gas links, Seite links	Mode 3 Gas rechts, Seite rechts
Mode 4 Gas links, Quer links	Mode 1 Gas rechts, Quer rechts

Hinweis:

Das wird leider immer wieder verwechselt oder gleichgesetzt!

Die Knüppelbelegung am Sender **Mode 1-Mode 4** und

Die Kanalvorbelegung **GQHS.....** für die Kanäle 1-4 Gas, Quer, Seite, Höhe

haben nichts miteinander zu tun und sind vollkommen unabhängig vneinander einzustellen.

Es gibt keine feste Kanalanordnung, die kann man nach eigenen Vorlieben frei einstellen.

Beispiel für übliche Kanalbelegungen:

Graupner Kanalbelegung 1-5: **GQHS Q2**

Futaba Kanalbelegung 1-6: **QHGS frei Q2**

Spektrum Kanalbelegung 1-5: **GQHS Q2**

Wer ein externes Modul von Graupner, Futaba, Spektrum verwendet oder einen Flugcontroller muss auf die richtige Kanalreihenfolge achten. **Besonders wenn man per S-Bus verbindet, damit in externen Gerät die Kanalreihenfolge stimmt!**

Die Micro SD-Karte Unterverzeichnisse (2/9)

Neben dem **Flashspeicher** für das Betriebssystem opentx und das **EEPROM** für die Modelle ist die **SD-Karte** der dritte Speicher des Senders (seine „Festplatte“).
 Hier laufen alle Zugriffe für Firmware, Modelle, Ansagen, Bilder, Log-Daten zusammen.
 Es müssen mindestens diese Verzeichnisse und Unterverzeichnisse vorhanden sein.
 Teilweise muss man diese Verzeichnisse selber von Hand einrichten!

Diese Unterverzeichnisse braucht man mindestens auf der SD-Karte:

/SOUNDS/en **freie** Englische Ansagetexte, Warnungen, Klänge, Töne, Melodien
 /SOUNDS/en/SYSTEM **feste** Ansage-Text, Dateinamen fix aber Inhalt anpassbar

/SOUNDS/de **freie** Deutsche Ansagetexte, Warnungen, Klänge, Töne, Melodien
 /SOUNDS/de/SYSTEM **feste** Ansage-Text, Dateinamen fix aber Inhalt anpassbar
Dateinamen max 7 Zeichen, keine Leerzeichen!

/SOUNDS/de/ <Modellname> z.B. Cessna_421 Mirage_2000 AS_21
 Spezielle Sounds für das Modell selbst (keine Leerzeichen zulässig!)
Das Verzeichniss muss exakt so heißen wie das Modell!
Die Modellsound-Datei wird beim Start aufgerufen wenn eine passende wav-Datei vorhanden ist. Exakt wie der Modellname!
 Also z.B. Cessna_421.wav Mirage_2000.wav AS_21.wav

Die anderen Sounddateinamen in Sounde/de/Modellverzeichnis müssen **exakt dieser Logik** entsprechen damit sie **automatisch** aufgerufen werden wenn sie aktiv werden (man braucht sie also nicht extra programmieren!)

Schalter	Pos. Schalter	Log Schalter	Flugmodes
SA-SH	P11-P36	L1-L32	Name
SA-up.wav	P11.wav	L1-off.wav	Mode_name-off.wav
SA-mid.wav	P23.wav	L1-on.wav	Mode_name-on.wav
SA-down.wav	P24.wav	L32-off.wav	
.....	
SH-down.wav	P36.wav	L32-on.wav	

/BMP Bilder im Format 64x32 4Bit, Splashscreens im Format 212x64 2Bit
Dateiname max 7 Zeichen sortiert: Großschreibung vor Kleinschreibung

/MODELS **einzelne Modelle** werden vom Modellspeicher hier abgespeichert, mit Restore zurück in den EEPROM-Speicher des Prozessors
 *.txt mit gleichem Modellnamen für die Display Checklist-Funktion
 *.wav mit gleichem Modellnamen für autom. Ansage beim Modellaufruf

/EEPROMS Für eine Sicherung des **kompletten Modellspeichers**
 Sender Grundeinstellungen, Seite 4/8, Version, dort [**Long Enter**]
 damit Sicherung aller Modelle vom EEPROM auf die SD-Karte
 Im Bootloader dann wieder zurück von der SD-Karte ins EEPROM

/SCREENSHOTS (ab OpenTx V2.0.17) für LCD-Screenkopien
 In den **Spezialfunktionen** programmieren bzw mit [**ENT**] + [**EXIT**].

/FIRMWARES Update von openTx mit Bootloader .

Alle Sender Firmwareupdates für openTx werden nur noch in diese Verzeichnis kopiert. Dann kann der Bootloader darauf zugreifen.

Dateiname max 16 Zeichen, eventl vorher umbenennen, kürzen

Hier kommen nur die openTx.bin Dateien rein!

~~/FIRMWARE für Update aller S-Port Geräte direkt vom Sender aus mit erweiterem Bootlaoder von Mike Blandford. (openTx V2.017)~~

~~**Hier kommen nur die *.frk Dateien rein!**~~

/SPORT_Update Updates aller S-Port Geräte direkt vom Sender aus

(Dateiname max 16 Zeichen, eventl vorher umbenennen, kürzen)

Hier kommen nur die die *.frk Dateien rein! ab openTx V2.10

LOGS alle aufgezeichnete Flugdaten und Telemetriedaten werden hier als *.csv Datei gespeichert.

Achtung es gibt verschiedene Soundpacks: Bis openTx V2.017 und ab openTxV2.10

Die Sound-System-Files ab openTx V2.10 sind etwas anders als vorher,

Namen, Nummern, Einheiten wurde etwas geändert und erweitert.

Wenn also z.B. Ampere statt Meter angesagt wird, ist wohl das falsche Soundpack geladen

Die verschiedenen Sound-Packs gibt es hier für openTx V2.0, V2.1 und V2.2

<http://www.open-tx.org/links.html>

LUA Scripte Verzeichnisbaum

Für das LUA System braucht man auf der SD-Karte **zusätzliche** Unterverzeichnisse.

/SCRIPTS/
/SCRIPTS/BMP/ Bilder für die LUA Scripte
/SCRIPTS/WIZARD/ Alle Scripte für Modellgenerator, neue Modelle erzeugen
/SCRIPTS/TEMPLATES/
/SCRIPTS/MIXES/
/SCRIPTS/FUNCTIONS/
/SCRIPTS/<MODELNAME>/telemXX.lua um eigene Telemetrianzeigen zu erzeugen.
/SCRIPTS/TELEMETRY für alle Telemetriscripts (ab openTx V2.10)

Mehr info zu LUA : <http://www.open-tx.org/lua-instructions.html>

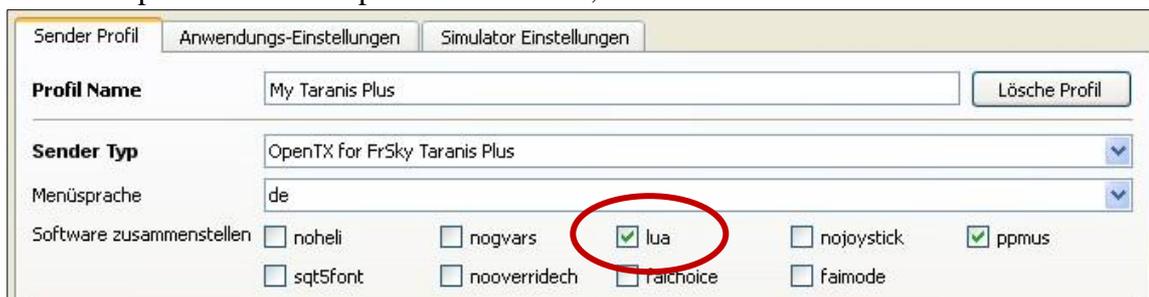
LUA- Modellgenerator zum download: <http://www.open-tx.org/2014/06/02/lua-wizard/>

Das LUA-System muss man zur Zeit noch von Hand einrichten,damit es läuft.
Das **wizard.zip** ins Unterverzeichnis /SCRIPTS/WIZARD reinkopieren, dort entpacken
Dann startet der Modellgenerator wenn ein neues Modell angelegt wird.

Achtung: Keine Sonderzeichen, keine Leerzeichen in Dateinamen!



Damit LUA-Scripte laufen können, muss der Interpreter in openTx eingebunden werden
LUA Interpreter unter Companion auswählen, dann Sender-Firmware downloaden und flashen



Globale Funktionen ab openTx V2.10

(3/9)

64 Globale Funktionen GF1 ... GF64

Der Unterschied:

Spezial Funktionen gelten **nur für ein Modell** deshalb in den Sender Grundeinstellungen
Globale Funktionen gelten **für alle Modelle** deshalb hier in den Sender Grundeinstellungen

Inhaltlich stehen hier die gleichen Funktionen wie in den Spezial Funktionen zur Auswahl



Damit kann man Dinge die man immer braucht schon hier festlegen und muss sie nicht für jedes Modell einzeln programmieren.

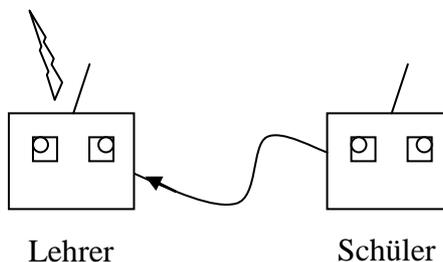
z.B.

immer der gleiche Motor Sicherheitschalter
immer die gleiche Telemetrieansagen, Anzeigen aufrufen
die gleichen Sounds, Alarme, Timeransagen, abrufen, usw.

Man kann die Globalen Funktionen verwenden, man muss es aber nicht.

Verfügbare Funktionen: Siehe Modelleinstellungen, wie bei Spezialfunktionen

Lehrer / Schüler Einstellungen TR1-TR16 (PPM1-PPM16) (4/9)



Lehrersender

Schülersender

```

MODELL EINSTELLUNGEN 2/12
Failsafe Mode      Hold
Externes HF Modul
Module             DJT
Channels Range     CH1-8
Empfänger Nr.
Failsafe Mode
Trainer Mode      Master
    
```

```

MODELL EINSTELLUNGEN 2/12
Failsafe Mode      Hold
Externes HF Modul
Module             XJT X16
Channels Range     CH1-16
Empfänger Nr.     03 [Bind] [Range]
Failsafe Mode
Trainer Mode      Slave
    
```

```

LEHRER/SCHÜLER 3/7
Modus % Quelle
Sei := 90 CH4
Höh := 60 CH3
Gas AUS 80 CH1
Que := 85 CH2
Multiplikator 1.0
Kal. 0 0 0 0
    
```

```

LEHRER/SCHÜLER 3/8
Schüler PPM1-16 als Ausgänger
    
```

Mit diesem Menü wird der Lehrersender (Master) eingestellt.

Dazu muss aber in den Modelleinstellungen 2/12 der TrainerMode von Schüler auf Lehrer umstellen. d.h. es wird festgelegt, wie der Sender die **max 16 PPM-Signale (TR1-TR16)**, die er über die Trainerbuchse/DSC-Buchse (3,5mm Mono) vom Schüler kommen, zuordnet und auf welchen Knüppel mit welchem Anteil aufmischt (**PPM= Puls Pausen Modulation**).

Der Lehrer-Sender kann sein Signal von verschiedenen Quellen erhalten (**ab opentxV2.1**)

**Lehrer-Sender: An der DSC-Buchse als CPPM-Signal,
oder am Pin 2 im Modulschacht als CPPM-Signal oder S-Bus-Signal**
(z.B. für Kabelloser Lehrer- Schüler Betrieb)
oder an der seriellen Schnittstelle im Akkufach als S-Bus-Signal

Schüler-Sender: Gibt im Regelfall ein CPPM-Signal an der DSC-Buchse aus
kann aber auch ein S-Bus-Signal erzeugen (serielle Schnittstelle im Akkufach)

Die 4 Schülerwerte ersetzen, angepasst und aufbereitet, **direkt** dort die 4 Kreuz-Knüppelwerte des Lehrersenders, mehr passiert da nicht. (Zum Verständnis: Siehe Blockschaltbild des Senders)

Normalerweise sind im Schülersender die 4 Hauptkanäle auf den Knüppeln und kommen als **PPM1- PPM4-Signale** zum Lehrer-Sender (**TR1-TR4**). Diese werden im Lehrersender wieder den 4 Hauptkanälen zugeordnet. Die restlichen **PPM5-PPM16 (TR5-TR16)** können weiterhin in Lehrer-Sender frei verwendet werden um alles möglich zu steuern. z.B. Fahrwerk ausfahren, Klappen setzen, usw. Oder sogar bei FPV als Beobachter-Funktion (Spotter-Funktion) den Lehrer-Sender via Custom Switch selber zu übernehmen.

Tip:

Man kann die 16 PPM1-PPM16 (**TR1-TR16**) Kanäle auch direkt verwenden, dann aber mit Offset und Gewichtung den **Abgleich selber** durchführen. Mehr macht das Kalibrieren der PPM1-PPM4 (**TR1-TR4**) auch nicht. Ziel: Gleiche Nulllage und gleiche Wege wie beim Lehrer.

Der Schülersender braucht nicht die gleichen Modelleinstellungen und Kanalbelegungen haben. Alle Mischer und Einstellungen am Lehrersender bleiben erhalten und werden mit dem Signal vom Schülersender bedient. Wenn also ein Ausgangssignal am Lehrersender mit einer Expokurve verarbeitet wird, so bleibt das erhalten.

Normal kommen aus jedem Schülersender nur die Knüppelsignale, eventl. mit Trimmwerten, aber keine nachverarbeiteten Signale!

1. **Mode:** Legt die Betriebsart fest,
AUS, Kanal wird nicht verwendet
+= Schülerwerte werden zu den mit den Lehrerwerten addiert
:= Schülerwerte ersetzen die Werte des Lehrers
2. **Percentage:** Prozentueller Anteil, wie stark die Schülerwerte übernommen werden. Das entspricht praktisch einer Reduzierung der max Steuerausschläge einzelner Kanäle (gut bei Schülern mit unruhigen, schlagartigen Knüppelbewegungen)
Die Werte gehen von +125% 0% -125%, negative Werte = (Servo)-Signalumkehr!
3. **Source:** freie Kanalzuordnung der 4 Schülerkanäle
z.B. Gas kommt von Schüler- Kanal3, Querruder vom Schüler-Kanal 1

Der **Multiplikator** bearbeitet alle 4 PPM-Schüler-Eingangskanäle gemeinsam.

Damit kann man Schülersender anpassen die keine Standard PPM-Signale erzeugen oder aber auch negative -PPM Signale ausgeben z.B. mit -1.0 (Signalpolarität und PPM-Impulsbreite anpassen)

4. **Kalibrierung: Am Schülersender alle Geber auf Mitte, Trimmungen auf Mitte**
dann am Lehrersender auf **Kal.** gehen, [**ENTER**] drücken,
alle 4 Kal. blinken, dann [**Menü**] um die PPM der Schüler-Mitten-Werte zu übernehmen.
Das ist der Mittenabgleich, damit die 4 Schülergeber zu den Lehrergebern exakt passen.
→ Das funktioniert aber erst wenn auch ein PPM Signal an der DSC Buchse anliegt!

Ansehen und vergleichen kann man dann die Signale im Limitmenü (Servowege), mal auf die Lehrerwerte, mal auf die Schülerwerte umschalten.

5. **Trimmungen** am Lehrersender wirken auch auf die eingehenden PPM-Werte die vom Schüler kommen. Der Lehrer kann also die Schüler-Werte selber nachtrimmen!

Das Eingangssignal an der DSC-Buchse sollte min. einen Pegel > 3V haben.

Es ist aber egal ob ein positives oder negatives PPM Signal an der (3,5mm Mono) DSC-Buchse eingespeist wird. PPM-Signal an der Spitze, Masse am Ring

Hinweis: Es gibt in den Foren viele Signal-Anpassschaltungen, vor allem wegen der vielen Graupnersender, die meist keine PPM- Normsignal mit 0V / 5V liefert, sondern Pegel von -2V +0,8V. Hier muss man genau nachmessen und aufpassen damit die richtigen Pegel angepasst werden und/oder einen Signalinverter / Levelshifter einbauen.

Spektrum Sender liefern bzw brauchen als PPM-Signal folgende Einstellung:

6-8 Kanäle, 22,5ms, 400us, Negativ

3,5mm Mono und Stereokabel funktionieren

Auch ohne Oszi kann man grob prüfen ob ein PPM-Signal kommt.

Mit einem Multimeter. Das liefert zwar nur einen Mittelwert, aber das reicht.

8 Kanal, 22,5ms, 300us, positiv, das Multimeter misst ca 0,370V

8 Kanal, 22,5ms, 300us, negativ, das Multimeter misst ca 2,780V (bei einem 3,3V Pegel)

Das PPM Signal, PPM Impulsrahmen, Einstellungen, im Detail

(Option **ppmus**, im Hauptmenü, Servomonitor, **alle** Kanäle werden in **µs** angezeigt statt **%**)

Normalerweise ist ein **PPM- Signal (Puls-Pausen Modulation)** so aufgebaut:

22.5ms Framezeit, (Gesamtzeit)

300µs Kanalstartimpulslänge (Positiv oder Negativ)

+ Positive PPM Impulsstart oder - Negative PPM Impulsstart

Kanalimpulslängen bei (-100%, 0% +100%)

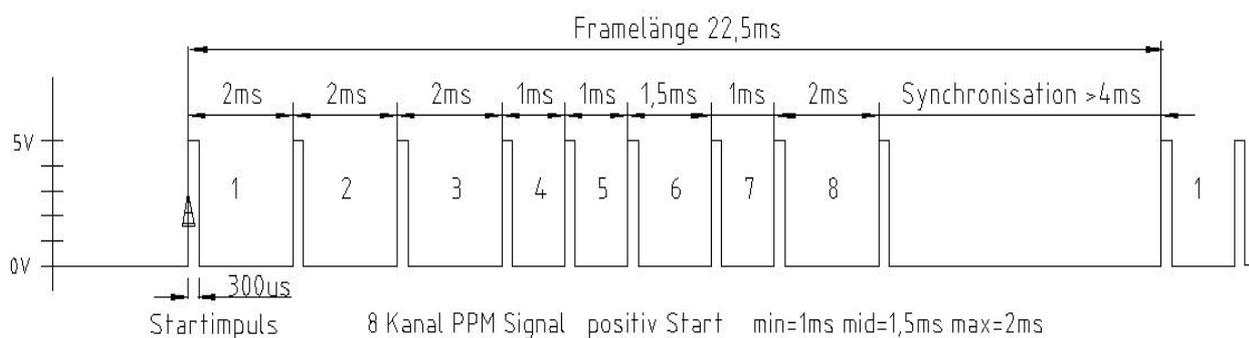
Min= 1,00ms Mitte= 1,50ms Max= 2,00ms

Bei 22,5ms Framezeit kann man max. 9 Kanäle übertragen.

(9*2ms=18ms, 22,5ms-18ms=4,5ms Start-Synchronisationszeit)

(8*2ms=16ms, 22,5ms-16ms=6,5ms Start-Synchronisationszeit)

Beispiel: 22,5ms Rahmen, Positive Kanalstartimpulse, 8 Kanäle, PPM-Signal



Achtung manche Systeme brauchen bzw. erzeugen andere Werte:

z.B. 27ms Framelänge, 400µs Kanaltrennimpuls, negative Impulsflanke

Die Impulsmitte ist teilweise nicht 1500µs sondern 1520µs bei Futaba, 1600µs teils Multiplex

mit Min/Max = +/-600µs oder Min/Max= +/-550µs

Und das wird dann auch noch als +/-125 % oder +/-150% oder gar +/- 160% angegeben!

Somit ergeben sich unterschiedliche Darstellungen, Werte und Umrechnungen!

Für Opentx gilt:

Normale Wege: Impuls-Mitte =0%= 1500µs, Min= -100%= 988µs, Max= +100% = 2012µs

Erweiterter Wege: Impuls-Mitte =0%= 1500µs, Min= -150%= 618µs, Max= +150% = 2268µs

Was bei opentx +100% ist, das wird bei Graupner mit +125% bezeichnet

Futaba und Multiplex haben andere Mitten und Wege 1520µs +/-500µs bzw 1480µs +/-550µs

Entscheidend sind immer die min und max Wege in µs und nicht die %-Angaben.

Beispiele für möglich PPM- Zeitwerte, das sind nur ca. Werte und je nach Senderhersteller etwas unterschiedlich.

8 Kanal: 8x2000us + 4000us Synch = 20000us meist 22500us

9 Kanal: 9x2000us + 4000us Synch = 22000us meist 22500us

12 Kanal: 12x2000us + 4000us Synch = 28000us meist 27000us

16 Kanal: 16x2000us + 4000us Synch = 37000us gibt es normal nicht

Im Lehrermode der Taranis werden die PPM-Signale an der DSC-Buchse eingespeist und als **TR1-TR16** verarbeitet.

Beispiel: Flugsimulator am PC

Auch hier wird die Betriebsart Slave verwendet und an der DSC-Buchse die bis zu 16 Kanäle als PPM-Signal auszugegeben. In der Regel steckt man dann dort einen Wandler ein, PPM to USB, der dann die Signale am PC als Joystick-Signale oder HIT (Human Interface) darstellt und diese dann vom Flugsimulator-Programm übernommen werden.

Aber am Markt gibt es jede Menge Billig-Schrott von diese PPM to USB Wandler für ca 5€, dann wird entweder gar nichts oder nur Kanal 1 nicht richtig gewandelt.

Gute Wandler kosten ca 15-20€. Oder aber selber bauen für ca 10 €
Suchbegriff unter Goggle: **PPM2USB**

Hinweis:

Ab opentx V2.05 wird der Sender auch automatisch an der USB-Schnittstelle als PC-Joystick erkannt → Details dazu siehe Beispiel!

Wenn in der Firmwarezusammenstellung die Option Joystick ausgewählt wurde.

Dann ändert sich aber auch das Verhalten und der Ablauf wenn man das USB-Kabel einsteckt wg Bootloader, Update, Joystick, SD-Karte erkennen

Varianten des USB Verbindungsaufbau mit Taranis X9D, X9E ab openTx V2.0

- 1. Taranis ausgeschaltet => USB anschließen => STM32 Bootloaderfunktion aktiv (DFU, Zadig)**
- 2. Taranis eingeschaltet => USB anschließen => Joystickfunktion aktiv**
- 3. Taranis Trims halten, einschalten => USB anschließen => SD-Datenträger, Update, Boot**

Beispiel: Trainer Funktion aktivieren und testen

Das ist eigentlich ganz einfach und wird in 3 Schritten erledigt.

1. Modelleinstellungen 2/12 als Master

Grundsätzlich mal man das Modell als Lehrer-Modell auswählen, also als Master definieren. d.h. dieses Modell soll auch von Schüler gesteuert werden können.

2. Im Sendermenü unter Lehr/Schül. 2/6

werden wie oben beschrieben die hereinkommenden Signale PPM1-PPM16 vom Schülersender gemessen, die Mittelstellungen, Min und Max Werte ermittelt und angepasst.

Für die Freigabe von Schülerkanälen verwendet man einen der Schalter **SA-SG** oder den Taster **SH** als Trainer-Taster um einen oder mehrere Schüler-Kanäle durchzuschalten. Lässt man den Taster **SH** los, wirken sofort wieder die Kreuzknüppel vom Lehrersender, genau so soll es sein.

Man Kann jeden beliebigen physischen oder virtuellen Schalter/Taster verwenden!

3. In den Spezial Funktionen Menü 11/12

Kann man dann einzelne Kanäle individuell freigeben oder sperren

Beispiel:

Spezial Funktionen 11/13

SH Lehrer Gas

SH Lehrer Qeu

SH Lehrer Hör

SH Lehrer Sei



oder alle 4 Kanäle zusammen freigeben/sperren **LehrerGQHS**

SH LehrerGQHS

Mit dem Häkchen kann man immer die Zeile der Spezialfunktionen ganz einfach sperren oder freigeben ohne sie löschen zu müssen.

Wird jetzt der **Trainer**-Taster betätigt, dann wird je nach gesetztem Häkchen der/die aktivierten Schüler-Eingänge (hier Höhe, Seite und Querruder) anstatt des Lehrer-Knüppel an die Mischer umgeleitet, verarbeitet und an das Modell gesendet.

Das ist einfach und sehr praktisch, da am Lehrermodell nichts geändert oder angepasst wird.

Im Limit-Menü 7/12 (Servoeinstellungen) kann man sich dann die Signale Kanal für Kanal ansehen und die Werte vergleichen, einmal vom Lehrersender und wenn man z.B. **SH**-Taster betätigt vom Schülersender, Darstellung in μs wenn **PPM μs** Option gewählt wurde. Dann sollte bei gleichen Geberstellungen von Schüler und Lehrer die gleichen Anzeigen erscheinen (eventl. per Multiplikator Signalpegel und Signalbreite anpassen)

Beim FrSky Taranis Sender muss man definieren ob der Sender als:

Master = Lehrer = PPM Signale an der DSC-Buchse als TR1-TR16 empfängt
oder

Slave = Schüler = PPM Signale an der DSC-Buchse ausgibt

Nur durch Einstecken des Kabel in der DSC-Buchse passiert noch gar nichts!

Beispiel: FPV Spotterfunktion

Auch das ist ganz einfach möglich!

Der Lehrersender ist der Sender mit dem der FPV'ler mit seinem Modell fliegt.

Am Schüler Sender ist der Spotter, der den Luftraum überwacht.

Er hat die Möglichkeit das Modell bei Gefahr selber sofort zu übernehmen,

er muss somit nicht warten bis der FPV'ler das Modell an ihn übergibt.

Der Spotter am Schülersender hat also Vorrang.

Das ist genau anders als beim normalen Lehrer/Schüler-Betrieb, wo der Lehrer Vorrang hat.

Das geht folgendermaßen:

Vom Schülersender kommen bis zu 16 Kanäle als **PPM1- PPM16** Signale (**TR1-TR16**) an den Lehrersender über die DSC-Buchse rein. Davon werden Kanal **CH1- CH4** in der Regel mit den 4 Hauptfunktionen belegt, die wie bei Lehrer/Schüler-Betrieb eingestellt und kalibriert werden.

Kanal 5 im Schülersender wird nun mit einem Schaltkanal -100% und +100% belegt, der dann als **PPM5 (TR5)** in Lehrersender als Logischer Schalter 10/13 (Custom Switch) aktiv wird z.B. **CS7 a>x PPM5 +50** d.h. CS7 wird aktiv wenn der Eingangskanal PPM5 >50% wird.

Mit diesem **CS7** aktivieren/sperren wir in den Spezial Funktionen 11/13

die 4 Hauptkanäle. Damit kann der Spotter das Modell des FPV'ler selber übernehmen.

Dort steht dann: **CF1 CS7 Lehrer**

Mit dem Häkchen kann man die Zeile der Spezialfunktionen sperren oder freigeben ohne sie löschen zu müssen.

```
CUSTOM SWITCHES 10/13
CS1 --- 0 ---
CS2 --- 0 ---
CS3 --- 0 ---
CS4 --- 0 ---
CS5 --- 0 ---
CS6 --- 0 ---
CS7 a>x PPM5 50 ---
```

```
CUSTOM FUNCTIONS 11/13
CF1 CS7 Trainer 
CF2 ---
CF3 ---
CF4 ---
CF5 ---
CF6 ---
CF7 ---
```

Je nach Firmwarestand und Sprache heist das:

Deutsch:

LS = Logischer Schalter 10/13

SF = Spezial Funktionen 11/13

Englisch:

CS= Custom Switches 10/13

CF=Custom Funktions 11/13

LS = PS = CS Logischer Schalter, Progr. Schalter, Custom Switch 10/13

SF= CF Spezial Funktionen

TR1-TR16 Trainer Input-Werte als PPM-Signal=Puls-Pausen Modulation **PPM1-PPM16**

Beispiel: Kabelloser Lehrer-Schüler Betrieb

Das Prinzip geht so:

Der Schüler hat einen Schülersender und einen Schülerempfänger die miteinander gebunden sind.
Der Schülerempfänger muss ein CPPM-Summensignal erzeugen.

Dieser Schülerempfänger wird am Lehrersender befestigt und das PPM-Summensignal wird an der DSC-Buchse des Lehrersenders eingespeist.

Die Stromversorgung für den Schülerempfänger kann aus dem Lehrersender erfolgen.
Das PPM-Summensignal muss in etwa der Norm (22,5ms, 300us, Positiv) entsprechen und
Der Pegel muss mind. 3,3V betragen.

Im Internet gibt es dazu zahlreichen Anleitungen.

Beispiel:

Empfänger D4R-II Stromversorgung und Signal intern mit Servokabel herausgeführt



Achtung: Maximale Versorgungsspannung des verwendeten Empfänger beachten!
eventl noch einen 5V oder 3,3V Regler davorschalten!

Alternative: Am Pin 2 im Modulschacht kann man auch CPPM-Signale
und S-Bus-Signale einspeisen.
Dort liegt auch die unregelte Akkuspannung und Masse an

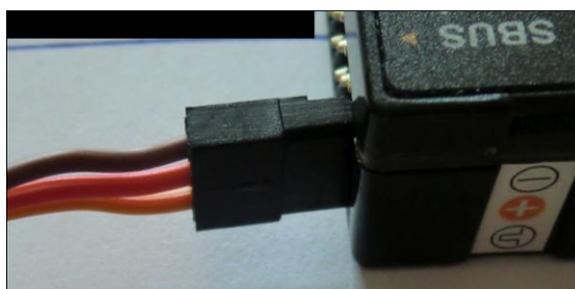
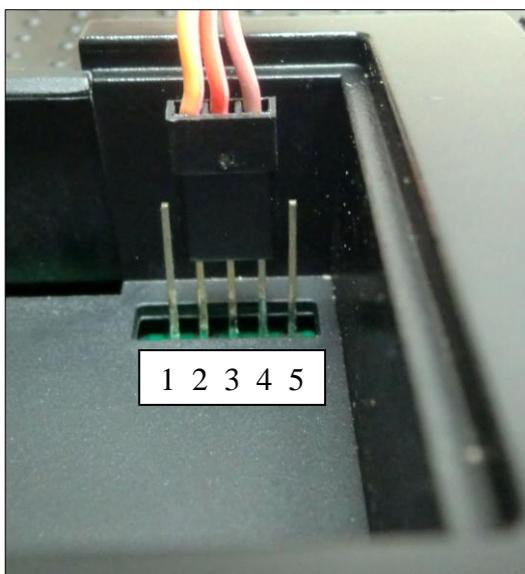
Ab openX V2.10 im Modulschacht Pin 2 als Input für CPPM-Signal und S-Bus-Signal

In den Modelleinstellungen freischalten:

Lehrer S-Bus: Empfängt S-Bus Signale am **Pin 2** von einen X-Empfänger mit S-Bus Ausgang

- | | |
|-------------------------|--|
| 1 CPPM oder PXX | Ausgang an externes HF-Modul |
| 2 Heartbeat | Eingang für S-Bus-Signal oder CPPM-Signale |
| 3 VCC | Akkuspannung (Achtung kontrollieren, dass der Empfänger das auch kann!) |
| 4 Ground | Masse |
| 5 S-PORT Ausgang | Für direktes Update von S-Port Geräten |

Sender und S-Bus Empfänger mit normalen Servokabel verbinden



An Pin 3 liegt die Akkuspannung des Senders. Jje nach Akku 6 -8 Zellen NiMh, 2- 3Lio
Viele Empfänger können nur 5V oder 6V Frsky max 10V
Also hier eventl eine Spannungsregler vorschalten.

Max Spannung des Empfängers beachten! X-Empfänger bis max 10V möglich.

Softwareversion (5/9)

```
VERSION 4/8
VERS : 2.1.0
DATE : 2014-08-13
TIME : 17:59:28
EEPR : 217

[ENTER Long] Backup EEPROM->SD-Card
```

Zeigt den Softwarestand und das Format des EEPROMs an

OpenTx V2.0 hat EEPROM V216

OpenTx V2.1 hat EEPROM V217

SVN: Software Versions Nummer SVN und Release-Stand

Date : Firmware Compiler Datum

Time : Firmware Compiler Uhrzeit

Da die Software OpenTx ständig weiterentwickelt wird, helfen diese Angaben falls Probleme oder Fehler auftreten bei der Fehlersuche.

Projekt Seite ist: <https://github.com/opentx/opentx>

Software-Seiten: <http://downloads-20.open-tx.org/firmware/binaries/>

<http://downloads-20.open-tx.org/companion/>

Im EEPROM des Senders sind die Modelle gespeichert.

Alle Modelle zusammen kann man hiermit auf die SD-Karte sichern.

Mit [EnterLong] kann der komplette Modellspeicher auf die SD-Karte ins Unterverzeichnis /EEPROMS kopiert werden
siehe SD-Karte Unterverzeichnisse

Zurück von SD-Karte ins EEPROM dann im Bootloadermenü

```
SD CARD 2/8
[BMP]
[Docs]
[EEPROMS]
[FIRMWARES]
[LOGS]
[MODELS]
[SCRIPTS]
```

Funktionstest aller EingabeTaster (6/9)

```
Schalter Test 5/7
Minus 0
Plus 0      Trim- +
Page 0      +⊗ 0 0
Enter 0     ↑⊗ 0 0
Exit 0      ⊗↓ 0 0
Menu 0      ⊗+ 0 0
```

Dieses Menü zeigt den digitalen Zustand von jedem Eingabe - und Trimtaster an. Drückt man einen Schalter/Taster so wird er invers dargestellt.

Funktionstest aller Analoggeber (7/9)

```
Analog Test 6/7
A1: FFF4 -1  A2: FDD7 -54
A3: 0155 33  A4: 00F7 24
A5: 02C9 -69 A6: 0400 100
A7: 001C -2  A8: 020E 51
Akku. Kalib. 120
```

Hier sieht man alle Analogeingänge als Hex-Zahl und als Dezimalzahl. Der Wertebereich geht von 0 bis 1024 (0 bis 0X3FF)

- **A1-A4** die Werte der 4 Steuerknüppel
- **A5-A8** die Werte der 4 Potentiometer am Sender

Auch die Sender-Akkuspannung wird gemessen und normal korrekt angezeigt.

Sender-Akku abgleichen:

→ Das ist immer dann notwendig wenn ein neues Firmware-Update aufgespielt wurde!

Man muss einmal auch die Akkuspannung unter Last mit einem Voltmeter messen und den genauen Wert hier eintragen. Damit der tatsächliche Wert auch angezeigt wird.

Nur dann kann die Akku-Unterspannungs-Warnung richtig eingestellt werden!

Hinweis:

Ein 6 Zellen NiMH-Akku ist voll bei 1,3V/Zelle = 7,8V und leer mit 1,1V/Zelle = 6,6V

Im Sendergrundmenü 1/8, Akku Spg-Bereich 6,0V - 8,0V für die Balkenanzeige einstellen.

Achtung:

Ich empfehle dringend nach einem Firmwareupdate auch die Akkuspannung nachzumessen und einzustellen. Vor allem dann, wenn man nicht absolut sicher ist welche Kalibrier- und Hardware-Einstellungen man vom PC in den Sender übertragen hat.

(Companion: Sender Grundeinstellungen beachten!)



Hardware einstellen (8/9)

Mit openTx V2.0 kann man auch die Hardware erweitern und mit Namen anpassen



S1, S2, S3 (S3 bei der Taranis Plus)
 Poti mit und ohne Mittenraster
 Stufenschalter mit 6 Stufen
Serial Port im Akkufach:
 S-Port Rohdaten oder Telemetriewerte

Aufpassen: S3 in der Taranis Plus nur aktivieren wenn er auch tatsächlich eingebaut ist, sonst piepst der Sender, da der S3 nicht gefunden wird!

Ab Opentx V2.10

Beim Sender X9E und X12D Horus gibt es bis zu 18 Schalter SA bis SR und 12 Geber

Die Namen aller Geber und Schalter kann man umbenennen. (3 Zeichen wg Platz)

Die Schalter kann man in der Funktion umstellen 2-stufig, 3-stufig, Toggle

Der Serielle Port wurde in den Funktionen erweitert.

Das PPM-Signale für den Lehrer hat mehrere Quellen: DSC-Buchse, CPPM-Modul, S-BusModul

Die Darstellung der Schalter wurde erweitert



6 Stufen-Schalter als Potiersatz

Die Potis S1, S2, S3 kann man durch Stufenschalter ersetzen (Multipos-Switch), bei Taranis-Plus. Die 6 Stufen kann man einlesen, abgleichen und/ oder mit einer Treppenkurve selbst verschieben. Die Stufen sind: -100% -60% -20% +20% +60% +100% alternativ: Widerstandswerte anpassen. S1 wird dann als S11 S12 S13 S14 S15 S16 angezeigt. (S2=S21..S26 S3=S31..S36)



Wenn ein Stufenschalter eingebaut wurde, muss er auch abgeglichen werden, siehe (8/8), da er ja ein Poti ersetzt!

Der Stufenschalter wird so abgeglichen, dass er **zuerst links steht**, das ist Stufe 1, dann nacheinander die Stufen durchschalten.

Youtube Video: <http://www.youtube.com/watch?v=Ts0EzeJsoNc>

Taranis Plus:
S3 an K2 anschließen
Masse, Signal, Plus



Sender Grundeinstellungen:

Schalterstellung verzögern um 150ms damit werden Umschaltssprünge ausgeblendet.

Einstellen, Anwenden und Anpassung der Werte siehe Beispiel

Haptikmodul: (Ist in der Taranis Plus schon eingebaut)

Für die Taranis gibt es auch ein Haptik / Vibratormodul zum einfachen Nachrüsten, das von der Software angesteuert werden kann.

Haptikmodul selber nachrüsten:

Je nach Version A01 oder B01 der Taranis muss die Signalleitung anders verdrahtet werden.

A01 Version: Signal von Pin 2 des Soundmodul, Plus und Masse wie bei B01-Version

B01 Version: <http://open-txu.org/how-to-install-frsky-haptic-vibration-x9d-taranis/>

Youtube Video: http://www.youtube.com/watch?v=T6iMOBOt_Jk

Dazu muss die opentx Software mit der Option Haptik auf dem Sender aktiviert sein.



Es gibt auch zahlreiche Selbstbausaltungen mit 3-4 Widerständen und 1 Transistor, 1 Freilaufdiode Vibratormotoren für 1,5-3,3V und ca 20-50mA gibt es z.B. bei Pollin oder Reichelt.

Serielle Schnittstelle für S-Port (Smart-Port) Output, Telemetrie-Input, Debug-Mode

Der serielle Port im Akkuschacht kann nun freigeschaltet werden und wird zukünftig noch weitere Funktionen erhalten.

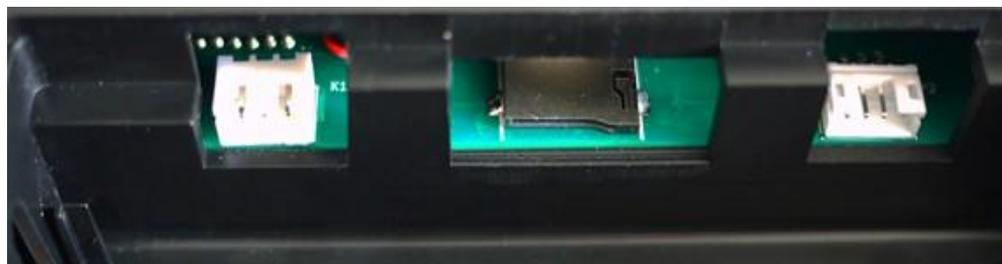
Er liefert jetzt die Werte des internen XJT-Moduls im S-PORT-Format so wie es ein externes XJT-Modul auch tut. Kann aber auch Telemetriewerte empfangen.

Der Debugmodus ist für Testfunktionen der Programmierer (normal nicht verfügbar!)

Akkuanschluss JST-XH

micro SD-Karte

serielle Schnittstelle JST-PH 2mm



Taranis Serieller Port Pinbelegung von links nach rechts:

Name	Funktion	Schnittstelle am PC bzw USB-Adapter
GND	Masse	Pin 5 des Serial Port (Ground)
VMAIN	ca 6-8V nach F1, D5	
UART Tx	Transmit von Taranis	Pin 2 des Serial Port (Rx)
UART Rx	Receive an Taranis	Pin 3 des Serial Port (Tx)

Port Funktion und Übertragungsparameter

Datenformat: 8 Bit Daten, 1 Stopbit, No Paritybit, No Flow Control

Die Baudrate ist von der Funktion abhängig:

S-Port-Mirror:	57600,8,1,N	Tx, Ausgang empfangene S-Port Telemetrie Daten durchreichen
Telemetrie:	9600,8,1,N	Rx, Eingang empfängt Telemetriewerte (für D-Empfänger)
S-Bus Schüler:		Tx, Ausgang ein S-Bus-Signal wird erzeugt
Debugmode:	115200,8,1,N	Tx, Ausgang (nur im Debug-Mode von openTx)

Elektrische Pegel

Der Taranis Serial Port verwendet zwar die RS-232 Polarität für RX und TX Signale, aber der Spannungspegel erreicht am TX Anschluß nicht den vollen RS-232 Wert (-15V to +15V) Der RX-Input der Taranis ist vollen RS-232 Pegeltolerant.

Trotzdem arbeitet er mit allen standard seriellen Adapter zusammen (auch USB Serial Adaptern)

Logik	RS-232 Standard	Taranis Pegel
0	+ 15V	3.3V
1	-15V	0V

Stecker-Typ:

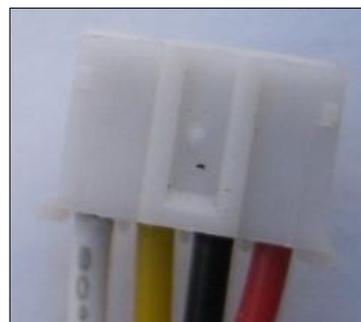
JST-PH 4-Pin 2.0mm

(wie bei vielen 3 Zellen Balancerkabeln)

USB to RS232 Adapter:

FrSky USB Adapter arbeitet perfekt!

FrSky FrUSB-3 (FUC-3) alle Kabel sind dabei



FrSky FrUSB-3 (FUC-3) mit allen Kabeln

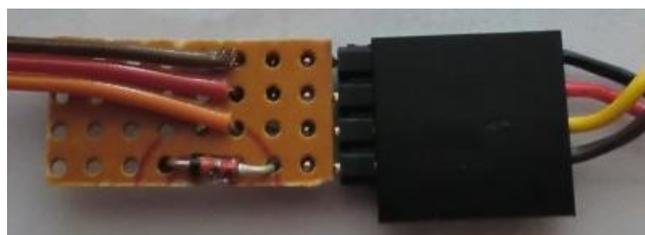
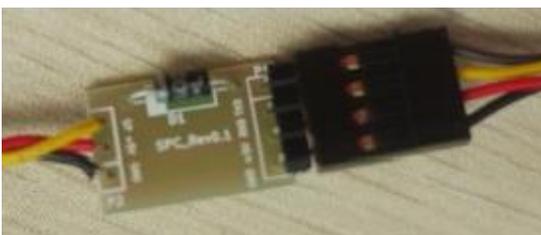
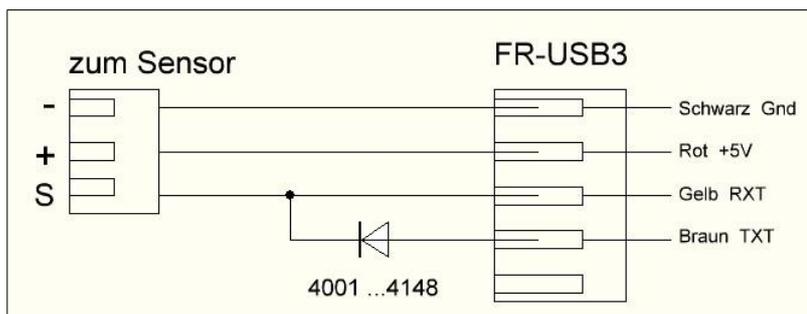


Der Frsky **FrUSB-3 und der Diodenadapter** wird für alle Software-Update **aller** Smart-Port Sensoren und für das EU-Update EN3003 328 ETSI V1.8.1 update **aller** HF-Module und Empfänger benötigt!

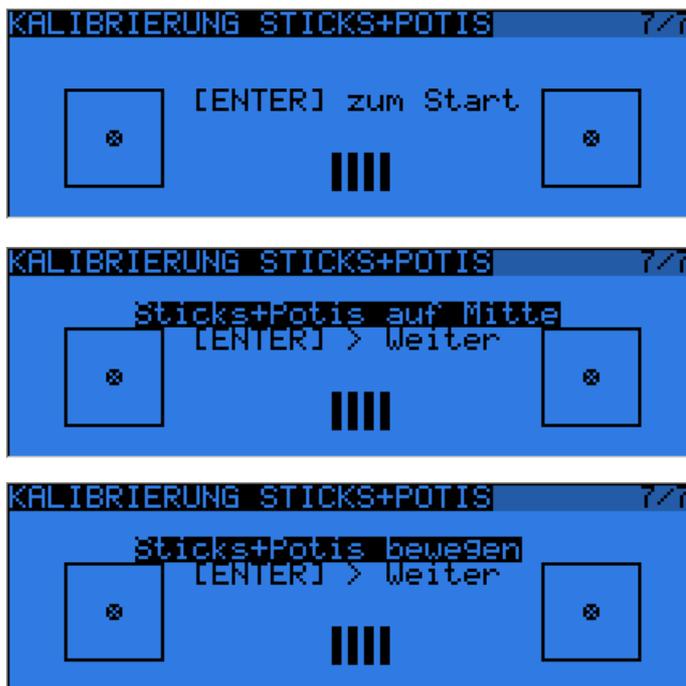
Unbedingt vorher den Silab-Treiber laden, nicht den FrSky FrUSB-3 vorher schon einstecken!

Siehe Beispiel: Smart-Port Sensoren Firmware updaten

**Zum Updaten braucht man auch diese Diodenanpassung:
Anpassung mit Diode 1N4001 zum Selbermachen auf Lochraster, oder fertig kaufen.**



Analoggeber abgleichen (9/9)



Hier **muss** man alle Analogeingänge **A1-A8** (4 Sicks, seidl. Slider, Potis) einmal abgleichen!

Auch ein Stufenschalter wird hier abgeglichen und die Stufen angezeigt.

→ Das ist immer dann notwendig wenn ein neues Firmware-Update aufgespielt wurde!

Das Abgleichen geht wie folgt:

1. **[ENTER]** drücken
2. Alle Analoggeber, **Sticks** und **Potis**, auf ca. Mittelstellung bringen
Beim Stufenschalter, zuerst ganz nach links stellen, das ist Stufe 1
3. **[ENTER]** drücken
4. Alle Analoggeber, **Sticks** und **Potis**, nacheinander ein paar Mal von Min nach Max bewegen.
Beim Stufenschalter nacheinander die Stufen einmal nach rechts durchdrehen.
5. **[EXIT]** drücken und die Werte werden gespeichert.

Achtung:

Ich empfehle dringend nach einem Firmwareupdate auch die Akkuspannung nachzumessen und einzustellen. Vor allem dann, wenn man nicht absolut sicher ist welche Kalibrier- und Hardware-Einstellungen man vom PC in den Sender übertragen hat.

(Companion: Sender Grundeinstellungen beachten!)



Modell Einstellungen

Vom Hauptbildschirm kommt man mit **[MENU]** direkt in die Modell-Auswahl und Modelleinstellungen.

Es gibt 60 Modellspeicher. Hier wird jedes Modell konfiguriert.

Das sind pro Modell bis zu 12 Seiten mit 4-6 Untermenüs möglich

Mit **[PAGE]** eine Seite vorwärts mit **[PAGE Long]** eine Seiten rückwärts.

Die 13 Modell-Menüs:

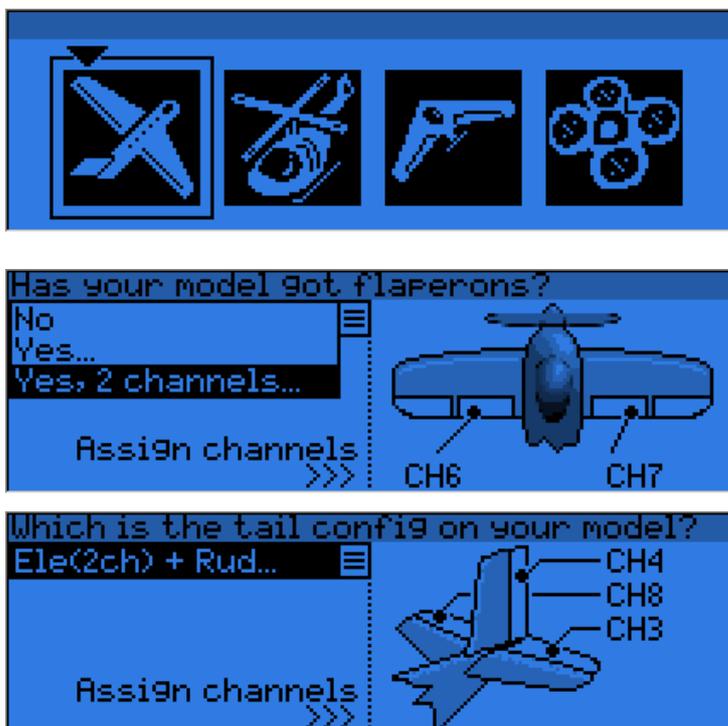
1. Modell Auswahl , neues Modell anlegen
2. Modell Einstellungen, Binden, Rangecheck, HF-Module
3. Helicopter Grundeinstellungen
4. Flugphasen, Flugmode, Flugbetriebsart
5. Inputs für Eingangsvorverarbeitung, z.B. Expokurve und Dualrate der Knüppel
6. Mischer (Das ist das wichtigste überhaupt, alles läuft über Mischer)
7. Limits, Servo-Wegeinstellungen, Begrenzungen und Servoumkehr, Servo-Reverse
8. Kurven definieren und eingeben
9. Globale Variablen Voreinstellungen in den Flugphasen
10. Logische Schalter, Programmierbare Schalter, Virtuelle Schalter, Softwareschalter
11. Spezial Funktionen, Funktions-Schalter
12. LUA Scripte aufrufen, LUA Interpreter starten,
13. Telemetrie mit einem Frsky -Modul

Neue Modell erzeugen mit dem LUA Script Modellgenerator

→Die Taranis hat keine Templates mehr, dafür gibt es LUA-Scripte.

LUA Scripte: Wird ein neues Modell angelegt, startet der geführte Modellgenerator.

Das ist ein LUA Interpreter und führt mit Abfragen durch die Varianten der unterschiedlichen Modelltypen, Flugmodelle, Helicopter, Deltamodelle, Multicopter usw.



LUA Scripts Verzeichnissbaum auf der SD-Karte ab V2.06

/SCRIPTS/

/SCRIPTS/BMP/

/SCRIPTS/WIZARD/ LUA Scripte+Bilder für Modellgenerator, neue Modelle erzeug.

/SCRIPTS/TEMPLATES/

/SCRIPTS/MIXES/

/SCRIPTS/FUNCTIONS

/SCRIPTS/<MODELNAME>/telemXX.lua um eigene Telemetrianzeigen zu erzeugen.

Mehr info zu LUA : <http://www.open-tx.org/lua-instructions.html>

LUA- Modellgenerator zum download: <http://www.open-tx.org/2014/06/02/lua-wizard/>

Das LUA-System muss man zur Zeit noch von Hand einrichten, damit es läuft.

Das **wizard.zip** ins Unterverzeichnis /SCRIPTS/WIZARD reinkopieren, dort entpacken
Dann startet der Modellgenerator wenn ein neues Modell angelegt wird.

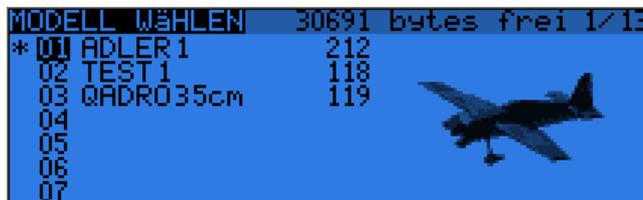
Achtung:

Dateiname 7-8 Zeichen, Dateityp 3 Zeichen, keine Sonderzeichen, keine Leerzeichen!

Modell auswählen, kopieren, verschieben, anlegen (1/13)

Es ist immer das Modell aktiv mit dem Stern * davor

Mit den Cursor-Tasten [+] [-] einen Modellspeicher anwählen und in Abhängigkeit ob der Platz frei oder belegt ist erscheinen mit [Enter Long] unterschiedliche Auswahlmenüs, mit [+] [-] auswählen, dann mit [Enter] bestätigen



Ein passendes Modellsymbol kann als BMP-Datei, Format 64x32Pixel 4 Bit angezeigt werden.
SD-Karte \BMP\...



Einzelnes Modell als **Backup** auf die SD-Karte unter /Models abspeichern
Siehe SD-Karte Unterverzeichnisse



Die Modellsymbole gibt es hier: <http://openrcforums.com/forum/viewtopic.php?f=92&t=3530> als *.zip Datei mit über 250 Symbolen aller Art. Oder selber erzeugen.



Achtung:
Dateiname 7-8 Zeichen,
Dateityp 3 Zeichen,
keine Sonderzeichen,
keine Leerzeichen!

Kleinr Auszug aus den BMP Dateien

Modell Grundeinstellungen Menüs Übersicht (2/13)

MODELL-EINSTELLUNGEN		2/13
Modellname		
Modellfoto	---	
Timer 1	AUS 00:00	
Name		
Permanent	AUS	
Minuten-Alarm	<input type="checkbox"/>	
Countdown	Kein	
Timer 2	AUS 00:00	
Name		
Permanent	AUS	
Minuten-Alarm	<input type="checkbox"/>	
Countdown	Kein	
Timer 3	AUS 00:00	
Name		
Permanent	AUS	
Minuten-Alarm	<input type="checkbox"/>	
Countdown	Kein	
Erv. Wege auf 150%	<input type="checkbox"/>	
Erv. Trim auf 100%	<input type="checkbox"/> [Reset]	
Dislay Trims	Was	
Trimmschritte	Fein	
---Gas-Kontrolle---		
Vollgas hinten?	<input type="checkbox"/>	
Gas-Timerquelle	Gas	
Gas-Leerlauftrim	<input type="checkbox"/>	
---Vorflug-Checkliste---		
Checkliste anzeigen	<input type="checkbox"/>	
Gas Alarm	<input checked="" type="checkbox"/>	
Schalter-Alarm	A↑B↑C↑D↑E↑F↑G↑	
Poti-Warnung	OFF	
Mittelst.-Pieps	SHGQ123LR	
Use Global Funcs	<input checked="" type="checkbox"/>	
---Internes HF-Modul---		
Modus	X18	
Ausgangs Kanäle	CH1-8	
Receiver No.	02 [Bind] [Range]	
Failsafe-Modus	Halte Pos.	
---Externes HF-Modul---		
Modus	PPM	
Ausgangs Kanäle	CH1-8	
PPM frame	22.5ms 300u -	

Ab openTx V2.1 zusätzlich:

- Trimmwerte anzeigen Nein, Ja, Kurz (bei Änderungen)
- Lehrer/ Schüler mehrere Signalquellen möglich
- Bei X9E Bluetooth aktivieren

Im Internen HF-Modul die Betriebsart einstellen

```

MODELL EINSTELLUNGEN 2/13
----Internes HF Modul-----
Modus X16
Ausgangs Kanäle CH1-8
Empfänger Nr. 01 [Bind] [Range]
Failsafe Mode Hold
    
```

D16 (X16) für X-Empfänger
Telemetrie mit Smart-Port Sensoren

D8 Modus für D- und V-II-Empfänger
LRS Long Range Modul

Bis 31.12.2014 ETSI Norm V1.7.1 gültig, deshalb D16, D8, LRS
 Seit 01.01.2015 ETSI Norm V1.8.1 gültig, deshalb in Europa nur noch D16, kein D8, kein LRS
 Umflashen auf alte Norm möglich.

Sendeleistung Normalbetrieb ca 92-95mW = ca 19,6dBm
Sendeleistung Rangetest und Binden 0,1 mW = -10dBm, also um 30dBm reduziert.

Im Externen HF-Modul die Betriebsarten einstellen

```

MODELL EINSTELLUNGEN 2/13
----Externes HF Modul-----
Modul-Typ PPM
Ausgangs Kanäle CH1-8
PPM Einst. 22.5ms 300u +
    
```

PPM
DSM2/DSMX / LP4 LP5 = Low Power
DJT
XJT in D16 (X16)/ D8/ LR12

```

Failsafe Mode Hold
Externes HF Modul
Module DJT
Channels Range CH1-8
Empfänger Nr.
Failsafe Mode
    
```

Trainer Mode Master (Lehrer)

oder Slave (Schüler) einstellen

```

MODELL EINSTELLUNGEN 2/12
Failsafe Mode Hold
Externes HF Modul
Module PPM
Channels Range CH1-16
PPM Einst. 22.5ms 300u -
Failsafe Mode
Trainer Mode Master
    
```

```

MODELL EINSTELLUNGEN 2/13
Modul-Typ PPM
Ausgangs Kanäle CH1-8
PPM Einst. 22.5ms 300u +
----Trainer Buchse Einst.-----
Modus Slave
Ausgangs Kanäle CH1-8
PPM Einst. 22.5ms 300u
    
```

Lehrer: Für PPM Input an der DSC-Buchse

Schüler: Für PPM Output an der DSC-Buchse

Ab openTx V2.1 gibt es mehrere Quelle für die Signale an den Lehrsender vom Schülerschüler
 Signale können dann nicht nur über die DSC-Buchse ausgegeben und eingespeist werden.
 Details bei der Hardware:

Anmerkung:

D16 bzw X16 ist das selbe, Je nach openTx-Version und Sprache nur unterschiedlich bezeichnet.

Modell Grundeinstellungen im Detail (2/13)

Hier gibt es sehr viele Optionen die mit den 2 Cursor-Tasten ausgewählt werden

1. **Name:** Modellname mit max 10 Zeichen.
 Editieren mit **[ENTER]**, dann wird das erste Zeichen invers dargestellt.
 Mit **[+]** / **[-]** kann man die Buchstaben, Zahlen usw. auswählen.
 Mit **[ENTER]** bestätigen und 1 Stelle weiter
 Die Umschaltung von Groß-/Kleinbuchstaben erfolgt mit **[ENTER Long]**
 Mit **[EXIT]** wird der Name übernommen.

- 1a. Ein **Modellbild** im BMP- Format mit 64*32 Pixel 4 Bit d.h. 16 Graustufen kann anstatt des FrSky Logo eingeblendet werden. Quelle ist die SD-Karte, Unterverzeichnis **BMP**

2. **Timer1:** Grundfunktion des Timer auswählen und Zeitwert eingeben
 Mit **[+]** / **[-]** springt man auf Minuten oder Sekunden
 Mit **[ENTER]** editieren, invers dargestellt.
 Mit **[+]** / **[-]** kann man Zeitwerte eingeben
 Und mit **[EXIT]** übernehmen.

- **Steht ein Wert von 00:00 drinnen läuft der Timer vorwärts**
- **Steht ein Wert von ungleich 00:00 drinnen läuft der Timer rückwärts.**

Trigger: Start des Timer, Timerfunktionen auslösen durch alle möglichen Schalter oder Geberstellungen. Mit vorangestelltem „!“ wird die Funktion umgekehrt, aus Schließer wird Öffner. Mit **[+]** / **[-]** die Funktionen auswählen.

- **AUS** - Timer ist ausgeschaltet.
- **ABS - EIN** -Timer ist immer ein.
- **GSs GS% GSt** – Timer in Abhängigkeit des Gasknüppels. Englisch: **THs / TH% / THt**
 „s“ bedeutet vom Stick, Knüppel, sobald Gas von Minimum wegbewegt wird startet der Timer, wenn er wieder auf Minimum steht stoppt er wieder. (**s** = start/ stop)
 „%“ die Zeit läuft proportional zur Gasstellung, d.h. bei Halbgas langsamer als bei Vollgas, das ist ganz praktisch um Flugzeiten einschätzen zu können (Tank oder Akku leer).
 „t“ die Zeit startet sobald einmal die minimum Gasstellung verlassen wurde und stoppt dann aber nicht mehr im Gegensatz zu (**t**= trigger)

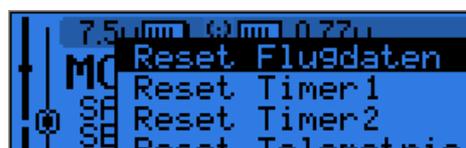
Switches – man kann auch alle beliebigen, physischen und virtuellen Schalter auswählen um den Timer zu starten/ stoppen.

Dabei bedeutet das „**t**“ z.B. bei **TRNt** einen „Momenten“- Impuls-Schalter. D.h. einmal kurz umschalten für Timer **EIN**, nochmal kurz umschalten Timer wieder **AUS**,
 oder auch Toggle Funktion „**t**“
 „**t**“ =Toggle: Ab OpentxV2.00 wird es ersetzt durch **SRFF** ein Set-Reset FlipFlop

(das ist nicht schwer, einfach mal ausprobieren auch das NOT „!“ geht)

Timer rücksetzen, setzen, neu starten

Timer kann man im Hauptmenü mit **[Enter Long]** oder in den Spezialfunktionen rücksetzen/Setzen



- 2a. **Modell- Zeit** Modell Laufzeit „P“ ermanent aufsummieren und speichern „P“= **Persistent**
Aus, Flight = Flugzeit startet mit **GSt**, Nur Manual Reset
Gesamtzeit kann man sich im StatistikMenü ansehen **TOT** = Total



- 2b. **Jede Minute** nach jeder Minute kommt ein Ansage
- 2c. **Countdown** Timer 30, 10, 5, 4, 3, 2,1, 0 Sekunden Signalton
Ansage-Auswahl ist: Stimme, Pieps, Ohne
3. **Timer2 und Timer 3**: Alles gleich wie bei Timer1 (ab opentx V2.1 gibt es 3 Timer)
4. **Erw.. Limits**: Extended Limits, Bereichserweiterung von +/-100% auf +/-150%
Impulslänge der PPM Signale. (-100%=1ms 0%=Mitte=1,5ms +100%=2ms)
Hier aufpassen, dass man die Servos nicht an ihre mechanischen Grenzen fährt und beschädigt.
Also nicht nur auf das Display schauen, sondern im Servo-Menü auch die Wege begrenzen.
+/-100% = 1500us +/-512us +/-125%= 1500us +/-640us +/-150%=1500us +/-768us
5. **Erw.. Trims**: Erweiterte Trimmwerte. Bereichserweiterung von +/-25% auf +/-100%.
Normal sind die 4 Trimmwerte auf 1/8 des max Weg/Servoweg begrenzt.
Das ist auch der Bereich der Trimbalken am Display.
Wenn der Trimmwert von über 1/8 (von +/- 125%) überschritten wird kommt ein kurzer Piepser und der Trimmcursor bleibt stehen, dann kann man nochmal die Trimmtasten drücken und die Trimmung geht weiter.
Besser ist es jedoch die Mechanik am Servogestänge anzupassen.
6. **Trimm Schritte Trim Inc.**: Trimmstufen, Trimmsschritte, Feintrimmung
- **Exp** – Exponential: Um die Mitte sind ganz feine Trimmstufen, je weiter weg von der Mitte desto größer werden die Trimmstufen
 - **ExFein** – Extra Fine, 1 Schritt pro Klick. = 1 us
 - **Fine** -Fine 2 Schritte pro Klick = 2us
 - **Mittel** -Medium (bevorzugt), 4 Schritte pro Klick = 4us
 - **Grob** - Coarse, 8 Schritte pro Klick = 8us
7. **Vollgas hinten, Thr(ottle) reverse**: Für besondere Leute die **Vollgas** nicht vorne, sondern **hinten** haben. Dadurch werden auch sämtliche Funktionen die mit der Gasstellung zu tun haben gedreht.
Also: Warnung Gasknüppel nicht Null, die Gasleerlauftrimmung, Motor in der Leerlaufstellung fahren usw.
8. **GasTimerQuelle (T-Trace T-Source)** Auswahl von welchem Geber kommt die Gasstellung für die Ansteuerung der Timerfunktionen **GSs, GS%, GSt (Ths, TH%, Tht)**, Damit kann auch eine andere Quelle als nur der Gasknüppel **Thr** den Timer triggern z.B. Throttle Cut per Schalter.
- **Gas** -Thr – Vom Gasknüppel (normalerweise)
 - **S1,S2 LS, RS** – von einem anderen Analoggeber, Potentiometer
 - **CH1 .. CH32** – von einem Ausgangs-Kanal
9. **Gas Leerlauf Trim T-Trim**: Gas Leerlauftrimmung aktivieren. Dabei passieren mehrere Dinge.
Die Trimm-Mitte wird in Richtung Leerlauf versetzt. Die Trimmung arbeitet nur wenn der Gasknüppel unterhalb der Mitte ist. Damit kann man den Leerlauf eines Verbrennungsmotors fein einstellen und ihn auch abstellen ohne die Vollgasstellung zu beeinflussen.
(Vollgasstellung macht man dann im Servoeinstellmenü Limit/Subtrim7/12)

10. **T-Warning:** Warnung Gasknüppel nicht in Leerlaufstellung, kommt beim Einschalten des Senders und gibt kein Ausgangssignal an den Empfänger ab bis der Alarm aufgehoben wird, d.h. Gasknüppel auf Leerlauf ziehen, oder per Taste quittiert wird.
11. **Schalter Alarm:** Wenn ein Schalter nicht in der vordefinierten Stellung ist, kommt beim Einschalten des Senders eine Warnung und gibt kein Ausgangssignal ab bis der Alarm aufgehoben wird.

Vordefinition in den Modelleinstellungen:

Jeder Schalter kann **auch einzeln** überwacht werden. Der jeweilige Schalter wird dann mit einem der drei Stellungssymbole **↑↓→** neben dem Schalter dargestellt.

Schalter ohne Symbol werden nicht überwacht (hier wird D E F nicht überwacht).



Wenn alle zu überwachenden Schalter ausgewählt sind dann und auf **<**] gehen und mit **[Enter Long]** werden jeweils die aktuellen Schalterstellungen übernommen.

Poti Alarm: als OFF, Man und Auto für **S1,S2, S3, LS, RS**

Mit **Auto** werden beim Ausschalten des Senders oder beim Modellwechsel die jeweilige aktuellen Positionen gespeichert

Mit **Man** kann jedes Poti in beliebiger Stellung überwacht werden

Poti einstellen, dann mit **[Enter Long]** und einem kurzen Pieps wird der Wert gespeichert

12. **Beep Ctr:** hier kann man einstellen ob bei Mittelstellung der Analoggeber eine Ansage oder kurzer Piepser kommen soll. **RETA12LR** bedeutet **Rud, Ele, Thr, Ail, Poti S1, S2, Geber LS RS**
Deutsch **SHGQ12LR= Seite, Höhe, Gas, Querruder, Poti S1,S2, Geber LS RS**
Aktiv ist was invers dargestellt wird. Mit den Cursern **[+] / [-]** auf den Buchstaben/Zahl gehen, mit **[ENTER]** An- oder abwählen. Überwacht wird was invers dargestellt wird. Das ist recht praktisch bei den Potistellungen ohne draufschauen zu müssen.

Tip 1: Wenn eine wav-Ansagedatei auf der SD-Karte nicht gefunden wird, dann kommt automatisch nur ein Pieps. Somit reicht es aus eine Ansagedatei nur umzubenennen, nicht löschen, wenn sie stört. Mich nerven manche dieser Ansage, Pieps ist ok.

„**Poti zentriert**“ SD-Karte Sounds/de/System/midpot.wav umbenennen
„**10sec, 10Sec,...**“ SD-Karte Sounds/de/System/ timer10.wav umbenennen

Tip2: Wird ein anderes Modell angewählt und gibt es dazu eine wav-Datei mit **exakt gleichem Namen** wie das Modell auf der SD-Karte im Unterverzeichnis /MODELS so wird diese wav-Datei automatisch beim Aufruf des Modells **einmal** abgespielt. So kann man sich das Modell oder andere Dinge ansagen lassen.

Tip3: Will man sich Schalterstellungen ansagen lassen, so kann man die in den Spezialfunktionen aufrufen. Einmal mit **1x** oder alle **X**-Sekunden usw.
Was stört sind die Ansagen schon beim Modellaufruf.
Das kann man mit **!1x (Not Ein mal)** ausblenden, „Ansagen, aber nicht beim Modellaufruf“

HF Module und Protokolle, Bind, Failsafe,

Internes HF Modul

Der Sender Taranis hat ein eingebautes HF-Module vom Typ **XJT**.

Dieses **XJT**-HF-Module kann in 3 Betriebsarten betrieben werden und ist damit mit allen alten und neuen FrSky Empfängern kompatibel

D16, X16 bis 16 Kanäle, Telemetrie für **SmartPort-Sensoren** und **X-Empfänger**

D8 bis 8 Kanälen, Telemetrie für **Hub-Sensoren** und alle **D- und V-II-Empfänger**

LRS 9/12 Kanäle im **Long Range Mode** ohne Telemetrie

Die Anzahl der Kanäle 1-16 und der Kanalbereich (z.B. Kanal 4-8) kann frei eingestellt werden.

Empfänger Nummer, Bind-Funktion, Range und Failsafe Mode kann eingestellt werden.

Externes HF-Modul

Im Schacht auf der Rückseite kann ein Modul mit JR-Modul Maßen eingebaut werden.

Das kann alles möglich sein, denn auch die Protokoll für diese Module können eingestellt werden (wird noch erweitert!).

Ein weiteres FrSky Modul **DJT** oder **XJT** Modul mit PXX Protokoll

PPM Modulation für diverse Fremd-Module

DSM2/DSMX div.Spektrum-Module mit PPM bzw mit serieller Schnittstelle DSM2 DSMX

LP45 Spektrum DSM2 Low Power Module LP4, LP5 4 und 5 Kanal

FASST, HOTT, FlySky, Multiplex M-Link, Jeti, Sanwa, Assan, Corona, usw.

Wird ein weiteres XJT Modul verwendet können noch mal 16 Kanäle übertragen werden.

Damit hat man echte $2 \times 16 = 32$ Kanäle mit gleicher Modellmatch-Nummer.

Die Anzahl der Kanäle 1-16 und der Kanalbereich (z.B. Kanal 4-8) kann frei eingestellt werden. Auch hier kann eine Empfänger- Nummer, Bind, Range und der Failsafe Mode eingestellt werden.

Das interne oder das externe oder beide HF-Module können gleichzeitig aktiv sein!

Failsafe Mode einstellen

Es gibt 4 Arten von Failsafe Einstellungen

Halte Position = halten der letzten gültigen Kommandos, Servos behalten ihre Position bei

Angepasst [Set] = anfahren von einzeln voreingestellte Servoeinstellungen, Gas, Ruder usw.

Kein Signal = das kann ein Flight Controller erkennen und dann darauf reagieren.

Empfänger = die im Empfänger per F/S -Taste gespeicherten Werte verwenden

FAILSAFE SETTINGS			
Ail	120.0	CH9	0.0
Ele	1.8	CH10	0.0
Thr	-100.0	CH11	0.0
Rud	3.5	CH12	0.0
Gear	-100.0	CH13	0.0
Flap	100.0	CH14	0.0
CH7	0.0	CH15	0.0
Cam	0.0	CH16	0.0

Failsafeanzeige

(Bis openTx V2.017)

Failsafe setzen			
CH1	HOLD	CH9	0.0
CH2	9.8	CH10	0.0
CH3	HOLD	CH11	0.0
CH4	-15.9	CH12	0.0
CH5	NONE	CH13	0.0
CH6	HOLD	CH14	0.0
CH7	14.0	CH15	0.0
CH8	SETR	CH16	0.0

Failsafeanzeige

(Ab openTx V2.1)

jeder Kanal einzeln einstellbar

Hold, None, Wert, Empfänger

Mit **[ENTER]** in der Funktion **Angepasst [Set]** kann für **jeden einzelnen Kanal** eine Failsafe Position programmiert werden. Mit **[ENTER LONG]** wird der Wert übernommen.

In der Anzeige erscheint pro Kanal zusätzlicher Text **NONE, SET**, bzw der gespeicherte Wert.

13. **Proto**: Protokoll auswählen, Sender Protokoll auswählen (für externes Modul!)

- **PPM** – das ist das normale PPM Signal- Protokoll das an das HF-Modul geht.
Man kann die Anzahl der Kanäle 4,6,8,10,12,14 bis 16 auswählen. Das ist praktisch für Systeme/Empfänger die nicht mehr Kanäle verarbeiten können.
- **PPM16** Ausgabe Kanal 1-8 über das HF-Modul, Kanal 9-16 über die DSC Buchse
- **PPMSim** 8Kanäle an die DSC-Buchse für PC Flugsimulator, keine HF-Abstrahlung
- **PXX** Das ist ein serielles Protokoll von FrSky für diese Module (sehr umfangreich!)
XJT 16 Kanal Protokoll
Num RX: Empfängernummer für die Model Match Funktion
Sync und Failsafe Definition
- **DSM2** serielles Protokoll für Spektrum-Module
Binding: TRN-Taste halten dann Power Ein. **Wichtig**: Splashscreen Aus und keine Warnungen aktiv, sonst funktioniert das Binden nicht!
LP4/LP5: für HP6DSM (LP4DSM2) Module mit kurzer Reichweiten (Short Range)
DSMonly: DSM2 only Übertragungsart festlegen
DSMX: automatische Auswahl der Übertragungsart DSMX/DSM2
NumRX: Empfängernummer für Modell match
RANGE: Auswahl und [MENU] Rangetest starten, beenden [MENU] und [Exit]

Failsafe im Detail: XJT-Modul im X16 / D16-Mode für X8R, X6R, X4R, LR12

Alte Anlagen hatten max 2 Failsafe-Mode im Empfänger:

Hold Last Command oder im Receiver fest abgespeicherte Positionen

Heute wird fast nur noch Custom Setting verwendet, Hold ist eigentlich immer falsch.

Die Taranis hat 4 Failsafe Modes. Die kann man im D16-Mode des XJT-HF-Moduls im Sender direkt einstellen. **Hold , Fester Wert , NONE No Pulses, Receiver**

Einstellen unter: Modelle 2/13 , Internes HF-Modul, D16-Mode, ganz unten

Alle 9 sec überträgt der Taranis-Sender Failsafe Modes und eventl. Einstellungen zum Empfänger. Also min. 9 sec warten bevor man zu einem Failsafetest den Sender ausschaltet.

1. Hold Last Command: (eigentlich veraltet, aber wg Kompatibilität zu D8 Mode)

Wenn man nichts macht, ist bei einem neuen Modell Hold aktiv. Der letzte gültige

Servo-Datensatz wird im Empfänger gehalten. Die Taranis überträgt nur den Befehl **Hold**

2. Fester Wert

Abgespeicherte Failsafe Positionen im Sender werden alle 9sec an den Empfänger neu übertragen und dort gespeichert. Taranis überträgt Befehl **Custom Setting** und die Positionen der **16 Kanäle**.

Am Sender einstellen: [Set], dann auf Kanal gehen, mit [Enter] Kanal blinkt,

Wert am Kanal eingeben [Long Enter] bis es Piepst, nächster Kanal usw.

3. NONE, No Pulses (nichts ausgeben)

Ausgang wird weggeschaltet auch keine 0,0% = 1500us = Servo Mitte

Spezielle Flugcontroller erkennen einen Systemausfall des Empfängers und reagieren selbständig.

Die Taranis überträgt nur den Befehl **No Pulses**

(der S-Bus hat kein No Pulses, aber 2 Fehlerbit die man auswertet, "Bad Frame", "Failsafe")

4. Receiver: (eigentlich veraltet, aber wg Kompatibilität zu D8 Mode)

Wird wie früher am Empfänger direkt per Hand und F/S-Taste fest gespeichert.

Die Taranis überträgt nur den Befehl **Receiver**. D.h. die Taranis muss auch auf **Receiver** stehen, damit die im Empfänger direkt gespeicherte Werte nicht durch andere Failsafe-Modes überschrieben werden.

Hinweise:

Wird ein Empfänger neu gebunden hat er erst mal intern Hold

Wird ein neues Modell in der Taranis angelegt, steht die Taranis auf Hold.

Wird ein Empfänger von Hand per F/S auf Failsafe gesetzt, nimmt er die Werte die von der Taranis kommen und speichert sie ab. Da die Taranis ihm aber weiterhin Hold sendet, wird er Hold ausführen. Deshalb die Taranis dann vorher auf Receiver stellen und dann sind die Werte im Empfänger aktiv.

1,2,4 Failsafe-Mode: die Servo PWM-Werte werden auch so auf den S-Bus gegeben.

Senderoption **ppmus** ist sehr praktisch, man sieht sofort die PPM-Werte in us die ein APM braucht.

Merke: 0,0% = 1500us = Mittelstellung des Servo

Youtube-Video: <http://www.youtube.com/watch?v=gj-MqyvbQ5I>

Binden des Senders mit internem HF- Moduls an den Empfänger

Taranis hat ein **Modellmatch** integriert, d.h. jeder Empfänger erhält beim Binden eine eigene Nummer zugeordnet und reagiert dann nur noch auf diese Nummer.

Ansonsten geht das binden so wie bei allen 2,4GHz-Systemen:

Abstand Sender zu Empfänger min ca 1m einhalten!

1. Am Sender:

Modelleinstellungen 2/13, Internes HF-Modul

Empfänger Nr xx einstellen, (Normal die Modellnummer verwenden, aber nicht die 00) dann [**Bind**] aufrufen, ein Fenster mit den RSSI Werten erscheint und der **Sender piepst**.

2. Am Empfänger:

Taste F/S gedrückt halten und einschalten,

Nach 1-2 sec erkennt man an der LED dass der Empfänger gebunden hat.

F/S Taste loslassen, Empfänger ausschalten.

Dann am Sender das Binden von Hand beenden

[**RANGE**] Reichweitencheck kann man dann auch gleich hier machen,

Dabei sendet der Sender mit vermindeter Leistung 0,1mW = -10dBm

Am Empfänger X8R, X6R kann man vor dem Binden noch per Jumper einstellen welche Kanäle (1-8 oder 9-16) an den Servo-Steckern ausgegeben werden sollen → siehe Empfänger Handbuch

Binden des Senders mit externem HF- Moduls an den Empfänger

Wenn das externe Modul ein XJT -Typ ist, dann wie beim internen Modul binden.

Ansonsten haben alle externen Module in der Regel einen Binde-Knopf, der gedrückt und gehalten werden muss bevor der Sender eingeschaltet wird.

Mehrere Empfänger an ein Modell binden

Auch das geht. Damit kann man 16 Servos direkt anschließen ohne S-Bus oder PWM-Decoder

Nur 1 Empfänger Telemetrie übertragen entsprechend Jumpern

Empfänger 1 Kanal 1- 8, mit Telemetrie

Empfänger 2 Kanal 9-16, ohne Telemetrie

Beide Empfänger die gleiche Modell-Match Numern zuweisen.

Mit einem Sender gleichzeitig 2 Modelle steuern

Auch das geht. Das wird vor allem im Funktionsmodellbau angewendet.

Nur 1 Empfänger darf Telemetrie übertragen, entsprechend Jumpern

Modell 1 Empfänger 1 Kanal 1-8, mit Telemetrie per Schalter Kanal 1-8 freischalten

Modell 2 Empfänger 2 Kanal 9-16, ohne Telemetrie per Schalter Kanal 9-16 freischalten

Beide Empfänger die gleiche Modell-Match Nummer zuweisen.

Dabei muss nicht mal das Modell im Sender gewechselt werden!

Reichweitentest durchführen und zu erwartende Werte mit X9E und X6R

Oft stellt sich die Frage was ist eine „gute“ „schlechte“ Reichweite mit einem Empfänger, ist das noch ok, was kann man für RSSI-Werte erwarten. Hier mal ein Ablauf mit konkreten Werten:

RSSI -Werte bei Normalbetrieb ablesen, Sendeleistung ca 92-95mW = ca 20dBm
RSSI -Werte bei Rangetest ablesen, Sendeleistung ca 0,1 mW = -10dBm,

Empfänger: X6R auf Holzstab 2m hoch, beide Antenne 90° zueinander, horizontal, kein Metall, keine Kohle, frei stehend.

Sender: X9E OpenTx V2.1.2 ETSI V1.7.1 bzw ETSI V1.8.1 (ist für den Test egal)
 Am Sender, in der Telemetrie SWR und RSSI-Werte aktivieren und zur Anzeige bringen.
 SWR muss 0 bis 1 anzeigen! Normale RSSI-Werte in ca 2m Abstand zum Empfänger ca 94dBm

Entfernung zum Empfänger ca 150-170m:

freie Sicht, keine Hindernisse, Wiese, keine anderen Sender imUmfeld
 Sender am Bauch zum Empfänger hin

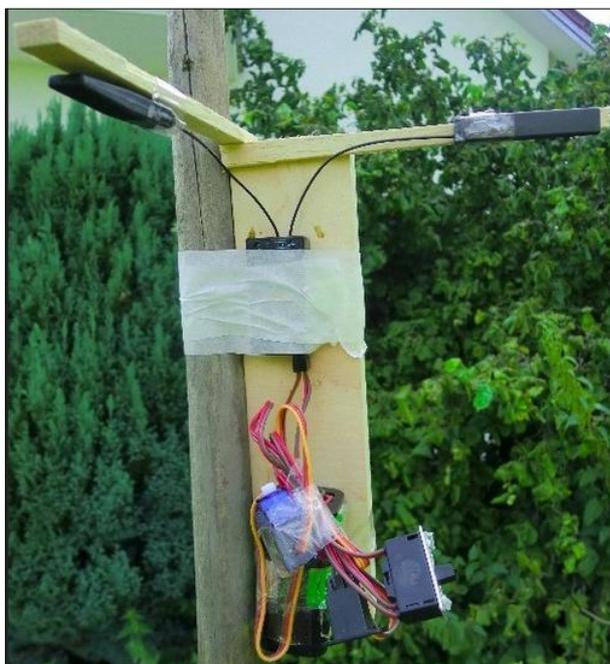
Dann in die Modelleinstellungen den Rangetest (neben dem Binden) aktivieren
 RSSI-Werte für reduzierte Sendeleistung (0,1mW) ablesen.

Abgelesene RSSI-Werte bei:

Normalbetrieb RSSI ca 74-78dBm (normaler Sendeleistung ca 92-93mW = ca 19,6dBm)
 Rangetest RSSI ca 46-48dBm (reduzierter Sendleistung ca 0,1mW = -10dB)

Das sind Werte unter optimalen Bedingungen. Je nach Einbau im Modell, Metall, Kohlefasern, Abschirmungen können sie stark schwanken.

Wenn man sich vom Empfänger wegdreht wird der RSSI-Wert um ca 3dB sinken.



-40 dBm	-70 dBW	100 nW	0.0001 mW
-30 dBm	-60 dBW	1 µW	0.001 mW
-20 dBm	-50 dBW	10 µW	0.01 mW
-10 dBm	-40 dBW	100 µW	0.1 mW
-1 dBm	-31 dBW	794 µW	0.794 mW
0 dBm	-30 dBW	1.000 mW	1.000 mW
1 dBm	-29 dBW	1.259 mW	1.259 mW
10 dBm	-20 dBW	10 mW	10 mW
20 dBm	-10 dBW	100 mW	100 mW
30 dBm	0 dBW	1 W	1000 mW

Sendeleistungen des XJT-Moduls:

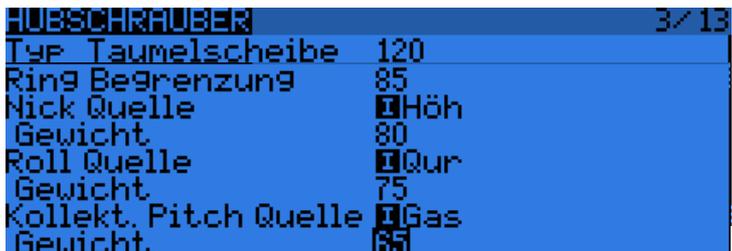
Normale Sendeleistung +20dBm = 100mW
 Rangetst und Binden -10dBm = 0,1mW
 Sendeleistung wird also um 30dBm reduziert

(Test mitX6R, da kommt noch ein schöneres Bild rein)

Hubschrauber Grundeinstellungen (3/12)



Bis V2.017



Ab V2.10 neue Möglichkeiten:

Die Eingaben vereinfachen das Einstellen der Taumelscheibewerte sehr.

Die Helimischervariablen CYC, CYC2, CYC3 enthalten dann die fertigen TS-Werte. Mit diesen dann in den Mischer weiterarbeiten.

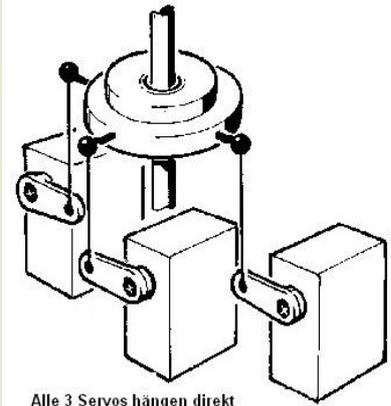
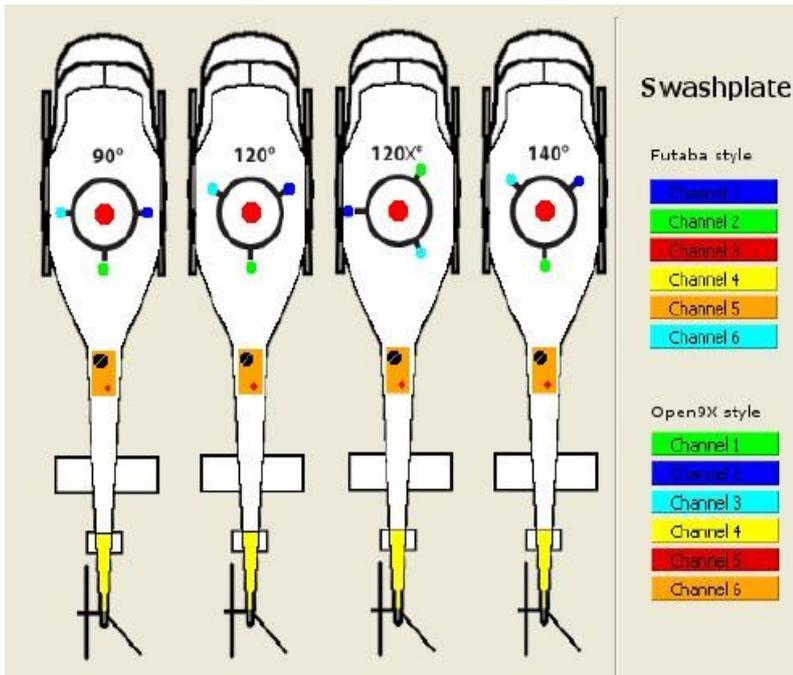
Gewichtung: +100% bis -100% (Signalumkehr hier mit neg Gewichtung machen!)

Hier werden die Grundeinstellungen für Hubschrauber gemacht:

Die Idee dahinter ist, dass man **hier** alle benötigten Taumelscheibeneinstellungen macht.

Taumelscheibentyp, Kollektiv-Pitch-Quelle, Zyklische Begrenzungen (Swash Ring), Servolaufrichtungen und damit vorberechnete Heli-Mischer-Variablen **CYC1, CYC2, CYC3** erhält die man dann beliebigen Kurven und Mischer und Schalter zuordnen kann.

1. **Swash Type:** Typ der Taumelscheibe, Art des Kopfes am Hubschrauber:
 - 120: "Standard" Kopf mit 120 °. Das "Pitch" Servo ist vorne oder hinten
 - 120X: auch 120 ° aber um 90 ° gedreht, das "Pitch" Servo ist an einer Seite.
 - 140: Kopf mit 140 ° - mit "Pitch" Servo vorne oder hinten.
 - 90: Ein einfacher Kopf mit 90° Aufteilung. 1 Servo für Pitch und 2 Servos für Roll
2. **Collective:** Kollektives Pitch Quelle von wo Coll. Pitch gesteuert wird, Kanal, Geber....
Die Idee dahinter ist, dass man Mischer erstellen kann, die alle benötigten Kurven und Schalter beinhaltet, dann hier verknüpft wird, damit der Rest gemischt werden kann.
3. **Swash Ring:** Zyklische Kopf-Begrenzung , Kreisförmig, für Roll und Nick-Achse in %
4. **ELE / AIL / COL Invert:** **Nick, Roll, Pitch** Eingangsfunktionen invertieren damit die Servo Bewegungen am Kopf richtig funktionieren und die Berechnungen und nachfolgenden Mischer richtig arbeiten.

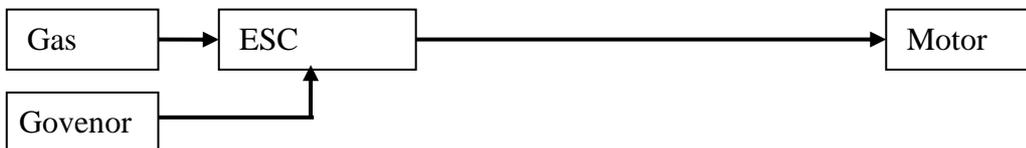
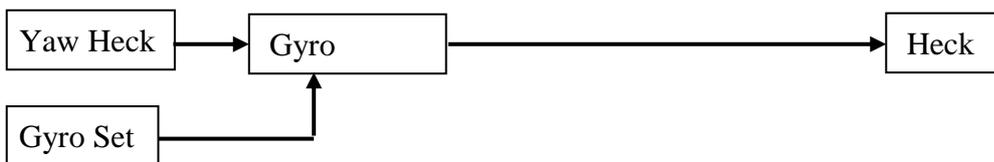
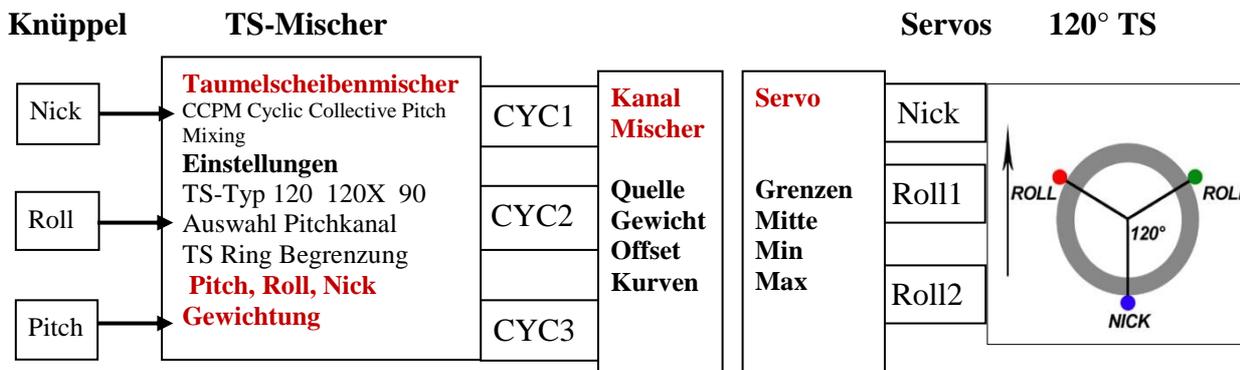


Alle 3 Servos hängen direkt unterhalb der Taumelscheibe

Übersicht Taumelscheibenmischer für 120° Taumelscheibe

Die 3 Knüppel für Pitch, Roll, Nick werden im TS-Mischer vorberechnet. Es kommen die 3 TS-Werte CYC1, CYC2, CYC3 raus. Nur diese werden in den Kanalmischern verwendet. Über die Servobegrenzungen gelangen sie an die Ausgänge und zu den 3 TS-Servos Nick, Roll 1, Roll 2 der 120° Taumelscheibe.

Servoinvers darf nur im TS-Mischer gemacht werden, damit die Berechnungen stimmen!



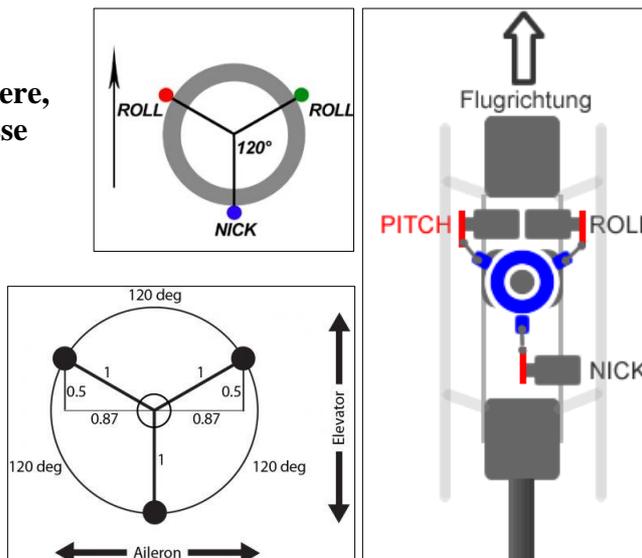
Leider hat jeder Sender-Hersteller eine andere, meist dumme Bezeichnung für die Anschlüsse

Mögliche Kanalbelegung für Taranis

- 1 CYC2 Roll 1 (rechts) Quer1, Ail1
- 2 CYC1 Nick (hinten) Höhe, Elev
- 3 Throttle ESC/GAS
- 4 Yaw Heck Seite, Rud
- 5 Gyro set Heading Hold
- 6 CYC3 Roll 2 (links), Pitch, Ail2
- 7 Govenor ESC Gov

Hier erkennt man auch wie die Mischeranteile 1,0 0,87 0,5

bei einer TS mit 120° entstehen. (Eine 135° TS hat eine bessere, symmetrischere Verteilung)



Viele fertige Hubi-Setups für Taranis findet man hier: <http://rcsettings.com/>

Flugphasen / Flugmode definieren (4/13)

FLUGPHASEN						4/13
FP0	(Normal)					0.0 0.0
FP1	SA-	1	1	1	1	0.0 0.0
FP2	SA↓	2	2	2	2	0.0 0.0
FP3	---	3	3	3	3	0.0 0.0
FP4	---	4	4	4	4	0.0 0.0
FP5	---	5	5	5	5	0.0 0.0
FP6	---	6	6	5	5	0.0 0.0

Diese Art der Flugphase sieht zunächst recht einfach aus im Vergleich zu anderen Sendern.

Aber das ist **nur die Grund-Definition** der Flugphasen für Name, den Aktivierungs-Schalter, die Trimmwerteübernahme und die Übergabe-Zeiten für Fade-In, Fade-Out

Das wesentliche geschieht in den Mischern und in den Inputs-Menüs

Dort werden die eigentlichen Flugphasen ausgewählt/aktiviert (012345678) und die verschiedenen Werte für jede Flugphase eingestellt.

Damit ist das sehr viel umfangreicher als alles was man sonst von anderen Sendern kennt.

Hier im 4/13 werden nur die Flugphasen definiert. Es gibt 8 (9) Flugphasen zur Auswahl.

Flugphase FP0 Normal (Default) ist aktiv, wenn keine andere Flugphase aktiv ist.

Jede erhält einen Namen und erscheint dann im Hauptbildschirm wenn sie aktiviert wird.

Die Flugphase FP1 hat die höchste Priorität, FP8 die niedrigste Priorität

Falls gleichzeitig mehrere Flugphasen aktiv sind hat die höhere Priorität Vorrang und löscht die FP mit der niedrigeren Priorität.

In der Statuszeile (erste Zeile) wird angezeigt was zu tun ist

1. **Name:** hier den Namen eingeben, max. 6 Zeichen lang, (editieren so wie beim eingeben des Modellnamen)
Der Name der aktiven Flugphase erscheint dann in der Hauptanzeige
2. **Switch:** Schalter mit dem diese Flugphase aktiviert wird
3. **Trims:** Jede Flugphase kann ihre eigenen Trimmwerte haben und aktiviert werden.
(Ruder / Elevator / Throttle / Ailerons) Deutsch (Seite/Höhe/ Gas/ Quer)
Beispiel: Flugphase FP1, Name TakeOff, mit Schalter SA↓ aktivieren,
4. **Fade In:** Einleiten, dies erlaubt einen sanften, langsamen Übergang von einer Flugphase zur nächsten und verhindert ruckartige Flugbewegungen, Übergänge und Ruderbewegungen.
Werte bis 15 Sekunde sind möglich.
5. **Fade Out:** das Gleiche nur für das Ausschalten der Flugphase

Mit [Exit] zurück in das Aufrufmenü

Trimmung bei Flugphasen / Flugmode

FLUGPHASEN						4/13	
FP0	(Normal)					0.0	0.0
FP1	SA-	:1	:1	:1	:1	0.0	0.0
FP2	SA↓	:2	:2	:2	:2	0.0	0.0
FP3	---	:3	:3	:3	:3	0.0	0.0
FP4	---	:4	:4	:4	:4	0.0	0.0
FP5	---	:5	:5	:5	:5	0.0	0.0
FP6	---	:6	:6	+5	:5	0.0	0.0

Höhe Trim						4/13	
(Normal)						0.0	0.0
---	:1	:1	:1	:1		0.0	0.0
---	+0	+0	+0	+0		0.0	0.0
---	:0	:0	:0	:0		0.0	0.0
---	:0	:0	:0	:0		0.0	0.0
---	:0	:0	:0	:0		0.0	0.0
---	:0	:0	:0	:0		0.0	0.0

Hier muss man höllisch aufpassen, dass man beim Umschalten von Flugphasen auch tatsächlich schon Trimmwerte hat!

Trimmungen müssen erfolgen werden!

Jede Flugphase braucht seine eigene Trimmung!

Jede Flugphase hat 4 Trimmwerte für Gas, Quer, Höhe, Seite! (siehe Statuszeile)

Werden Flugphasen verwendet, ohne dass man etwas einstellt, so hat zuerst mal jede Flugphase die 4 Trimmwerte der **FP0**. Das erkennt man daran dass überall die gleiche Nummer steht. Also z-B. bei **FP2 :0 :0 :0 :0** oder **FP4 :0 :0 :0 :0**

Da hat man dann zumindest schon mal eine Trimmung erfolgen.

Die aber nicht passen wird.

Oder: Man kann man die Trimmungen der aktiven Flugphase übernehmen.

also für FP4 **FP4 :4 :4 :4 :4**

Da steht aber dann zuerst mal die Trimmung auf Mitte, also Null.

Oder: Man kann die Trimmugwerte von beliebig anderen Flugphase übernehmen

z.B. **FP1 :3 :1 :0 :7**

Auch da muss man aufpassen was jeweils in den anderen FPx drinnen steht

Oder: Man kann mit einer Kombination von Grundwert und Offsetwerten von beliebigen Flugphasen arbeiten, das erkennt man am Pluszeichen +

z.B. **FP2 +0 +0 +0 +0**

d.h. Flugphase 2 übernimmt die 4 Trimmwerte der FP0 als Grundwert und addiert dazu die eigenen dazu

Aber verändert sich die Trimmung von FPx dann verändert sich die Trimmung von FPy

Oder: beliebige Kombinationen aus beiden Varianten für jede einzelne Trimmung

:Trimmung + Offsettrimmung

Am Besten:

Für jede Flugphase seine komplett eigene Trimmung erfliegen!

Also FP2 :2 :2 :2 :2 oder FP4 :4 :4 :4 :4

Inputs als Signalvorverarbeitung (5/13)

Dieses Menü und Untermenü sieht ähnlich aus wie das Mischermenü und kann auch ähnliche Dinge. Durch die 2-stufige Signalverarbeitung mit Inputs und dann auf die Mischer, erreicht man einen höheren Grad der Flexibilität. Auch hier sind bis zu 64 Zeilen möglich.

Alle Geber, Kanäle, alle Telemetriewerte können, **müssen aber nicht**, vorverarbeitet werden. 4 Kreuzknüppel, 4 Trimmungen, 4 Potis, 8 Schalter, Max, alle 32 Kanäle CH1-CH32, PPM1-PPM16 bzw TR1-TR16 und auch alle Telemetriewerte können angepasst werden.

Hier werden auch die **Expokurven** eingegeben und die **Dualrate-Umschaltungen** gemacht. **Merke:** Expo und Dualrate sind Funktionen des Knüppels, also praktischerweise hier eingeben. Differenzierungen sind Funktionen des Ruders, also in den Mischern eingeben.

Das vorangestellte inverse **I** bedeutet immer, dass ein Signal über die Vorverarbeitung lief. **Das begegnet uns dann bei den Mischern wieder.**

Eine **Fette Darstellung** in der Zeile bedeutet dass diese Zeile aktiv geschaltet ist.

Die erste Zeile "8/64" bedeutet, dass 8 von 64 Inputzeilen benutzt werden.

```

INPUTS 8/64 5/13
IGas 100 Gas --- 012345678
IQue 100 Que --- 012345678
IH12 100 Höh E54 SA↑ 012345678 HochDiff
      80 Höh E35 SA- 012345678
      65 Höh E40 SA↓ 012345678
      -18 S2 --- 012345678
ISei 100 Sei --- 012345678
    
```

Mit **[Enter Long]** kommt man ins Untermenü

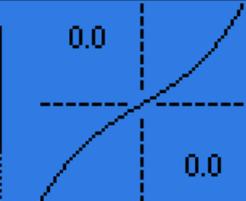
```

INPUTS 8/64 5/13
IGas Zeile Editieren
IQue Neue Zeile davor
IH12 Neue Zeile danach Diff
      Zeile kopieren
      Zeile verschieben
ISei Zeile löschen
    
```

Das Untermenü zur Werteingabe

```

INPUTS IQc1
Input Name Qc1
Line Name Querrude
Quelle Que
Gewicht 100
Offset 0
Kurve Expo 40
Phasen 012345678
    
```



Der Input-Name erscheint in den Mischern (hier Qc1)

Der Line-Name ist für die Bezeichnung rechts (Querrude) eine Infos-Zeile

Die Signal-Vorberechnungen erfolgt nach der Formel:

$$[(\text{Quelle} * \text{Gewichtung}) + \text{Offset}] = \text{Output} \rightarrow \text{Trimung} \rightarrow \text{Kurve}$$

Das ist die gleiche Formel wie in den Mischer

(Dadurch ergeben sich einfachere Möglichkeiten der Bereichsanpassung).

Dann laufen diese Werte durch Kurven und Funktionen, Expokurven, Fertige Funktionen, eigene Kurven und können für Flugphasen freigegeben Und/oder per Schalter komplett freigegeben oder gesperrt werden.

Achtung:

Alle möglichen Trimmungen können **hier** ausgewählt und in die Berechnung mit einbezogen. In den Mischer müssen Trimmungen aber auch aktiviert sein damit sie dann in die Kanäle „durchgereicht“ werden und wirken. Also in den Mixern jeweils mit „EIN“ aktivieren.



Beispiel: Dualrate und Exponentialanteil in 3 Stufen umschalten

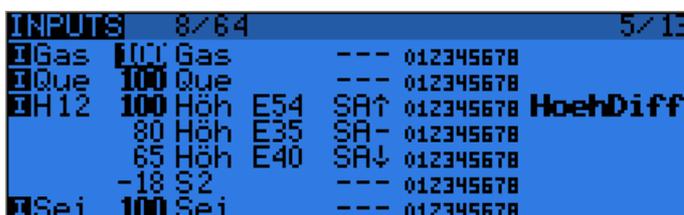
Damit kann man die linearen Knüppelwege abschwächen, die Knüppelkurve (meist eine Exponential-Funktion) um die Mittellage unempfindlicher machen und somit das Modell sanfter steuern. Gute Startwerte für Expo liegen bei ca. 35%.

Für jede Einstellung/Stick/Knüppel kann es mehrere Zeilen mit Parametern geben, die **per Schalter** aktiviert werden und dann die anderen Zeilen deaktiviert.

Aber: Pro Input kann aber immer nur eine Zeile aktiv sein!

Wenn man mehrere Zeilen mit verschiedenen Werten hat, dann den Zeilen immer einen Schalter zuordnen! **Zeilen ohne Schalter werden sonst nie aktiv!**

Sind mehrere Zeilen per Schalter aktiv hat immer die erste aktive Zeile Priorität und deaktiviert die nachfolgenden Zeilen!

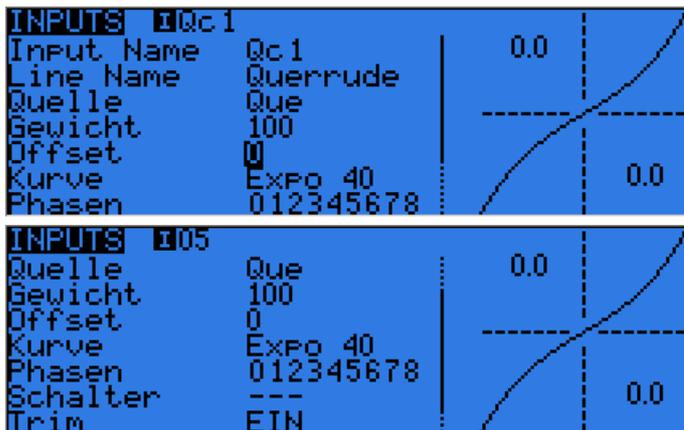


Am Beispiel Expo und Dualrate in 3 Stufen umschalten:

IH12 der Weg des Höhenruderstick Höh wird mit Schalter **SA** in 3 Stufen von 100%, 80%, 65% umgeschaltet und hat dabei einen Expoanteil von E54%, E35%, E40%

SA↑ ist aktiv da Fett dargestellt

Inputs Untermenü und Detailansicht:



Dieser Screen ist in 2 Hälften aufgeteilt:

- Links die Parameter zu der Zeile in der man gerade ist (vom Aufrufmenü)
- Rechts die Grafik entsprechend der **tatsächlichen Schalterstellung** mit den Koordinaten (Y, X) und der Knüppelstellung als kleines Kreuz

Der Inputname in den Input-Überschrift und auch so in den Mischern
Der Linename ist ein Infoname, eine Ergänzung im Inputmenü.

Eingabwerte:

1. **Quelle:** Das Eingangssignal das verarbeitet werden soll
2. **Gewicht:** das ist der Multiplikator mit der die Quelle verrechnet wird. Das kann aber auch durch eine globale Variable ersetzt werden, per [**Enter Long**] umschalten
3. **Offset:** Verschiebung nach oben und unten, entlang der Y-Achse. Das kann aber auch durch eine globale Variable ersetzt werden, per [**Enter Long**] umschalten
4. **Kurven:** Auswahl von Expofunktion, Differenzierungen, vordef. Funktionen, freie Kurven
5. **Differenzierungen,** sind einseitige Geradestücke deren obere oder untere Teile eine andere Steigung erhalten können.

6. **Kurven:** Eine frei definierbare Kurve **KV1..KV32**.

Wenn man hier eine Kurve **KV1-KV32** auswählt kommt man mit [**ENTER**] gleich in die entsprechende Kurvendarstellung von Screen (8/11) und kann sie sofort anpassen.

7. vordefinierte Funktionen mit Bedingungen

x > 0 wenn der x-Wert positiv wird er übernommen, sonst $x=0$

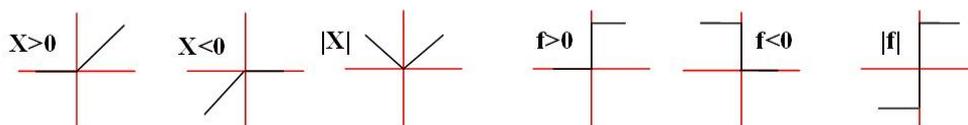
x < 0 wenn der x-Wert negativ wird er übernommen, sonst $x=0$

|x| der absolute Wert wird übernommen, d.h. immer ein positiver Wert erzeugt

f > 0 wenn der x-Wert positiv ist wird **fix** +100% übernommen, sonst $x=0$

f < 0 wenn der x-Wert negativ ist wird **fix** +100% übernommen, sonst $x=0$

|f| wenn der x-Wert negativ dann -100%, wenn der x-Wert positiv dann +100%



8. **Phase:** die Flugphasen **012345678** für die diese Zeile gültig sein soll.
Inverse Darstellung ist dann aktiv z.B. **01**2345678 nur in Flugphase 0 und 1 aktiv
(Vorgabe ist **012345678** also alle aktiv, das ist viel praktischer)
9. **Switch:** das ist der Schalter mit der die Zeile aktiviert oder deaktiviert wird.
Alle Arten von Schaltern sind möglich 1- 2- 3-stufige Zuweisungen sind möglich.
Hier können auch alle logischen Schalter **L1- L32** und inverse Schalter „!**!**“ stehen.
10. **Trim:** Es kann jeder einzelne Trimmwert (**Ail, Rud, Thr, Rud**) frei zugeordnet und übergeben werden, es gibt keine feste Zuordnung.
Trimmungen muss man von den Inputs zu den Mischern „durchreichen“, also im Mischer Trimmung „**EIN**“, damit sie am Kanal wirken können.

Hier hilft nur etwas ausprobieren, dann werden die vielen Möglichkeiten schnell klar.

Achtung aufpassen!

Linke Hälfte: Diese Werte in den Zeilen werden gerade editiert.

Rechte Hälfte: Die Grafik zeigt das Ergebnis an, aber in Abhängigkeit der **tatsächlichen Schalterstellung!**
Also hier mal den Schalter hin und her schalten,
damit man die Gesamt-Wirkung sieht!

Beispiel: Telemetriewerte anpassen, Geschwindigkeit normieren

Den Telemetriewert der Geschwindigkeit erhalten wir über einen Sensor (egal mal wie) und er liefert einen Bereich von z.B. 0 bis 350km/h.

Wenn wir diesen Wert weiter verwenden wollen, um damit etwas zu verrechnen, z.B. geschwindigkeitsabhängige Höhenrunderkorrektur, so muss man den Wert normieren, d.h. auf einen Bereich von 0 bis 100% umrechnen. Das muss man hier in der Signalvorverarbeitung machen.

Beispiel: Dualrate/Expo mit 3-Stufenschalter

Vorab: Ein Schalter kann als **Mischer-Quelle** und/oder als **Mischer-Schalter** eingesetzt werden. Das gilt auch sinngemäß für die Inputs-Verarbeitung, **Inputs-Quelle Inputs-Schalter**
 Als Mischer-Quelle liefert ein Schalter: 2-Stufig -100% und +100% 3-Stufig -100% 0% +100%
 Als Mischer-Schalter aktiviert oder deaktiviert er die Mischerzeile / Inputszeile

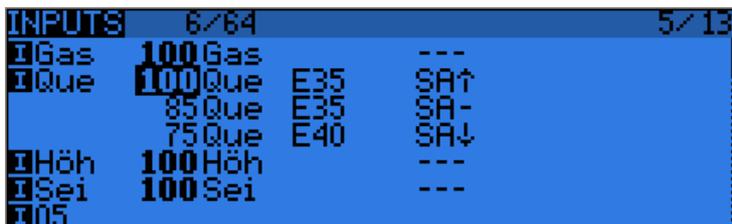
Jeder 3-Stufen Schalter kann auch als 2-Stufen-Schalter umprogrammiert/verwendet werden.
SA↑ SA- SA↓ in dem mit einer **Nicht-Funktion „!“** die anderen 2 Stufen ausgeblendet werden.
SA↑ !SA↑ oder **SA↓ !SA↓** oder **SA- !SA-**
 Er steht **„!“ nicht** in dieser Stellung, dann steht er eben in eine der 2 anderen Stellungen.

(Wem die vielen 3-Stufenschalter zu viele sind kann sie einfach durch 2-Stufenschalter ersetzen.)

Anwendung: Umschaltung von Daulrate und Expowerte in 3 Stufen.
 Das wird **normal in den Inputs gemacht**, kann aber genauso in den Mischer erfolgen.
 In den Inputs kann immer nur 1 Zeile aktiv sein, deshalb Zeilen umschalten per Schalter.

Dualrate und Expo im Querruder soll mit **SA** umgeschaltet werden.
 Wir brauchen 3 Zeilen in den Inputs des Querruders

Dualrate = Gewichtung: 100% 85% 75% **Expo:** 35% 35% 40% mit **Schalter:** SA↑ SA- SA↓



„Daulrate“ ist nur eine Umschaltung der Gewichtungen der Quelle
 Damit werden die Wege umgeschaltet



Links:
 Werden die Werte eingegeben



Rechts:
 Die angezeigte Grafik entspricht der tatsächlichen Schalterstellung!
 Also mal den Schalter betätigen!

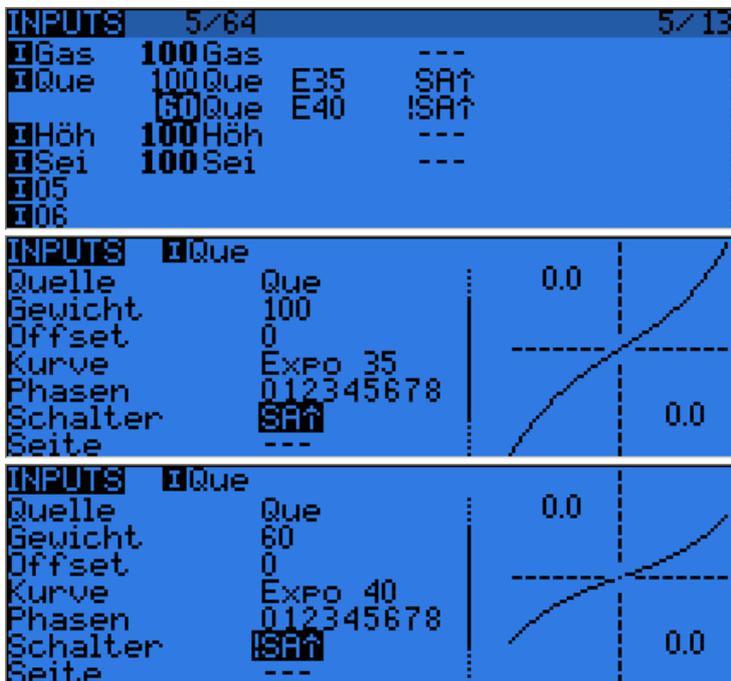


Beispiel: Dualrate/Expo mit 2-Stufenschalter

Anwendung: Umschaltung von Dualrate und Expowerte in 3 Stufen.

Dualrate und Expo 2 Stufen-Schalter **SA↑ !SA↑** Dazu brauchen wir nur 2 Zeilen

Dualrate = Gewichtung: 100%, 60 Expo: 35% 40% mit **Schalter: SA↑ !SA↑**



Festwerte / Variable Werte

In den Quellen und den Expowerten haben wir hier einfach feste Werte eingetragen. Genauso könnten wir hier variable Werte eintragen und diese dann im Flug anpassen. Dazu gibt es Globale Variablen GVx
Bei der Eingabe, mit **[Enter Long]** erfolgt die Umschaltung von Festwert auf GVx

Begriffe:

„Dualrate“ ursprünglich hatten Sender nur **eine einfache** Umschaltung von Weg und Expowert
Mit einem 3-Stufen Schalter kann man aber zwischen 3 Werte umschalten.

Schalterbezeichnungen: Taranis, TH9x, 9XR

Die Schalterbezeichnungen sind so wie sie am Sender angebracht sind. SA..... GEA....TH...
In OpenTx sind sie aber nicht festen Funktionen zugeordnet, sondern frei verwendbar.
Schalter sind immer in Großbuchstaben (SA, SB, GEA, THR)
Geber haben Kleinbuchstaben, Gas, Que, Höe, Rud, das wird beim Programmieren oft verwechselt!

Mischerfunktionen (6/13)

Das ist das wichtigste Menü, alles läuft über Mischer!

Haben alle anderen Sender dutzende fester Funktionen für alles Mögliche und nur wenige freie Mischer, gibt es bei openTx keine dieser vorgefertigten Funktionen sondern 64 freie Mischer. Das ist zuerst mal ungewohnt, bietet aber maximale Flexibilität. Es gibt keine Beschränkungen oder feste Voreinstellungen wie bei anderen Fernsteuerungen. Die Kanalzuordnungen sind total frei und das Programmieren ist immer gleich. Für einfache Grundmodelle gibt es fertige Templates.

In den Mixern wird alles zusammengeführt, verrechnet und an die Ausgangskanäle **CH1-CH32** weitergeleitet. Nichtbenötigte Mischerkanäle kann man als virtuelle Kanäle für Berechnungen zur Vorverarbeitung nutzen und sie dann in anderen Mixern oder für globale Variablen zur Weiterverarbeitung wieder aufrufen. Das vereinfacht vieles und spart Schreibaufwand.

Mischer Hauptbildschirm und Übersichten

```
MISCHER 4/64 6/13
CH1 Gas 100 SD↑S 012345678 Buttflay
CH2 Que 100 012345678
CH3 Höh 100 012345678
CH4 Sei 100 012345678
CH5
CH6
CH7
```

Mit **[Menü Long]** erhält man den Kanal-Monitor und kann dann die Mischer-Ergebnisse des angewählten Kanal ansehen.

```
MISCHER 5/64 6/13
CH1 S Zeile Editieren
+= S Neue Zeile davor
CH2 H Neue Zeile danach
CH3 H Zeile Kopieren
CH4 H Zeile Verschieben
CH5 H Zeile Löschen
CH6
```

Mit **[Enter Long]** ins Auswahlmenu

```
MISCHER Edit CH1
Mix Name  | Schalter ---
Quelle Sei | Warnung AUS
Gewicht 100 | Wirkung Addier
Offset 0 | Verz. Up 0.0
Trim FEI DRex  | Verz. Dn 0.0
Kurve Diff 0 | Langs.Up 0.0
Phasen 0000000000 | Langs.Dn 0.0
```

Bereichsdarstellung im Mischer
Gewichtung 50 und Offset 25
führt zu einem Weg von -25 bis +75

```
Gewicht 50 -25 75
Offset 25
```

Die erste Zeile "4/64" bedeutet, dass 4 von 64 Mixern benutzt werden,
Hier gibt es je nach Funktion bis zu 6 Spalten mit unterschiedlichen Bedeutungen

Bedeutung der Spalten in der Mischer Hauptanzeige 6/13:

- Kanalnummern (CH1..CH32) oder die Operatoren (+=/*/=/:=)
Bedeutet (Addiere, Multipliziere, Ersetze) die darüberstehende Zeile zum CH
- Signalquelle für den Mischer (wird **fett** dargestellt wenn die Quelle aktiv wird)
- Gewichtung, Anteil der übergeben wird. **-125% bis +125%**
- Die Kurve, oder die Bedingung damit die Quelle aktiv wird, oder Ruder-Differenzierung (30)
- Ein Schalter, der diese Mischer-Zeile aktiviert (SA↑)
- Verzögerung Delay(D), Langsam Slow(S) oder beides (*) damit die Zeile aktiv wird
- Der Name des Mixers, max. 8 Zeichen

Ähnlich wie bei DR/Expo kann man einem Kanal mehrere Zeilen (Mischer) zuordnen. Sie sind aber hier **alle aktiv** und werden durch die Operatoren gesteuert die durch Addiere +=, Multipliziere *=, Repalce := die Zeilen für diesen Ausgangskanal verrechnen.

Mischerzeilen kann man so wie immer auswählen, kopieren, verschieben oder löschen wie in allen anderen Menüs auch. So wie bei Modellauswahl, oder DR/Expo beschrieben. Das läuft immer gleich ab.

Zeilen einfügen und ins Untermenü (wie bei Dualrate/Expo)

Mit [+] [-] die Mischer-Zeile anwählen, dann kommt man mit [ENTER LONG] ein Auswahlmenü



Mit [Enter Long] ins Auswahlmenü

Mischer Detailansicht, Untermenü, Werte editieren



Bereichsdarstellung im Mischer
Gewichtung 50 und Offset 25
führt zu einem Weg von -25 bis +75



Mit den 2 Cursortasten [+] [-] kann man frei in alle Eingabespalten/Zeilen navigieren und dann mit [ENTER] die Editorfunktion auswählen, der Bereich blinkt dann.

Bedeutung der Optionen und möglich Werte:

Jeder Mischerzeile kann auch einen individuellen Namen erhalten z.B. QuerLi, QuerRe

Mischer Name mit max 8 Zeichen

1. **Quelle**: Die Quelle für den Mischer, d.h. wo kommen die Signale her:
 - a. Analogwerte, die 4 Knüppel und 3 Potis (**Sei, Hoe, Gas, Que, S1, S2, S3, LS, RS,**) (Analoggeber liefern als Source/Quelle -100%+100%)
 - b. Trimmungen – **TrmA, TrmE, TrmT, TrmR** auch die Trimmungen können als Eingangswerte verwendet werden. Frei verwendbar, unabhängig von Ihrer Zugehörigkeit zu den Knüppeln. Das wird oft verwendet für die Gas Trimmungen (Deutsch: **TrmS, TrmH, TrmG, TrmQ**) Seite, Höhe, Gas, Quer. Trimmungen liefern standardmäßig -25% bis + 25%, wer mehr braucht muss Extended Trims wählen!

openTx für Taranis Anleitung Deutsch

- c. **MAX**: liefert als Quelle einen Festen Wert von 100% .
MAX wird oft in Verbindung mit Schaltern verwendet.
Mit Weight (-100% bis +100%) kann man den Wert anpassen und auch invertieren.
 - d. **CYC1, CYC2, CYC3**: Das sind die 3 Heli-Taumelscheibenmischer wenn sie im Helimenü 3/10 aktiviert wurden. **CY1** liefert den Pitch-Wert, die beiden anderen die Rollwerte, aber bei **120X**-Mode liefert **CY1** den Roll-Wert
 - e. **TR1..TR16**: Sind die Eingangskanäle die über die DSC-Buchse (Trainer-Port) reinkommen. Damit kann man den Sender auch mit zusätzlichen Eingangskanälen erweitern (z.B. für FPV, Headtracking). Das hat nichts mit der Trainerfunktion zu tun, die ja nur die 4 Knüppel im Lehrer/Schülerbetrieb ersetzen!
Dazu muss der Sender auf Lehrer stehen. Dann wird automatisch erkannt welche und wieviele PPM Signal an der DSC Buchse ansteigt.
 - f. **CH1 .. CH32**: Das sind die Ausgänge von anderen Mixern die auch als Eingänge benutzt werden können. z.B. der fertige verrechnete Kanal 14 (egal wie aufwendig der entstanden ist) soll für Kanal 4 als Eingang verwendet werden. Damit kann ein sehr komplexes Verhalten programmiert werden.
 - g. Alle logischen, virtuellen, programmierbaren, Schalter **LS1- LS32** (siehe dazu im spez. Kapitel).
Alle 8 physischen Schalter **SA..SH**, als 2Pos und 3Pos –Schalter
Stellungen einzeln abfragen
2 Stufen- Schalter als Quelle liefern entweder -100% oder +100%
3 Stufen Schalter als Quelle liefern -100% 0% +100%
2. **Gewicht**: Gewichtungsfaktor (Multiplikator/Verstärkung) für die Verrechnung des Eingangs/Quelle. Werte von -500% bis 500% sind möglich. Als Gewichtung kann auch eine globale Variable verwendet werden (zum Verständnis, siehe Beispiele!)
 3. **Offset**: Ausgleich, dieser Offsetwert addiert seine Wert zum Eingangswert/Quelle Werte von -500% bis 500% sind möglich. Das entspricht einer Verschiebung. Als Offset kann auch eine globale Variable verwendet werden (zum Verständnis, siehe Beispiele!)

Trimmungen können auch in den Inputs 5/13 eingegeben werden

4. **Trim**: Mit dieser Option **EIN** werden die Trimmwerte der Knüppel übernommen und an den Mischer mit übergeben. Wenn **AUS** werden sie ignoriert.
Es kann jeder einzelne Trimmwert (**Ail, Rud, Thr, Rud**) frei zugeordnet und übergeben werden, es gibt keine feste Zuordnung.
Das ist recht praktisch wenn z.B. die Gas-Trimmmung für das Gas nicht gebraucht wird, kann man sie frei für etwas anders verwenden, oder für Überkreuz-Trimmmungen, oder für Geber die keine eigene Trimmung haben, z.B. die Potis

Kommt die Mischerquellen aus der Inputs-Vorverarbeitung z.B. als „**I**Gas“, müssen dort auch die Trimmungen auf „**EIN**“ sein, damit sie zum Mischer durchgereicht werden.

Kurz: **In den Inputs EIN und in den Mischer EIN**, damit sie am Kanal (Servo) wirken!

5. **Kurve:** Umschalten auf Diff, Expo, Func, Kurve
hier werden Kurven und Funktionen mit der Mischerquelle verrechnet.

Die spezielle Kurve Diff:

Diff: das ist die Querruder-Differenzierung. Der Wert rechts gibt die % - Differenzierung an. Anstatt einem festen Wert kann auch eine globale Variable verwendet werden. Wenn Diff = 0% wird keine Ruder-Differenzierung aktiviert. Diff = 100% volle Ruder-Differenzierung, d. H. das Ruder geht nicht mehr nach unten. Diff=60% Querruder geht nur noch zu 40% nach unten. **Diff als positive Werte eingeben!** Wird immer vom **negativen** Ruder-Wert berechnet ($-100\% + 60\% = -40\%$)

Die Querruder-Differenzierungs-Funktion ist ganz einfach:

- o Kanal für das rechte Querruder wird mit Weight auf z.B. +100% und 60% Diff
- o Kanal für das linke Querruder wird mit Weight -100% und 60% Diff gesetzt.
- o Das bedeutet, wenn das rechte Querruder auf 100% nach oben geht, wird das linke Querruder nur auf 40% nach unten gehen und umgekehrt.

Damit spart man 2 Mischer. Es gibt auch noch einen anderen Weg 2 Mischer zu sparen, mit Kurven, das ist aber etwas aufwändiger.

Achtung: Ruderlaufrichtungen werden im Servo-Limitmenü eingestellt. Auf keinen Fall die Mischerberechnungen "verbiegen" damit das Ruder richtig läuft!

Expo Expowerte für alle möglichen Quellen. Falls nicht angewählt, kommt der Eingang direkt vom Stick und nicht über die Dualrate/Expo Funktion.

Funktion: hier werden mit Bedingungen, feste Werte erzeugt..

$x > 0$: der Wert der Quelle wird nur dann verwendet wenn sie positive Werte liefert, ansonsten wird Null "0" verwendet.

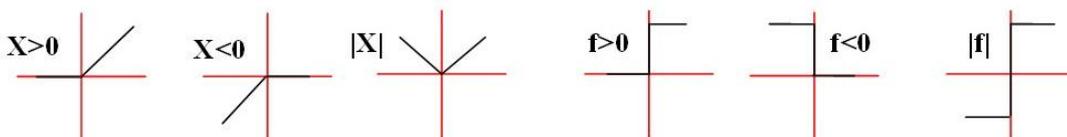
$x < 0$: Wie oben nur bei negative Werten .

$|x|$: Der Wert den die Quelle liefert ist immer Betrags absolut d.h. nur positiv.

$f > 0$: Wenn die Quelle positiv ist, dann wird Weight "+Weight", ansonsten Null "0".

$f < 0$: Wenn die Quelle negativ ist, dann wird Weight "-Weight", ansonsten Null "0"

$|f|$: Je nach Vorzeichen der Quelle wird auch "+Weight" oder "-Weight"



Kurve: Kurven KV1 bis KV32 auswählen die im Menü KURVEN 8/12 erstellt wurden. Wenn man [ENTER] drückt kommt man sofort/direkt in die Kurven rein zum editieren.

7. **Phasen:** hier werden die Flugphasen FP0-FP8 definiert die aktiv sein sollen. Wird hier nichts geändert, sind für diese Mischerzeile alle Flugphasen automatisch aktiv. Man kann damit in einem Mischer mehrere Flugphasen zu-und abschalten. Aktiv ist was invers in der Zeichenkette 012345678 dargestellt wird (012345678) Das passiert auf bekannter Weise. Mit den Cursors [+] / [-] ein Zeichen anwählen, mit [ENTER] eine Flugphase zu-und wegschalten.

8. **Switch**: hier wird der Schalter in seiner Stellung \downarrow - \uparrow festgelegt der die Mischerzeile aktiv schaltet. Wird kein Schalter ausgewählt ist die Mischerzeile immer aktiv und wird von Source gesteuert.

9. **Warning**: hier kann man 1- bis 3 kurze Piepser/Alarmtöne wählen wenn ein Mischer aktiviert wird (aber nur wenn er mit einem Schalter aktiviert wird). Die Töne bleiben so lange ein bis diese Mischer-Zeile per Schalter wieder ausgeschaltet wird.

10. **Multipx**: hier wird ab der 2. Mischerzeile im Kanal definiert wie eine zusätzliche Mischer- Zeile zum Kanal verrechnet wird
 - Add +=** Der Wert dieser Mischer-Zeile wird zur vorherigen dazu addiert und dem Kanal zugeordnet.
 - Multiply *=** Der Wert dieser Mischer-Zeile mit der vorherigen multipliziert und dem Kanal zugeordnet.
 - Replace :=** Diese Zeile ersetzt die **vorherigen** Mischer-Zeile, wenn sie mit einem Schalter aktiviert (**ON**) wird. Dann wird diese Zeile dem Kanal zugordnet. Zeilen **danach bleiben** weiterhin erhalten/ aktiv! Solange der Schalter (**OFF**) ist wird diese Zeile ignoriert.

11. **Delay Down / Up**: Verzögerungszeit, Einschaltverzögerung, Ausschaltverzögerung, bis der Kanal reagiert d.h. eine Bewegung beginnt. Wird normal mit einem Schalter aktiviert. Wenn der Schalter "**ON**" oder "**OFF**" geht erfolgt die Reaktion in der Mischerzeile erst wenn die Zeit (max. 15s) abgelaufen ist. Also erst dann beginnt die Bearbeitung der Mischerzeile. (z.B. für Fahrwerks-Klappen)

12. **Slow Down / Up**: Verlangsamung der Wertänderung eines Kanal. Wenn der Wert nicht Null "**0**" gibt dieser Wert die Zeit (max. 15s) in Sekunden an für eine Wertänderung die von -100% auf +100% vergeht. (für langsame Übergänge/langsame Servobewegungen)

Zeiten (Delay/Slow) in einer Mischerzeile werden nur aktiviert, wenn sich die Mischer**quelle** ändert (z.B. per Schalter umschalten). Nicht aber, wenn nur eine andere Mischerzeile bei gleicher Quelle aktiv wird. Ansonsten mit einem freien Vorverarbeitungskanal arbeiten und den dann im Mischerkanal zur Umschaltung verwenden. Der virtuelle Kanal schaltet um wenn er aktiv wird, der reale Kanal läuft dann langsam, da sich die Quelle geändert hat. Siehe Beispiele Teil C

Alternativ mit einem logischen Schalter umschalten, das ist auch ein „anderer“ Schalter. Dazu gibt es hinten ein Beispiel

Aber: Flugphasen schalten hingegen immer komplette Mischerzeilen um.
Mit den FadeIn- FadeOut- Zeiten in den Flugphasen 4/13 kann man den Übergang von einer FlugPhase in die andere das schön sanft und landsam einleiten.

Zu beachten Die tatsächliche Geschwindigkeit der Servobewegung hängt auch von den Kurven ab. Eine flache Kurve führt zu einer langsameren Bewegung als eine steile Kurve.

Grundprinzip der Mischerberechnungen: gilt ab OpenTx2.0!

Die Berechnungen in den Mischerzeilen erfolgen stark vereinfacht so:

CHx= [(Source * Gewichtung) + Offset] → Trim → Kurve.....

oder wenn es mehrere Mischerzeilen gibt:

CHx= (Ergebnis der ersten Mischer Zeile) +=, *=, := (Ergebnis der zweiten Mischer Zeile)
(addiere +=, multipliziere *=, ersetze :=)

Beispiel: [(Source * Weightt) +Offset] → Trim → Kurve → Chx

Source z.B. ein Poti liefert -100% bis + 100% und soll nur noch 0% bis 100% liefern.

Die Spanne soll anstatt 200% nur noch 100%

Dafür müssen wir den Werte für Weight und Offset ausrechnen.

Weight = Spanne / 200 somit 100/200=50%

Offset = auf die Mitte des neuen Bereichs, dient als Nullpunktverschiebung, hier 50%

Die Min und Max Werte sind jetzt:

Min= $((-100% * 50%) + 50%) = 0%$ Max= $((+100% * 50%) + 50%) = 100%$

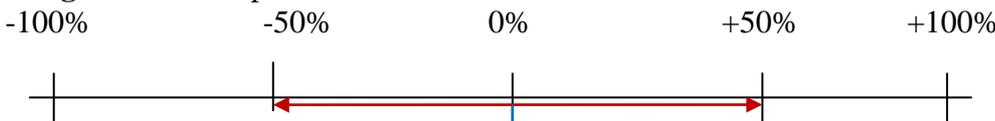
Damit liefert das Poti an nur noch positive Werte von 0 bis 100%

Grafische Darstellung der Bereichsanpassung und Bereichsumrechnung in 2 Schritten

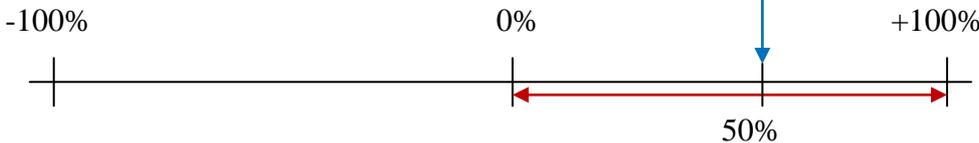
Ausgangsbereich: -100% bis +100% = 200



Weight: Aus der Spanne von 100% berechnen 100/200 = 50%



Offset: verschieben auf die Mitte des neuen Bereichs



Das ergibt das Ziel: Spanne mit 100% im Bereich von 0% bis 100%

Wir merken uns:

**Weight mit der Spanne ausrechnen,
Offset die Mitte des neuen Bereichs verwenden!**

Darstellung des Bereichs im Mischer

Gewicht	50	-25	75
Offset	25		

Weight und Offset können auch mit GVARs variabel eingestellt werden.

Beispiele der Mischer Berechnungen ab OpenTx2.0

Die Mischerberechnung hat sich in openTx2.0 geändert, dadurch ist eine einfachere und bessere Anpassung möglich. Die gleiche Berechnung gilt auch für die Inputs.

Berechnungsformel für den Mischerwert:

Bis opentxV2.00 galt die alte Mischerberechnung:

$[(\text{Quelle} + \text{Offset}) * \text{Gewichtung}] = \text{Mischerwert} + \text{Trim} \rightarrow (\text{DR/Expo/Kurve}) \rightarrow \text{Kanal}$

Da war der Trick mit einem negativer Offset nötig: **Minus * Minus = Plus**

um bei **negativer** Gewichtung den Offsetwert selbst ins Positive zu bringen.

- Offset * - Gewichtung = + Ausgang

Ab opentxV2.00 gibt es die neue Mischerberechnung:

$[(\text{Quelle} * \text{Gewichtung}) + \text{Offset}] = \text{Mischerwert} + \text{Trim} \rightarrow (\text{DR/Expo/Kurve}) \rightarrow \text{Kanal}$

den Trick braucht man nicht mehr, Offset selbst gleich Positiv reinstellen, wenn er nach oben soll

Beispiel: Wertebereich verschieben von **-100% bis +100%** nach **0% bis +100%**

Source liefert -100% 0% +100%, Gewichtung ist der Multiplikator, Offset ist die Verschiebung

Bisher wurde so gerechnet:

$[(-100\% -100\%) * -50] = \mathbf{0\%} + \text{Trim} \rightarrow (\text{DR/Expo/Kurve}) \rightarrow \text{Kanal}$

$[(0\% -100\%) * -50] = \mathbf{+50\%} + \text{Trim} \rightarrow (\text{DR/Expo/Kurve}) \rightarrow \text{Kanal}$

$[(+100\% -100\%) * -50] = \mathbf{+100\%} + \text{Trim} \rightarrow (\text{DR/Expo/Kurve}) \rightarrow \text{Kanal}$

Ab openTx2.0 wird so gerechnet:

$[(-100\% * 0,5) + 50\%] = \mathbf{+0\%} + \text{Trim} \rightarrow (\text{DR/Expo/Kurve}) \rightarrow \text{Kanal}$

$[(0\% * 0,5) + 50\%] = \mathbf{+50\%} + \text{Trim} \rightarrow (\text{DR/Expo/Kurve}) \rightarrow \text{Kanal}$

$[(+100\% * 0,5) + 50\%] = \mathbf{+100\%} + \text{Trim} \rightarrow (\text{DR/Expo/Kurve}) \rightarrow \text{Kanal}$

Es kann jeder Wertebereich von -100% bis +100% = 200% beliebig angepasst werden:

z.B. +0% bis +100%, Spanne ist 100%, Mitte ist bei+50%

Berechnung: Gewichtung ist $100/200=0,5=50\%$ Offset =+50%

z.B. +0% bis +80%, Spanne ist 80%, Mitte ist bei+40%,

Berechnung: Gewichtung ist $80/200=0,4=40\%$, Offset =+40%

z.B. +35% bis +85%, Spanne ist 50%, Mitte ist bei +60%

Berechnung: Gewichtung ist $50/200=25\%$, Offset ist +60%

z.B. +25% bis +50%, Spanne ist 25%, Mitte ist bei+37,5%,

Berechnung: Gewichtung ist $25/200=0,125=12,5\%$ Offset =+37,5%

z.B. -50% bis +100%, Spanne ist 150%, Mitte ist bei +25%,

Berechnung: Gewichtung ist $150/200=0,75=75\%$ Offset = +25%

z.B. -50% bis -10%, Spanne ist 40%, Mitte ist bei -30%,

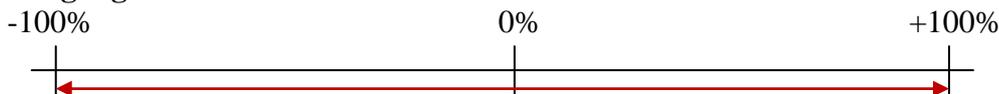
Berechnung: Gewichtung ist $40/200=0,2=20\%$ Offset = - 30%

z.B. -85% bis -35%, Spanne ist 50%, Mitte ist bei -60%

Berechnung: Gewichtung ist $50/200=25\%$, Offset = - 60%

Grafische Darstellung der Bereichsanpassung und Bereichsumrechnung in 2 Schritten

Ausgangsbereich: -100% bis +100% = 200



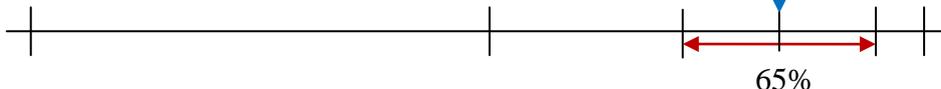
Berechnung: Gewichtung Aus der Spanne von 60% berechnen $60/200=33,3\%$

-100% -30% 0% +30% +100%



Offset: verschieben auf die Mitte des neuen Bereichs

-100% 0% +35% +95% +100%



Das ergibt das Ziel: Spanne = 60% im Bereich von 35% bis 95%

Bitte mal im Kanalmonitor ansehen!

Wir merken uns:

- Gewichtung mit der Spanne ausrechnen
- Offset die Mitte des neuen Bereichs

Darstellung des Bereichs im Mischer

Gewicht	50	-25	75
Offset	25		

Gewichtung und Offset können auch mit Globalen Variablen GV variabel eingestellt werden.

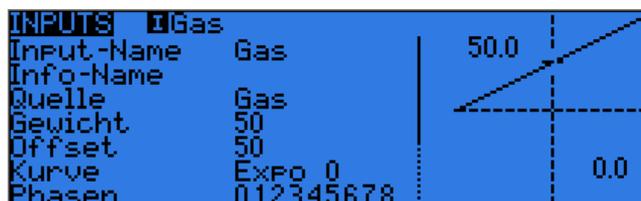
Tip:

Wer sich das mal ansehen will, kann in den Inputs mit Gewicht und Offset spielen und sich das Ergebnis dann im Kanalmonitor ansehen.

Gewicht:100 Offset: 0



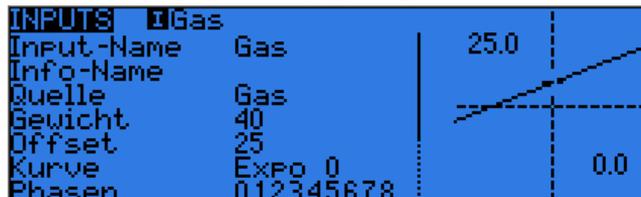
Gewicht: 50 Offset:50



Gewicht: 50 Offset: 25



Gewicht: 40 Offset: 25



Mischerverarbeitung im Detail: EVA-Prinzip Eingabe, Verarbeitung, Ausgabe

Es müssen immer 3 Dinge geklärt werden:

1. Wo kommt das Signal her, was ist die Signalquelle
2. Was soll das Signal tun, wie muss es dazu verrechnet werden
3. Wo soll das Signal dann wirken, auf welchen Kanal und wie

Mit Schalter oder Flugphasen wird der Mischer grundsätzlich freigegeben oder gesperrt.

Dann läuft erst mal die Verzögerung an und wartet bis sie abgelaufen ist.

Jetzt werden Werte (Sticks) abgefragt, gehen eventl. über DR/Expo und stehen als Quelle bereit.

Erst jetzt läuft die Mischer-Berechnung mit Source, Offset, Weight und Trimm an

Das Zwischenergebnis durchläuft eventl. noch eine ausgewählte Kurve, geht dann durch die Langsamfunktion und kommt an Ausgangskanal CHx an.

Mischer Freigabe: → Schalter oder Flugphasen → eventl. Verzögerung starten

Eingabe: Stick → eventl. DR/Expo → Source

Verarbeitung: → [(Source * Weight) + Offset] + Trim → ausgew. Kurve →

Ausgabe: → eventl. Langsam → CHx → **Servo Limits** (7/12) → Servo

Verknüpfung mit weiteren Mischerzeilen: += oder *= oder := → CHx

```
MIXER 7/64 6/13
CH4 Rud 100 012345678
CH5
CH6 MAX 100 SA↑ 012345678
:= MAX 0 SA- 012345678
:= MAX -100 SA↓ 012345678
CH7

```

Hier Kanal 6 mit 3 Stufen Schalter **SA** und **MAX** als Quelle und **:=** für Replace, ersetze

Hinweis:

In jeder Mischerzeile kann die Langsam-Funktion enthalten sein, aber es kann pro Ausgangskanal CHx immer **nur eine** Mischerzeile mit langsam aktiv geschaltet sein!

Werden Funktionen mehrfach benötigt wird einfach ein freier Mischer als Vorverarbeitungs-Mischer verwendet, der dann selber wieder auf andere Ausgangsmischer/Kanäle wirken kann.

Das spart Programmieraufwand da dieser Vorverarbeitungs-Hilfsmischer dann nur einmal benötigt wird.

Grundverständnis der Mischerberechnungen

Am Beispiel Querruder wird das klar: Ich gebe Knüppel Querruder rechts.

Dann:

kommt von Knüppel ein positiver Wert als Quelle an den Mischer für das rechte Querruder z.B. CH2 und das Ruder soll nach oben gehen (das ist die positive Richtung)

Gleichzeitig

kommt von Knüppel ein positiver Wert als Quelle an den Mischer für das linke Querruder z.B. CH5 und das Ruder soll nach unten gehen (das ist die negative Richtung)

Die Mischerberechnung muss jetzt so gehen:

CH2: Knüppel Querruder mit **positiver** Gewichtung $CH2 = \text{Quer1} * (+100\%)$

und

CH5: Knüppel Querruder mit **negativer** Gewichtung $CH5 = \text{Quer2} * (-100\%)$

Damit sind die beiden Mischerberechnungen mathematisch richtig!

Die tatsächlichen Laufrichtungen der Ruder werden erst in den Servoeinstellungen gemacht!

Wenn ich jetzt für beide Querruder eine Landeklappenfunktion zumische setze ich zwei zusätzliche Mischer ein und gebe bei beiden einfach per Schalter z.B. -25% als Gewichtung ein, dann wird wieder math. richtig gerechnet und beide Querruder gehen gleichmäßig nach unten!

Das sieht dann so aus:

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail (Quer1)
	(-25%) MAX Schalter (SA↓) (Lande1)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD)
CH05	(-100%) Ail (Quer2)
	(-25%) MAX Schalter (SA↓) (Lande2)
CH06	
CH07	

Wenn ich das beachte werde ich nie ungleiche Ruderausschläge erhalten!

Soweit klar?

Bei fast allen anderen Sender gebe ich für beide Querruder positive Werte ein.

(Das liegt daran, dass Querruder am Ruderhorn des Servos normal gespiegelt angelenkt werden und damit dort die Invertierung erfolgt). Dann brauche ich jetzt aber eine Spezialfunktion Landeklappe, dort gebe ich für beide Kanäle -25% ein und dann wird ein Kanal intern verdeckt wieder invertiert, damit richtig gerechnet wird.

Beispiel: Schalter als Mischer-Quelle mit 2 -Stufen und 3-Stufen Schaltern

Knüppel und Potis liefern als **Quelle variable Werte** aus von -100% bis +100%
 Schalter als **Quelle** liefern **automatisch** Festwerte -100% 0% +100%

Damit wird es verblüffend einfach um z.B. Servo links, mitte, rechts zu steuern

Ein 2-fach Schalter z.B. **SG (SG↑ SG↓)** als **Quelle** (bei Mischer oder Inputs)
 liefert automatisch 2 feste Werte -100% +100% d.h. Servo links, rechts

MISCHER HINZ. CH6			
Mix-Name		Schalter	---
Quelle	SG	Warnung	AUS
Gewicht	100	Wirkung	Addiere
Offset	0	Verz. Up	0.0
Trim	<input checked="" type="checkbox"/>	Verz. Dn	0.0
Kurve	Diff 0	Langs.Up	0.0
Phasen	012345678	Langs.Dn	0.0

Anpassung per Gewichtung, Freigabe mit einem Schalter möglich

Ein 3-fach Schalter z. B. **SC (SC↑ SC- SC↓)** als **Quelle** (bei Mischer oder Inputs)
 liefert automatisch 3 feste Werte -100% +0% +100% d.h. Servo links, mitte, rechts

MISCHER HINZ. CH6			
Mix-Name		Schalter	---
Quelle	SC	Warnung	AUS
Gewicht	100	Wirkung	Addiere
Offset	0	Verz. Up	0.0
Trim	<input checked="" type="checkbox"/>	Verz. Dn	0.0
Kurve	Diff 0	Langs.Up	0.0
Phasen	012345678	Langs.Dn	0.0

Anpassung per Gewichtung, Freigabe mit einem Schalter möglich

Der Festwert **MAX** als **Quelle** (bei Mischer oder Inputs)

liefert als festen Wert immer +100% Servo rechts,

Max kann mit einem Schalter aktiviert werden und per Gewichtung (-100 bis +100)
 angepasst, umgeschaltet, weggeschaltet werden

MISCHER HINZ. CH6			
Mix-Name		Schalter	SA↑
Quelle	MAX	Warnung	AUS
Gewicht	100	Wirkung	Addiere
Offset	0	Verz. Up	0.0
Trim	<input checked="" type="checkbox"/>	Verz. Dn	0.0
Kurve	Diff 0	Langs.Up	0.0
Phasen	012345678	Langs.Dn	0.0

MISCHER HINZ. CH6			
Mix-Name		Schalter	SA-
Quelle	MAX	Warnung	AUS
Gewicht	85	Wirkung	Addiere
Offset	0	Verz. Up	0.0
Trim	<input checked="" type="checkbox"/>	Verz. Dn	0.0
Kurve	Diff 0	Langs.Up	0.0
Phasen	012345678	Langs.Dn	0.0

Ein Schalter als Mischerschalter / Inputsschalter aktiviert/ deaktiviert die Zeile!

Somit kann man auch kombinieren. **SA** als Mischerquelle die -100% 0% +100%
 und **SA↑ SA- SA↓** als Mischerschalter die die Zeilen freigibt/sperrt.

Beispiel: Landeklappen mit 3 Stufen-Schalter fahren

Variante 1: 1 Servo, beide Landeklappen sind am Kanal 6 angeschlossen.

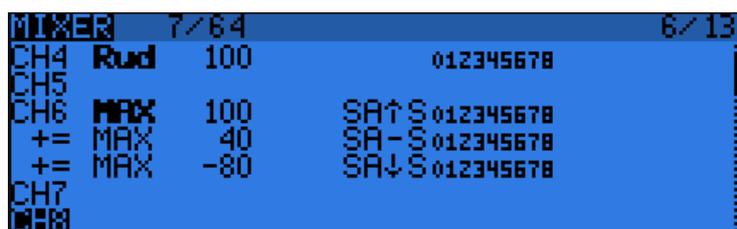
Bei einem 3 Stufenschalter ist immer nur eine Stufe aktiv. Beispiel: **SA↑ SA— SA↓**

Damit schaltet man 3 Mischer mit 3 komplett getrennten Einstellungen um.

Die Quelle ist **MAX** (MAX = Festwert mit 100%)

Mit einer Langsamfunktion **S=Slow** fahren die Klappen mit unterschiedlichen Geschwindigkeiten aus und wieder ein. **u=Up** geht in Richtung +100% **d=Down** geht in Richtung -100%

Hinweis: Hier im Mischer mit Addiere += arbeiten, da immer nur eine Stufe des Schalters aktiv ist, kann auch immer nur eine Mischerzeile aktiv sein. Kein **Replace :=** verwenden!



Mit Companion9x programmiert sieht das dann im Detail so aus:

CH05	
CH06	(+100%)MAX Schalter(SA↑)Langsam/u0.5:d1.5)
	(+40%)MAX Schalter(SA-)Langsam/u0.5:d1.5)
	(-80%)MAX Schalter(SA↓)Langsam/u0.5:d1.5)
CH07	

3 Stufen +100% , +40% -80%
MAX liefert Festwerte 100%
 Mit **SA** als 3-Stufen-Schalter
 0,5s für up 1,5s für down

Variante 2: 2 Servos, Landeklappe links Kanal 6, Landeklappe rechts Kanal 7

Wie geht das jetzt? Die Lösung ist ganz einfach:

Kanal 7 erhält als Quelle den Kanal 6, da dort ja schon alles fertig berechnet ist. Das wars, fertig!

Das sieht dann in Companion9x so aus:

CH05	
CH06	(+100%)MAX Schalter(SA↑)Langsam/u0.5:d1.5)
	(+40%)MAX Schalter(SA-)Langsam/u0.5:d1.5)
	(-80%)MAX Schalter(SA↓)Langsam/u0.5:d1.5)
CH07	(+100%)CH06
CH08	

Kanal 7: Quelle ist Kanal 6

Jetzt läuft aber mindestens ein Servos noch „falsch“ rum, was tun?

Die tatsächliche **Laufrichtungen** und auch die Wegbegrenzungen **werden**

nicht in den Mischern sondern im Servomenu eingestellt! Nur dort werden die errechneten Mischerwerte den tatsächlichen physikalischen Verhältnissen so angepasst, dass das Servo „richtig“ rum läuft, egal wie die Einbaulage und das Ruderhorn angelenkt wird.

Nochmal:

Mischer mit Ihren Verknüpfungen berechnen Werte so, dass positive Werte ein Ruder nach oben bzw. nach rechts bewegen. Nicht schon in den Mischern die Servo-Drehrichtungen anpassen und „verbiegen“ dass es passt. **Das muss man sich merken!**

Beispiel: Motor Sicherheits-Schalter (Throttle Cut) in 4 Varianten

Gerade bei Modellen mit Eelektromotor ist ein Gassicherhreitsschalter sehr wichtig. Damit wird verhindert, dass der Motor sofort anläuft wenn man an den Gasknüppel kommt.

Hier mal 4 Varianten, 2 einfache, 2 bessere

1. Mit einer zusätzlichen Mischerzeile, die den Gasknüppel überschreibt solange sie aktiv ist.



Im Mischer als Quelle **MAX** (Max hat einen Festwert von +100%) und eine Gewichtung (Weight/Anteil) von **-100%** eingeben.

Dann brauchen wir einen Schalter, hier **SE**, der diese Zeile aktiviert oder freigibt, Dann soll diese Mischerzeile die vorherige ersetzen, also **Ersetze / Replace (:=)** eingeben.

Das liest sich nun für Kanal 1 so:

Normal bekommt Kanal 1 sein Analogsignal (-100% 0% +100%) vom Gasknüppel (**Gas**) mit einem Anteil von 100%. Das Servo kann von Links über Mitte nach Rechts laufen.

Wenn der Schalter **SE** betätigt wird greift Zeile 2, Zeile 1 wird ungültig da **Ersetze :=** Kanal 1 erhält jetzt von Max den Wert 100, mal Gewichtung = -100% somit also $(100 * -100\%) = -100\%$ Das Servo läuft ganz nach links, der Motor schaltet ab bzw., läuft nicht an.

Der Schalter **SE** hat jetzt eine Freigabe/Sperrfunktion für den Gaskanal

Achtung Vorsicht:

Dies ist ein einfaches Beispiel für einen Sicherheitsschalter, der den Kanal freigibt sobald der Schalter betätigt wird, egal wo der Gasknüppel tatsächlich steht!

2. Motor-Sicherheitsschalter in den Spezialfunktionen, „Sicher“ bzw „Override“

„Überschreiben eines Kanal“ mit einem Festwert wenn ein Schalter (hier **SE**) aktiviert ist

Schalter SE↓ und Wert -100 eingetragen, per Häkchen die Funktion freigeben!



Achtung: In der Mischerzeilen sieht man aber diesen Schalter nicht!

Achtung Vorsicht:

Auch hier, SE↓ gibt den Kanal 1 frei sobald der Schalter betätigt wird, egal wo der Gasknüppel tatsächlich steht!

3. Motorfreigabe mit Überwachung der Gasknüppelstellung, die bessere Lösung

Ein Sicherheitsschalter der den Gasknüppel überwacht und nur freigibt wenn er auch tatsächlich auf -100 (ganz unten) stehts sieht so aus:

Sticky Throttle Cut (Freigabe nur wenn Gas auch auf Null steht!)

2 Logische Schalter überwachen den Gas-Knüppel und merken sich die Stellung

	Funktion	V1	V2	UND Schalter
L1	a<x	Gas	-99	----
L2	OR	L2	L1	SF1
L3	---	----	0	----

Im Mischer: Kanal 3 ist hier Gas und wird nur freigegeben wenn er tatsächlich AUS ist.

CH2	
CH3	[I3]Gas Gewichtung(+100%) R MAX Gewichtung(-100%) Schalter(!L2) (Safe)
CH4	

4. Motorfreigabe mit Überwachung der Knüppelstellung, die bessere Lösung

Ein Sicherheitsschalter der den Gasknüppel überwacht und nur freigibt wenn er auch tatsächlich auf -100 (ganz unten) stehts sieht so aus:

2 logische Schalter, ein FlipFlop und die Spezial Funktion Override

Log. Schalter **L4** überwacht die Position des Gasknüppels (hier den Inputs [I]Gas)

Log. Schalter **L5** setzt ein Flipflop, merkt sich damit wenn die Position auf -100% (ganz unten) ist.

Spezialfunktionen **SF3** überschreibt (Override) den Gaskanal (hier Kanal 1) mit -100% falls er **nicht** unten ist.

Schalter **SF↓** Sperre (Schalter zu mir her, oben) **SF↑** Freigabe (Schalter von mir weg, unten)

Sticky Throttle Cut (Freigabe nur wenn Gas auch auf Null steht!)

L4	a<x	[I]Gas	-98	SF↑
L5	SRFF	L4	SF↓	----

SF3	IL5	Override CH1	-100	<input checked="" type="checkbox"/> EIN
-----	-----	--------------	------	---

Achtung: In der Mischerzeilen (hier Kanal 1) sieht man aber diese Funktion nicht!

Beispiel: Mischer mit Offset und Weight anpassen

Wir wollen zu einem Kanal einen bestimmten Poti-Anteil dazumischen.

Das macht man im Mischer mit Addiere (+=)

Das Poti soll aber nur positive Werte liefern und auch nur einen Anteil von 0-20% dazumischen.

Hintergrund:

Jeder Analogkanal (auch Poti) liefert -100% bis +100%

Mit Gewichtung und Offset wird der Kanalbereich angepasst,

mit Limits 7/12 auf die tatsächliche Drehrichtung, Mitte und Endlagen des Servos begrenzt,

egal was der Mischer für Werte berechnet hat.

Achtung neue Mischerberechnung ab openTx2.0

[(Quelle * Gewichtung) + Offset]= Mischerwert

Ein Poti liefert -100% bis +100%, das ist ein Bereich von 200%

Wir wollen 0-100% haben, also ist die Gewichtung = $100\% / 200\% = 50\%$

Der Offset ist in der Mitte des neuen Bereichs von 0-100% also bei 50%

Damit liefert das Poti jetzt nur noch Werte von 0 bis 100%

Wir wollen aber nur einen Bereich von 0% bis 20% haben,

damit ist die Gewichtung = $20\% / 200\% = 10\%$

Der Offset ist in der Mitte des neuen Bereichs von 0-20% also bei 10%

```

MISCHER 7/32      6/12
CH7
CH8 Ele 100
  += P1 10      ELE
CH9
CH10
CH11

```

Kanal 8, zum Höhenruder (Ele)

Addiert (+=) man einen Potiwert von 0-20% dazu.

Freigegeben wird das Poti mit einem Schalter

(hier ELE-Schalter, kann auch ein beliebiger anderer Schalter sein)

```

MISCHER 7/32      6/12
CH7
CH8 Ele 100
  += P1 15      ELE
CH9
CH10
CH11

```

Wollen wir aber positive und negative Werte haben

z.B. Min -15% bis Max +15%

dazu brauchen wir keine Offset-Verschiebung, denn

die Mitte ist des neuen Bereichs $-15\% - +15\% = 0$

```

MISCHER 7/32      6/12
CH7
CH8 Ele 100
  += TrmE 60
CH9
CH10
CH11

```

Das könnte man aber auch gleich per Trimmung machen.

Die Trimmung liefert -25% bis +25% den wollen wir auf -15% bis +15% reduzieren.

Berechnung: $15\% / 25\% = 0.6$ also Gewichtung 60%

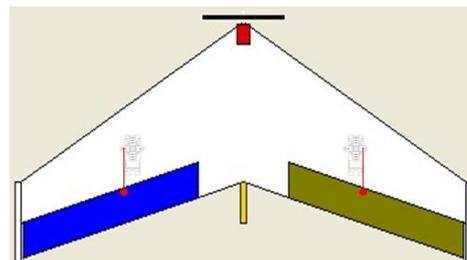
Damit verbrauchen wir keinen Schalter und kein Poti.

Hintergrund: Wir können jeden Trimmknopf frei verwenden, nicht nur wie hier den TrmE-Trimmknopf zum Ele Stick.

Trimmen kann man komplett frei zuordnen und parametrieren!

Beispiel: Servos für Delta und V-Leitwerk richtig mischen

In praktisch jeder Fernsteuerung gibt es die Funktion Deltamischer und V-Leitwerk. Hier werden 2 Kanäle miteinander gegenläufig gemischt. Höhenruder und Querruder. Aber egal was man macht wenn man diese Funktion aufruft, die Servos laufen einfach immer falsch rum. Das passt praktisch nie und dann wird wild rumprogrammiert.



Eigentlich ist das ganz einfach wenn man **2 Grundregeln** beachtet und genau in dieser Reihenfolge vorgeht. **Dieses Vorgehen gilt für alle programmierbaren Mischer in allen Fernsteuerungen.**

Hintergrund: Die **Ruderlaufrichtung** ist abhängig von der Einbaulage der Servos und davon ob das Gestänge am Ruderhorn links oder rechts angeschlossen ist.

Das Anpassen der **Ruderlaufrichtung** wird **grundsätzlich nur** in den Servo-Limit-Menüs gemacht und nicht in den Mixern! Mixer müssen mathematisch korrekt rechnen, positive Mischer-Werte sollen Ruder nach **oben** bzw. **rechts** bewegen. Servo-Limits passen dann die mathematisch errechneten Mischer-Werte den tatsächlichen physischen Einbaulagen und Drehrichtungen der Servos so an, dass die **Ruderwirkung** stimmt. Oft sind die 2 Servos symmetrisch gespiegelt eingebaut, so dass eins links und das andere rechts rum laufen muss, damit die **2 Ruder** gleich laufen. Man beachte ich spreche von **Ruder-Laufrichtung** und nicht von Servolaufrichtung!

Wie das Servo letztlich dreht ist völlig egal, ich brauche immer die **richtige Ruderwirkung!**

In allen Fernsteuerungen und Flugzeugen gilt bei Kanal-Mixern die Vereinbarung, dass positive Werte (+100%) ein Ruder nach **oben** bzw. nach **rechts** bewegen soll.

1. Zuerst wird die gleichlaufende Funktion, hier Höhenruder, eingestellt.

Knüppel Höhe ziehen und beide Mischer-Kanäle für Höhe bei Gewichtung auf +100% einstellen! Wenn am Höhenruder gezogen wird, **müssen** beide **Kanal-Mischer** in Richtung +100% gehen.

Jetzt wird im Servo-Limit-Menü die Laufrichtung für jeden Kanal einzeln solange umgedreht / Servo-Reverse bis beide Ruder nach oben gehen. Je nach Fernsteuerung kann da stehen: Norm, Rev, --- Inv, $\rightarrow \leftarrow$, $\uparrow \downarrow$ oder sonstige Sonderzeichen für Servo-Umkehr. Oder für die Servowege kann stehen +100 +100, -100 -100, +100 -100, -100 +100, je nach Einbaulage und Anschluss am Ruderhorn.

Das ist aber völlig egal, Hauptsache das **Ruder** geht „**richtig**“ rum.

Ab jetzt wird nichts mehr im Servo-Limit-Menü gedreht oder invertiert!

Nur Servo Mitte und Min/Max-Wege werden angepasst, damit sie nicht mechanisch auflaufen

2. Dann wird die gegenläufigen Funktion, hier Querruder, eingestellt.

Das wird in den 2 Querruder-Mixern gemacht da diese Funktion dem jetzt schon richtig laufenden Höhenruder gegenläufig dazugemischt wird.

Knüppel Querruder voll rechts geben, Ruder rechts muss nach oben gehen (Ruder links mal noch egal!)

Am Kanal-Mischer für rechts jetzt Gewichtung +100 einstellen, bis das rechte Ruder nach oben geht.

Knüppel Querruder weiter voll rechts halten, Ruder links muss nach unten gehen. Am Kanal-Mischer für links jetzt Gewichtung -100 einstellen, bis das linke Ruder nach unten geht. Beide Querruder-Mischer haben jetzt für die Querruderfunktion unterschiedliche Vorzeichen, somit gegenläufige Funktion.

Jetzt noch die maximalen Wege und Mischeranteile anpassen. Fertig, das wars.

Beispiel: Mit Mischern 2 Motoren auf Gleichlauf einstellen

An einem Flugmodell sind 2 Außenläufermotoren mit 2 ESC Antrieben. Leider laufen die beiden Motoren im Flug nicht exakt gleich. Gut wäre es wenn man sie im Flug um ca 5-10% fein nachtrimmen könnte und dadurch nicht dauernd per Seitenruder dagegen halten muss.

Dazu gibt es mehrere Möglichkeiten: Mit einem Poti oder mit einem freien Trimmer

Hintergrund:

Poti liefern -100% 0% +100% Trimmer liefern standardmäßig -25% 0% +25%

Das Poti soll **bei positiven** Werten den **rechten Motor beschleunigen** und

Das Poti soll **bei negativen** Werten den **linken Motor beschleunigen**

CH6= Gas + (+ Poti * + Gewichtung) für $x > 0$

CH7= Gas + (- Poti * - Gewichtung) (Minus mal Minus = Plus!) für $x < 0$

Trimmer-Taster kann man auch frei zuordnen. Meist braucht man den Seitenruder-Trimmer gar nicht. Sollte doch mal ein Trimmwerte gebraucht werden, so kann man den auch im Limitmenü 7/12 direkt einstellen.

Anschlüsse: CH6 Motor links, CH7 Motor rechts

Mischer Untermenü



Quelle ist Poti P1 an Kanal 6
+10% Gewichtung
 $x > 0$ Nur bei positiven Werte der Quelle, sonst kommt 0

(eventl. noch per Schalter freigeben)



Quelle ist Poti P1 an Kanal 7
-10% Gewichtung
 $x < 0$ Nur bei negativen Werte der Quelle, sonst kommt 0

(eventl. noch per Schalter freigeben)

Mischer Hauptmenü



CH6 Gas Stick += (Addiere) Potiwert dazu
(+ Poti1 mit +10% Gewichtung) nur bei positiven Werten

Ch7 Gas Stick += (Addiere) Potiwert dazu
(- Poti1 mit -10% Gewichtung) nur bei negativen Werten

Anstatt einem Poti als Quelle kann auch die TrmS (Seitenruder-Trimmmung) verwenden.
TrmS liefert -25% %0 + 25% (Taster nach links=negativ, Taster nach rechts=positiv)
Gewichtung = 40% (10% = 25%*40%)

Dann kann man diese 2 Motor-Kanäle noch auf einen Sicherheitsschalter legen, damit die Motoren nicht ungewollt anlaufen wenn man versehentlich an den Gasstick kommt.

Die Motorfeintrimmung kann man mit einem Schalter noch komplett freigeben oder sperren.

Beispiel: Mischer Bereiche einstellen und berechnen im Detail, Kurven als Variante

Ein Stick gibt -100% bis +100% als Eingang an einen Mischer

Das soll in einen ganz bestimmten Bereich umgesetzt werden

Die vereinfachte Mischerberechnung = $[(\text{Source} * \text{Gewichtung}) + \text{Offset}] + \text{Trimm}$

Beispiel 1: Stick Eingang von -100% bis +100%

Mischer Ausgang -30% bis +50%

Berechnung: Eingangsbereich von -100% bis +100% = 200%

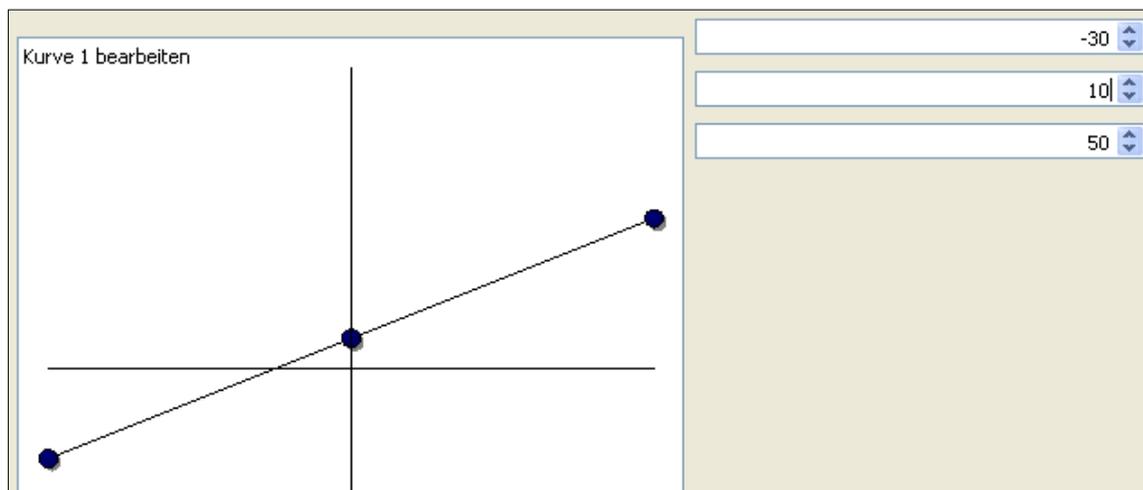
Mischer Bereich -30% bis +50% = 80% absolut

Gewichtung $80\% / 200\% = 40\%$

Offset = Mitte von -30% bis +50% = +10%

Einstellung: Weight = 40% Offset=10 (das ist gut machbar und einstellbar)

Alternative: 3-Punkt Kurve definieren, dann aber Weight= 100%, Offset=0



Beispiel 2: Stick Eingang von -100% bis +100%
Mischer Ausgang von +40% bis +65%

Berechnung: Eingangsbereich von -100% bis +100% = 200
Mischer Bereich +40% bis +65% = 25 absolut
Gewichtung $25\% / 200\% = 12,5\%$ gewählt 13%
Offset = Mitte von +40% bis +65% = +52,5% gewählt 53

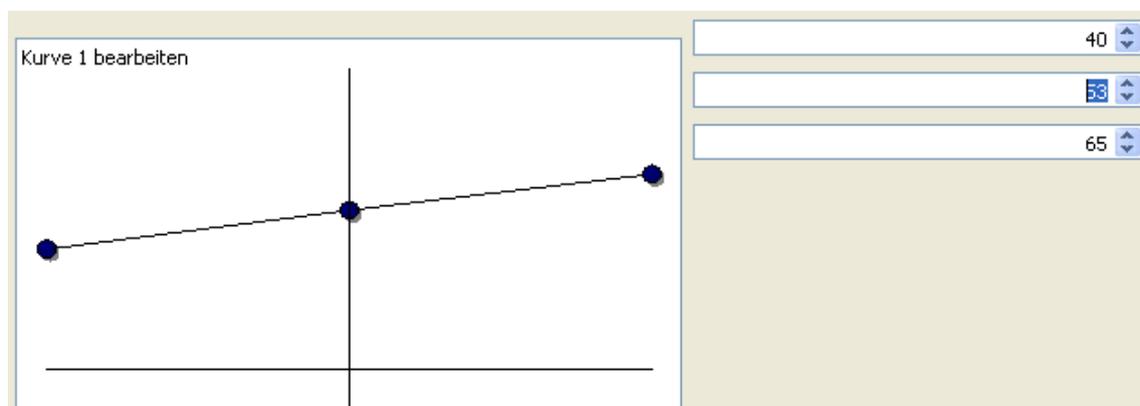
Einstellung: Weight = 13% Offset=53 (auch das ist noch gut machbar und einstellbar!)

Alternative: 3-Punkt-Kurve, dann aber Weight = 100%, Offset=0

1. Punkt: Links X= -100 Y= +40 3. Punkt: Rechts X+100 Y= +65

2. Punkt so verschieben, dass es eine Gerade ergibt,

oder ausrechnen (Strahlensatz) $(25 * 100 / 200) + 40 = 52,5$ also 2. Punkt : X=0, Y=53



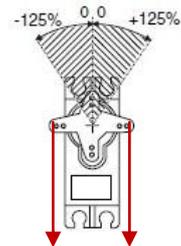
Mit Kurven kann man jeden Wertebereich den ein Mischer erzeugen soll einstellen, egal ob als Gerade oder gekrümmte Werte.

Servotrimm -Mitte, -Wege, -Umkehr, 7/12)

```

SERVOs 1500us 7/13
CH1 0.0 0- 0 → KU1 1500^
CH2 0.0 0- 0 → --- 1500^
CH3 0.0 0- 0 → --- 1500^
CH4 0.0 0- 0 → --- 1500^
CH5
CH6
CH7
SERVO WEG 1500us 7/13
CH1 0.0 -512.0 -512.0 → --- 1500Δ
CH2
CH3
CH4
CH5
CH6
CH7

```



Mit **[Ent Long]** in ein Untermenü um Trimmwerte eines Kanal als Subtrim zu übernehmen.

Mit Option **ppmus** Darstellung **aller** Kanal-Impulsbreite **anstatt von -100% bis +100% in μ s** im Hauptmenü, Servomonitor in 988 μ s bis 2012 μ s, in den Limits, -512 (= -100%) +512 (= +100%)

Das ist die zweit-wichtigste Anzeige. Hier werden die **mechanischen Servowege**, Servobegrenzungen links/rechts (Travel Adjust), Servo-Mittelstellung (Subtrim) und Servolaufrichtung (Servo-Reverse, INV) eingestellt. Es dient der Bewegungs-Begrenzung der Servos damit mechanische Grenzen nicht überfahren werden. Egal was vorher als Mischerwert errechnet wurde und wie groß auch die Werte sind, ob positive oder negative Werte. Hier geschieht die **Anpassung an die reale Welt und die Wege werden gnadenlos begrenzt.**

Entscheidend ist die Einbaulage und die Seite der Ruderanlenkung damit's "**richtig**" dreht.

Im Blockschaltbild sieht man dass dies erst nach den Kanalberechnungen durch die Mischer erfolgt, unmittelbar bevor die Signale für die Servokanäle erzeugt werden.

Jeder Kanal CH1..CH32 kann individuell eingestellt werden.

Bedeutung der Spalten: In der Statuszeile wird ein Hilfstext angezeigt

1. **Name** (max 6 Zeichen) kann frei vergeben werden z.B. QuerLi, QuerRe, Gas, Fahrw, Flap
Dieser Name erscheint dann im Kanalmonitor anstatt der Kanalnummer CHx
2. **Mitte, Subtrim, Offset:** Kanalmitte, bzw. Servomitte bzw. Trimmwerte
Die Servomitte kann Werte von **-100%** bis **+100%** annehmen mit einer feinen Auflösung von **0,1**. Damit hat man eine exzellente Feinauslösung für die Servo-Mittelstellung. Nur sehr hochwertige Servo können überhaupt so fein mechanisch auflösen!

Hinweis: Das kann man auch mit den Knüppel (Sticks) machen.

Einfach wenn die Kanalanzeige blinkt mit dem Knüppel (Stick) das Servo bewegen, dann mit **[ENTER LONG]** bestätigen und der Wert ist übernommen.

Ob das wirklich sinnvoll ist, ist eine andere Frage.

3. **Minimum channel Limit:** und
4. **Maximum channel Limit :** Kanalendanschläge, Servoendanschläge, Endpunkte .
Wenn die Funktion „**Erw. Limits**“ aktiviert ist sind die Grenzen hier +25% bis -125% für Minimum und -25% bis +125% für Maximum
Ansonsten +25% bis -100% und -25% bis 100%

Limits begrenzen den maximalen mechanischen Wege des Servos, egal welchen Wert die Mischerberechnung ergeben hat. Sie schützen Servo und Mechanik und verhindern ein blockieren von Servo oder Ruder.

Die Eingabe erfolgt wie immer, mit dem Cursor die Zeile/Spalte auswählen, mit [ENTER] die Eingabe aktivieren, dann mit den Cursors die Werte von -100 bis 100 ändern und mit [ENTER] Eingabe abschließen.

5. Falls die option **ppmus** ausgewählt,

Hier werden in den Spalten die Richtungszeichen (-> - <-) für die Servo Richtung nach rechts oder links angezeigt. Als Hinweis für die Trimmrichtung.

5a. **Kurve**: Hier kann auch eine Kurve **KV1.. KV32** aktiviert werden um Servobewegungen unabhängig von der Mischerberechnung zu beeinflussen, z.B Linearisierung des Kreisbogens vom Ruderarm, mechanisches Spiel ausgleichen, exakter Gleichlauf einstellen bei bautechnischen Mängeln.

6. **INV**: Servo-Revers, Servoumkehr, Invertierung eines Kanals,

Damit wird die Servodrehrichtung umgekehrt und dem tatsächlichen Ruderverlauf d.h. Servohebel/Wirkrichtung angepasst.

Je nach verwendeter Option in der Firmware steht hier

"INV" bzw "----" oder aber '<-' für Reverse "->" für Normal

Eingaben erfolgen wie sonst auch, einfach mit dem Cursor auf die Position gehen und mit [MENU] umschalten.

7. **PPM center value** (in Mikrosekunden).

Hier kann die Servo Mittelstellung in μs angegeben werden.

Je nach Servo-Hersteller hat ein Servo unterschiedliche Werte für Mittelstellungen von ca. 1450 μs bis ca. 1550 μs (Futaba, Graupner, Multiplex alt 1600 μs , usw.)

das kann hier korrigiert werden. Normal ist die Mitte bei 1500 μs

Das ist vor allem dann interessant wenn die Servowege auf beiden Seiten voll ausgenutzt werden sollen.

8. **Art der Begrenzung, Subtrimm-Typ, Kind of limits:**

Klassisch ('^') oder Symmetrisch ("=")

Klassische Limits ('^'): Die Min/Max Grenzen werden unabhängig von der Mitte (Sub-trimm) auf -100% bis 100% gehalten und die min/max Wege haben 2 unterschiedliche Steigungen (**Rote Kurve**). Die Kurve hat 2 Steigungen!

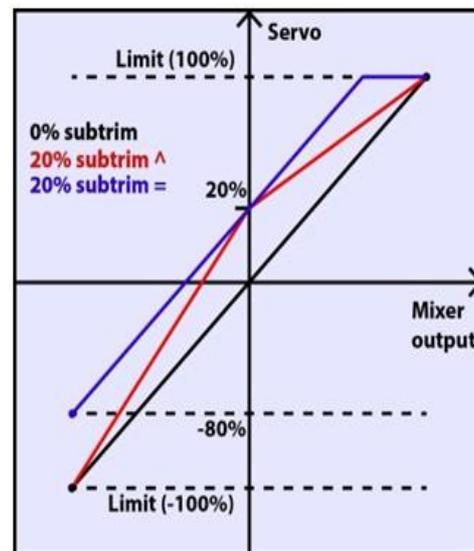
Man kann den vollen Weg ausnutzen.

Der Nachteil ist, dass gleiche positive und negative Mischerwerte zu unterschiedlichen Servowege führen.

Symmetrische Limits ("="): Die Min/Max Grenzen werden in Abhängigkeit von der Mitte (Sub-Trim) symmetrisch mit verschoben und behält die Steigung bei (**Blaue Kurve**). Die Kurve hat die gleiche Steigung!

Man kann nicht den vollen Weg ausnutzen.

Der Vorteil ist, dass gleiche positive und negative Mischerwerte zu gleichen Servowege führen.



Die Trimmsschritte in us:

Normaler Trimbereich $\pm 25\% = \pm 125$ Schritte = 125us

Erweiterer Trimbereich $\pm 100\% = \pm 500$ Schritte = 500us

Auflösung: Extra fein= 1us Fein = 2us Mittel = 4us Grob= 8 us

Diese einfache Umrechnung der Trimmsschritte in us gilt nur bei

Symmetrische Limits ("=") da nur hier die Kurve gerade ist.

Bei **Klassische Limits ('^')** wird die aktuelle Kurvensteigung mit verrechnet!

d.h. pro Schritt weniger als 1 us

Trimmwerte der Kanäle als Subtrim (Servo-Mittelstellung) übernehmen und abspeichern

Es gibt 3 Varianten um Trimmwerte als neue Servomitte (Subtrim) zu übernehmen.

Einzelnen Kanal, einzelne Knüppelstellung oder alle Kanäle gemeinsam übernehmen.

Nach jeder Subtrimübernahme hat man von der neuen Mitte wieder +/-125us Trimmweg!

1. Im Kanal mit [Enter Long] einen Trimmwert nur für diesen Kanal übernehmen

Hier Aufpassen! Wenn die Trimmwerte für 2 Kanäle gelten

(z.B. bei Querruder, Flaps usw.) dann auch den anderen Kanal nicht vergessen!

```
SERVO WEG 1500us 7/13
CH1 0.0 -512.0-512.0 → --- 1500Δ
CH2 Reset Servowerte 1500Δ
CH3 Kopie Trimm to Servo-Mitte 1500Δ
CH4 Kopie Stick to Servo-Mitte 1500Δ
CH5 0.0 -512.0-512.0 → --- 1500Δ
CH6 0.0 -512.0-512.0 → --- 1500Δ
CH7 0.0 -512.0-512.0 → --- 1500Δ
```

2. Ganz unten mit [Enter Long] alle Trimmwerte auf die Kanäle übernehmen

```
SERVO WEG 7/13
CH27 0.0 -512.0-512.0 → --- 1500Δ
CH28 0.0 -512.0-512.0 → --- 1500Δ
CH29 0.0 -512.0-512.0 → --- 1500Δ
CH30 0.0 -512.0-512.0 → --- 1500Δ
CH31 0.0 -512.0-512.0 → --- 1500Δ
CH32 0.0 -512.0-512.0 → --- 1500Δ
Kopie alle Trims to Servo-Mitte
```

Nach dem Kanal 32 gibt es noch die Funktion **Kopie alle Trims to Servo-Mitte**

Bzw (**Trims => Offsets**) (Offset = Subtrim = Servo-Mittelstellung)

um **alle** (erfolgten) Trimmwerte nach dem Flug als Subtrim zu übernehmen.

Einfach diese unterste Zeile anwählen, dann werden mit **[ENTER LONG]** alle Trimmwerte auf die entsprechenden Kanäle als Subtrim mit übernommen und die Trimmwerte selbst wieder auf Null gestellt.

Das entspricht einer Mittelstellungsverschiebung / Subtrimverschiebung des Servos!

Achtung aufpassen, Trimmwerte sind flugphasenabhängig!

Jede Flugphase FP0-FP8 hat in der Regel ihre eigenen etwas anderen Trimmwerte.

Wenn ich also hier die Trimmwerte der aktiven Flugphase auf die Subtrim des Servos übernehme verstelle ich ja die Mitte der Servos. Die stehen dann bei allen Flugphasen so aus der Mitte!

3. Eine Knüppelstellung (Stick) als Subtrim für einen Kanal übernehmen

Man kann auch anstatt von Trimmwerte einen Knüppel (Sticks) verwenden.

Knüppel auf Position halten und dann diese Stellung als Subtrim des Kanals übernehmen.

```
SERVO WEG 1500us 7/13
CH1 0.0 -512.0-512.0 → --- 1500Δ
CH2 Reset Servowerte 1500Δ
CH3 Kopie Trimm to Servo-Mitte 1500Δ
CH4 Kopie Stick to Servo-Mitte 1500Δ
CH5 0.0 -512.0-512.0 → --- 1500Δ
CH6 0.0 -512.0-512.0 → --- 1500Δ
CH7 0.0 -512.0-512.0 → --- 1500Δ
```

Servo-Min und Servo-Max genauer betrachtet:

Die Servo-Min und Servo-Max sind nicht einfache starre Grenzen, da steckt mehr dahinter.

Hinweis: Andere Sender (DX9...MZ18) habe hier 4 Werte zum einstellen.
2 für den eigentlichen, linearen Servoweg, 2 für einen Grenzwert-Limiter
Das dann kann dazu führen dass schon bei 50% Knüppelweg ein Servo begrenzt, was dumm ist, aber historisch bedingt.

OpenTx berechnet und arbeitet hier anderst, da alles über Mischerzeilen läuft und mehrere Mischerzeilen auf einen Kanal wirken können.

Servo-Min und Servo-Max sind die Servoweg- Grenzen die nicht überschritten werden. Daraus werden intern mit dem Kanal Faktoren berechnet der zu den Min/ Max Servowege führen. (die Faktoren sieht man nicht, z.B. 37,5%Min Servo / 50%Kanal = 75%Faktor)

Die Berechnung läuft so:

%Output des Kanals * %Faktor = % Min/ Max Servoweg

Beispiel:

100% Kanal * 100% Faktor = 100,0% Min/ Max Servoweg

100% Kanal * 50% Faktor = 50,0% Min/ Max Servoweg

50% Kanal * 75% Faktor = 37,5% Min/ Max Servoweg

50% Kanal * 125% Faktor = 62,5% Min/ Max Servoweg

Und für was soll das jetzt gut sein?

Das sieht man erst wenn man mit mehreren Mischerzeilen pro Kanal arbeitet:

Bei 3 Mischern additiv auf einen Kanal sieht das dann so aus:

(50% Mischer1 + 40%Mischer2 + 35%Mischer3) * Faktor = 62,5% tatsächlicher Servoweg

Damit bleiben die drei %Mischer-Verhältnisse untereinander immer gleich, selbst wenn ich dann das Servolimit von 50% auf 65% oder auf 40% ändere

Bisheriger Servoweg: (0,5+0,4+0,35) * Faktor = 50% tatsächlicher Servoweg

Mehr Servoweg: (0,5+0,4+0,35) * Faktor = 65% tatsächlicher Servoweg

Weniger Servoweg: (0,5+0,4+0,35) * Faktor = 40% tatsächlicher Servoweg

Wenn ich einen anderen Servoweg brauche genügt es bei Servo-Min oder Servo-Max den Wert zu verändern, soweit es mechanisch sinnvoll ist.

Meine eigentlichen Mischerverhältnisse untereinander bleiben erhalten!

Das fällt erst auf, wenn, so wie bei Seglern, viele Mischer ineinander greifen.

Erst dort wird einem dann klar wie genial das Konzept ist.

Bei nur einer Mischerzeile im Kanal fällt das gar nicht auf.

Natürlich gibt es max Servo-Grenzen

Je nachdem was für max Servowege eingestellt wurden:
(siehe bei Modelleinstellungen, Erweiterte Wege 150%)

Normale Servowege: $\pm 100\%$ = 988us und 2012us ($\pm 512\mu\text{s}$ von 1500us)

$\pm 125\%$ = 860us und 2160us ($\pm 640\mu\text{s}$ von 1500us)

Erweitere Servowege: $\pm 150\%$ = 732us und 2268us ($\pm 768\mu\text{s}$ von 1500us)

Die tatsächliche Grenze gibt aber immer die Mechanik vor!

Kurve für ein Servo

Jedes Servo kann noch eine beliebige Kurve erhalten.

Das ist viel flexibler als nur lineare Servowege oder feste Grenzwert Limiter.

Damit kann man auch noch Ruderwege linearisieren.

SERVOs	1500us	Curve	7/13
CH1	0.0	-100.0 - 100.0 → ---	1500Δ
CH2	0.0	-80.0 - 75.0 → ---	1500Δ
CH3	0.0	-100.0 - 100.0 → ---	1500Δ
CH4	0.0	-100.0 - 100.0 → ---	1500Δ
CH5	0.0	-100.0 - 100.0 → ---	1500Δ
CH6	0.0	-100.0 - 100.0 → ---	1500Δ
CH7	0.0	-100.0 - 100.0 → ---	1500Δ

Konfiguration	Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Spezial Funktionen	Telemetrie
Name	Mitte	Min	Max	Richtung	Kurve	PPM Mitte	Lineare Mitte		
Kanal 1 Fahrw	0,0	-100,0	100,0	---	Kurve(2)	1500	<input type="checkbox"/>		
Kanal 2 Hoehe	0,0	-80,0	60,0	INV	----	1500	<input checked="" type="checkbox"/>		
Kanal 3 Klapp1	0,0	-100,0	100,0	---	----	1520	<input checked="" type="checkbox"/>		
Kanal 4	0,0	-100,0	100,0	---	----	1500	<input type="checkbox"/>		
Kanal 5	0,0	-100,0	100,0	---	----	1500	<input type="checkbox"/>		
Kanal 6	0,0	-100,0	100,0	---	----	1500	<input type="checkbox"/>		
Kanal 7	0,0	-100,0	100,0	---	----	1500	<input type="checkbox"/>		

Kurven eingeben (8/13)



Kurven sind ein ganz wichtiger Bestandteil in der Beschreibung wie Ausgangssignale von ihren Eingängen beeinflusst werden.

Das beste Beispiel ist wohl die Mischerfunktion von Gaskurve und die Pitchkurve beim Hubschrauber. Aber es gibt beliebig viele andere Anwendungen für Kurven z.B. Ruderdifferenzierungen, Landeklappen, Wölbklappen in Abhängigkeit von verschiedenen Flugphasen, Fahrwerkklappen, Doorsequenzer, Linearisierung von Drehbewegungen usw. Man kann sogar mit globalen Variablen Kurvenwerte anzeigen, variabel verändern und im Flug anpassen in dem mit der Gewichtung=Verstärkungsfaktor die Steilheit angepasst wird.

Es gibt 32 frei definierbare Kurven mit jeweils 2-17 Stützpunkten.

Kurven kann man an 3 Stellen anwenden, dabei gibt es keine Einschränkungen der Art.

- Input-Signalvorverarbeitung Seite 5/13
- Mischer/Kanal Verarbeitung Seite 6/13
- Servos Bewegungen anpassen Seite 7/13

Es sind auch immer alle Kurventypen möglich:

Standard = variable Y-Werte und fixe X-Werte mit 2-17 Stützpunkte

Custom = variable Y-Werte und variable X-Werte mit 2-17 Stützpunkte

Kurven mit 2-17 Stützpunkten

Standard-Typ: fixe X-Werte, variable Y-Werte

Es gibt Kurven mit **festen X-Werten** (horizontal/waagrecht), die Y-Werte (vertikal/senkrecht) sind variabel und können dabei eingegeben werden.

Hier nur mal ein Auszug von ein paar festen Kurvenpunkten

2pt Kurve hat die X-Positionen -100% +100%

3pt Kurve hat die X-Positionen -100%, 0%, 100%.

...

5pt Kurve hat die X-Positionen -100%, -50%, 0%, 50%, 100%.

.....

9pt Kurve hat die X-Positionen -100%, -75%, -50%, -25%, 0%, 25%, 50%, 75%, 100%.

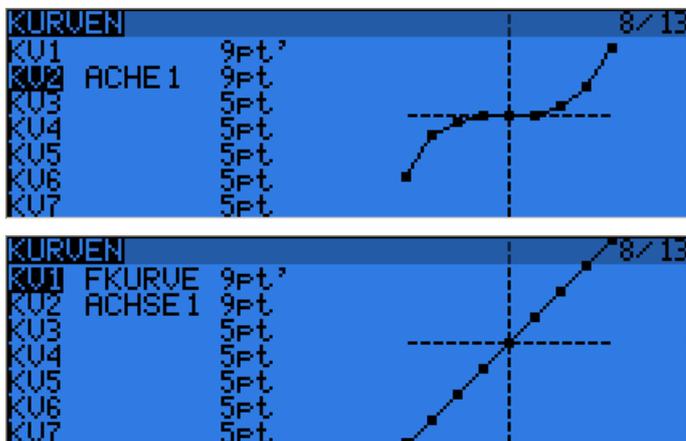
.....

17pt Kurve hat die X-Positionen -100%, -88%, -75%, -63%, -50%, -38%, -25%, -13%, 0%, 12%, 25%, 37%, 50%, 62%, 75%, 87%, 100%.

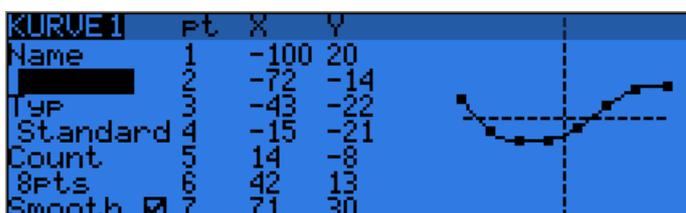
Custom-Typ = Variable X-Werte und variabel Y-Werte

Dann gibt es noch Kurven mit **variablen X-Werten** und **variablen Y-Werten**
 Hier können **beide Koordinatenpunkte (X , Y)** frei eingegeben werden.

Kurven editieren



Hier kann man die 32 Kurven auswählen, alle je von 2-17 Punkten
 Fährt man mit dem Cursor runter werden die Kurven gleich rechts dargestellt.
 Mit **[ENTER]** kommt man dann in das Untermenü um die Kurve zu editieren.
 Abhängig vom Kurventyp 2- 17 werden die X-Stützpunkte als 2-17 Punkte dargestellt.

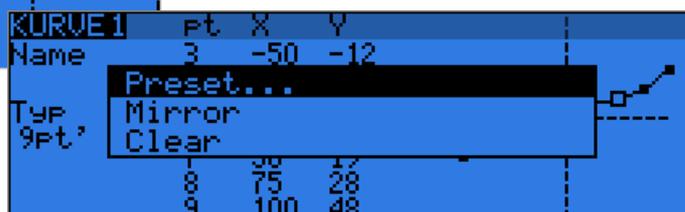
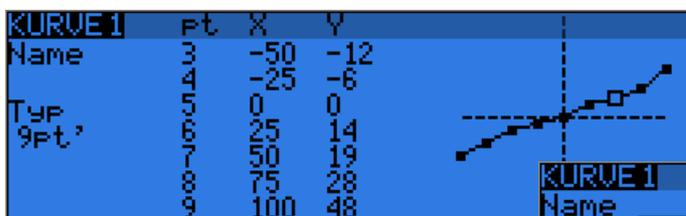


Kurventypen:

Standard: mit festen X-Werten, und variablem Y-Wert

Custom: mit **variablen X-Werten** und **variablen Y-Werten**

Mit **Smooth** kann werden die Kurven noch mit einer Splinekurve abgerundet/verrundet.



Wenn man **in** den Zahleneingaben ist und drückt dann **[Enter Long]**,
 kommt das Menü für Preset, Kurve spiegeln und löschen

Es gibt dort feste Voreinstellungen, Preset- Kurven/Gerade mit 11° 22° 33° 45°

Dann kann man noch Kurven mit **Mirror** an der **X-Achse** spiegeln.

Y-Werte eingeben bei festen X-Werten

Die X-Stützpunkte werden mit **[+]** / **[-]** ausgewählt, dann **[ENTER]**, und mit **[+]** / **[-]** die Y-Werte eingegeben die Kurve passt sich grafisch an.

Variable X und Y-Werte eingeben

[+]/**[-]** einen Punkt auswählen (kleines Quadrat)

Mit **[ENTER]** blinkt der ausgewählte Punkt

Dann kann man die X/Y Koordinaten eingeben:

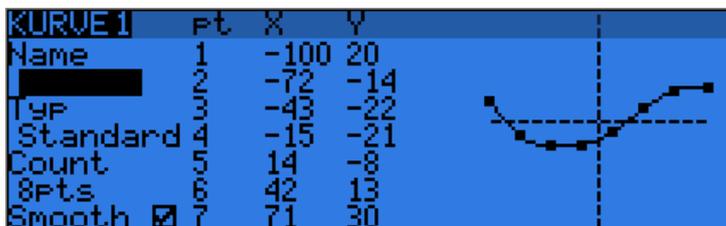
Mit **[+]** / **[-]** die X-Werte, **[ENTER]** mit **[+]** / **[-]** die Y-Werte

Mit **[ENTER]** wird der Punkt übernommen.

Dann der nächste Punkt ausgewählt usw. bis die freie Kurve fertig ist.

Mit **Zweimal [EXIT]** verlässt man die Kurvengabe und kommt ins Kurven-Hauptmenu 8/12 zurück

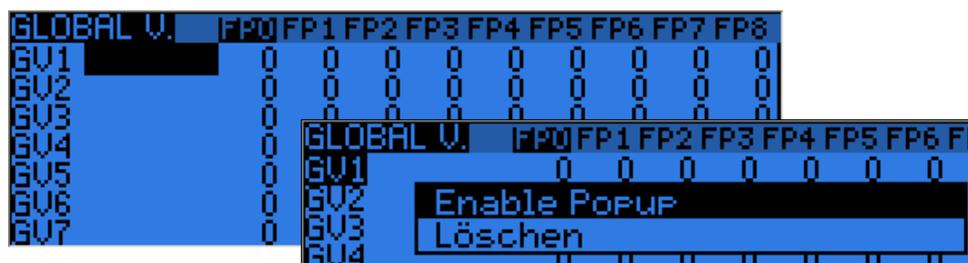
Auswahl der Kurventypen für feste oder variable X-Werte



Umschalten der Kurvenarten und Kurventypen

Einfach im Feld Typ auf editieren **[Enter]** gehen, dann kann man durchscrollen und die Kurvenart auswählen von 2pt bis 17pt

Globale Variablen GV1-GV9 (9/13)



Mit **[Ent Long]** ein Untermenü

GVAR Anzeigebox im Hauptscreen freigeben

GV1! Das „!“ Wert anzeigen im Popup-Fenster

Es gibt 9 Globale Variablen **GV1 .. GV9**
 und das für jeden der 9 Flugzustände **FM0 .. FM8**
 somit stehen $9 \times 9 = 81$ Variablen mit Wertebereich **-1024 bis +1024** zur Verfügung

Sie erweitern die Möglichkeit um Eingangswerte von Mischern, Dualrate, Expowerte zu beeinflussen ganz erheblich.

Die Idee hinter den globalen Variablen ist, dort wo Werte mehrfach gebraucht werden oder gemeinsam verändert werden müssen, dies mit einer Variablen global machen zu können.

Globale Variablen können feste Werte haben, sie können aber auch jederzeit verändert werden, indem man Ihnen einen variablen Analogwert zuweist.

Damit können Einstellungen für bestimmte Funktionen im Flug verändert werden.

z.B. variable Querruderdifferenzierung im Flug verändern.

Bei openTx für Taranis kann **jede** globale Variable für **jede** Flugphase andere Werte haben

Jede Globale Variable kann einen eigenen Namen haben (max 6 Zeichen).

Diese sind dann in den Untermenüs der Flugphasen zugeordnet.

Dazu ist diese Eingabe-Tabelle vorhanden.

GVAR festen Wert zuweisen

Den Wert einer globalen Variable kann man im Menü Globale Variablen (9/13) ansehen und festlegen, und zwar mit [+]/[-]. Beenden der Eingabe mit [Enter] oder [EXIT].

Mit **[Enter Long]** umschalten von Zahlen nach Variablen!

Damit haben die globalen Variablen erst mal einen festen Wert.

Der Sinn ist, dass man diesen Wert an mehreren Stellen gleichzeitig verwenden kann. Zum Beispiel für das Gewicht von Quer- und Höhenruder. Ändert man den Wert der entsprechenden Variable, wird das Gewicht von Quer- und Höhenruder gleichzeitig angepasst.

GVAR Wert von anderer Flugphase übernehmen

Globale Variablen können auch den Wert derselben Variable einer anderen Flugphase übernehmen.

Dazu bei der Eingabe des Wertes [Enter Long] drücken und mit [+]/[-] die

Quell-Flugphase (FP0-FP8) auswählen.

GVAR veränderbare Werte zuweisen

Mit den Spezialfunktionen (11/12) kann man den globalen Variablen unterschiedliche Werte zuweisen, in Abhängigkeit von Bedingungen, Schalterstellungen, Telemetriewerten usw. Wie das funktioniert, steht im Kapitel Spezialfunktionen.

Dazu gleich mal ein Beispiel, Details in den Spezialfunktionen

GVAR in den Spezialfunktionen veränderliche Werte zuweisen

Globale Variablen werden in den Spezial Funktionen 11/13 aufgerufen, mit einem Analogwert versorgt und können damit verändert werden.

Das **verändern** kann per **ON/EIN** dauernd freigeschaltet sein oder aber nur wenn ein Schalter aktiv ist. Dann können die GVARs auch noch komplett gesperrt bzw freigegeben werden mit

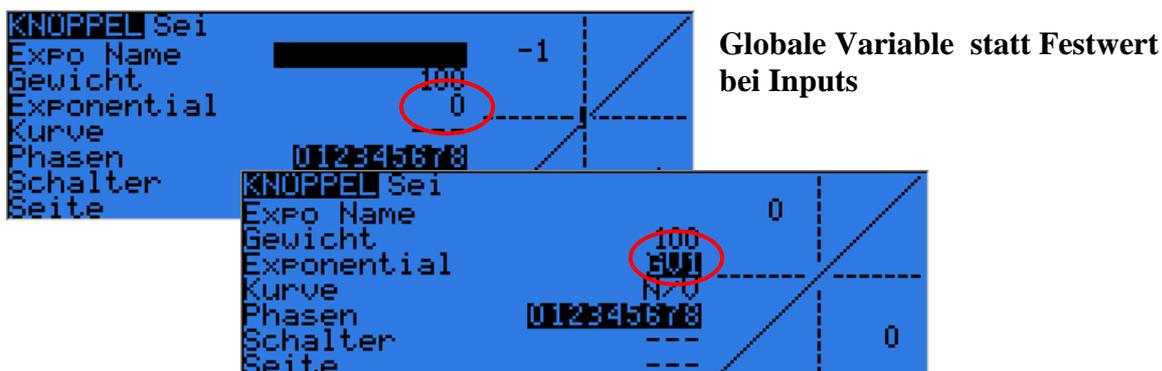
Als Quelle kann man alle Analogwerte nehmen.

Mit [**Enter Long**] umschalten von Zahlen nach Variablen!

Rud, Ele, Thr, Ail, S1, S2, LS, RS, TmrR, TmrE, TmrT, TmrA, alle PSx,
alle Schalter SA-SH, MAX, 3POS, CYC1, CYC2, CYC3, TR1-TR16, CH1-CH32.



Anwendung von Globalen Variablen GVx



Globale Variablen können an vielen Stellen verwendet werden. Fast überall da wo feste Werten verwendet werden kann man diese durch variable Werte GVx ersetzen.

Dort wo eine feste Zahl steht und durch eine Globale Variable ersetzt werden soll, kann man mit **[Enter LONG]** umschalten von Zahl auf **GVx** und zurück.

Mit **[+]** und **[-]** kann man dann die 9 möglichen **GV1..GV9** auswählen.

Mit **[Enter LONG]** kann man das auch abbrechen und wieder auf den alten Festwert zurückschalten.

Wird nun der Wert einer globalen Variablen verändert, erscheint kurz ein Fenster mit dem neuen Wert der Globalen Variablen (mit **Enable Popup**) **GV1!** Das „!“ heist mit Popup-Fenster



Beispiel: DR/Expo und Anwenden von globalen Variablen

Nun einmal ein etwas ausführlicheres Beispiel:

Wir wollen Dualrate/Expo mit dem Schalter **GEA** aktivieren und mit 65% Dualrate und 35% Expoanteil beginnen. Es soll nur der positive Anteil der Kurve ($x > 0$) wirken.

Das geht ganz einfach:

Untermenü für DR/Expo 6/13 (Knüppel) und dort die Werte eingeben:



Linke Seite die Eingabewerte, wie weiter oben erklärt.

Rechte Seite die Kurve und wenn man dann noch Rud bewegt sieht man die Ausgabewerte 0 bis ...

Im Hauptmenü erscheint dann genau das:



Soweit ist das alles klar. Wird **GEA** betätigt wirken die eingestellten Werte mit 65% und 35% und die halbe pos. Expokurve.

Ist **GEA** aus, wirkt Dualrate nicht, Weg = 100% , keine Expokurve und die gerade Kurve.

Mit **[+]/ [-]** kann man den Wert für das Dualrate direkt ändern. So weit so gut.

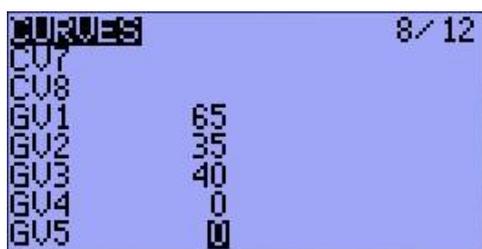
Beispiel: Anwenden von Globalen Variablen in 4 Schritten

Wir wollen die festen Werte für Dualrate und Expo variabel gestalten.

Dazu brauchen wir 2 globale Variablen **GV1** und **GV2**.

1. Vorbelegen

Unter Globale Variablen 9/13 belegen wir jetzt mal **GV1** mit 65% (für Dualrate) und **GV2** mit 35% (für Expo) vor. Das sind die gleichen Startwerte wie vorher (Zufall, muss aber nicht so sein), damit man das versteht (GV3 mit 40 ist für ein anderes Beispiel).



Vorbelegen der GVx muss nicht sein, ist aber sicherer, denn haben wir schon mal fixe, gute, passende Startwerte.

2. Freischalten und versorgen

Jetzt müssen wir die globalen Variablen freischalten. Entweder dauernd mit **ON** oder über einen Schalter und angeben woher **GV1** und **GV2** ihre Werte bekommen sollen.

Dazu sind die Spezial Funktionen 11/13 da.



Dauernd ein mit **ON** oder

Schaltbar mit einem Schalter ist der bessere Weg, dann kann man den Wert der GVx nicht versehentlich ändern!

Adjust GV1 wird von Poti **P1** und **GV2** von Poti **P2** mit Werten versorgt. Damit sind die GVx mal scharf geschaltet.

3. Anwenden

Im Untermenü von DR/Expo 5/13 (Knüppel) müssen wir jetzt statt den Festwerten 65% und 35% die globalen Variablen **GV1** und **GV2** eintragen.

Einfach mit den Cursor auf diese Werte gehen, mit **[MENU LONG]** umschalten und **GV1** und **GV2** auswählen. Das wars, jetzt sind wir bereit.



Vorbelegt sind die **GV1** mit 65% und **GV2** mit 35%. Wenn wir **GEA** aktivieren wird Dualrate und Expo damit berechnet. Wenn nicht, Weg 100% und Expo 0%

Soweit ist das nichts anderes als normales DR/Expo mit einem Schalter. (22 und 59 sind X-Werte, da hab ich das Ruder bewegt)

4. Benutzen der globalen Variablen:

Wenn wir jetzt aber an **P1** oder **P2** drehen kommt kurz einen Anzeigebildschirm mit den neuen Werten und schon sind die neuen Dualrate-Werte von **P1** an **GV1** und die Expowerte von **P2** an **GV2** übergeben und aktiv. Das wars, wir können aktiv im Flug neue Werte erzeugen und übernehmen.



Ganz einfach, oder?

Was man sonst noch alles damit anstellen kann, darauf kommt an erst so nach und nach.

Globale Variablen gibt es meines Wissens nirgends in den Super-High-Tech-Kompliziert-Umständlich-Anlagen.

Beispiel: Globale Variablen vorverarbeiten und Bereich einschränken/anpassen

Normal werden Globale Variablen in den Spezialfunktionen gleich mit Analogwerten versorgt.

Beispiel: **ON Ändere GV1 S1.**

Damit haben Globale Variablen von aber den vollen Bereich -100% bis +100% von S1 erhalten.

Das ist aber oft viel zu viel da man meist nur eine kleine Korrektur durchführen will und damit die GVAR nur einen Bereich von 0-10% oder 0-25% überstreichen soll.

Will man diesen Bereich einschränken, so dass z.B. die globale Variable nur noch Werte von 0% - 25% liefert, so macht man das in einem freien Kanal als Vorverarbeitung mit einer Mischerzeile. Oder via Kurve, da hat man dann noch mehr Möglichkeit

Das geht nach bekanntem Muster der Mischerberechnung:

Mischer-Berechnung = [(Source * Gewichtung) + Offset

Weight: 25%/200% = 12,5 Offset = Mitte des neuen Bereichs 0% - 25% = 12,5 (gewählt 13)

Analogwert → Vorverarbeitung in CHx → Spezialfunktionen → ON Ändere GVn CHx
 S1 CH12= [(S1* 13)+ 13] GV erhält Analogwert ON Ändere GV4 CH12



Das ergibt einen Bereich der GV4 von 0-25% , Vorverarbeitung in Ch12



Anwendung: variable Ruder-Differenzierung für Querruder wo nur Festwerte oder GVars möglich sind, Dualrate/Expo mit engem Bereich

Man kann auch eine Kurve verwenden um einer GVAR einen Bereich zuzuweisen!

Siehe ausführliches nachfolgendes Beispiel mit Companion

- S1, S2 vorberechnet
- S1,S2 mit Kurven

Beispiel: Globale Variablen auf Gewichtung und Expo anwenden

Hier mal ein „Schulbeispiel“ für das Verständnis von **globalen Variablen und Kurven**.
 Der Ruderweg für Quer und Höhe soll von 50% bis 100% variabel mit Poti S2 einstellbar sein
 Das Expo für Quer und Höhe soll von 0 bis 40% variabel mit Poti S1 einstellbar sein.

Um mit globalen Variablen arbeiten zu können sind immer 3 Schritte nötig:
 → 1. Werte vorberechnen, → 2. GV zuweisen, → 3. GV anwenden.

1. Expo mit S1, Gewichtung mit S2 vorberechnen

Expo: S1 soll einen Weg von 0 bis 40% machen (statt -100% bis +100%)
 Gewichtung: S2 soll einen Weg von 50 bis 100% machen (statt -100% bis +100%)
 Das müssen wir also erst mal vorberechnen, dazu braucht man je eine freie Mischerzeile
 (hier mal in CH10 und CH11)
S1: von 0 bis 40% sind 40% Berechnung → $40/200 = 0,20 = 20\%$ Gewichtung,
 Die Mitte von 0 bis 40% ist 20% = Offset
S2: von 50% bis 100% sind 50% Berechnung → $50/200 = 0,25 = 25\%$ Gewichtung
 Die Mitte von 50 bis 100% ist 75% = Offset

CH9	
CH10	S1 Gewichtung (+20%) Offset (20%) (Exp 0-40)
CH11	S2 Gewichtung (+25%) Offset (75%) (R50-100)
CH12	

2. GV zuweisen

Diese neuen min und max Kanalwerte in CH10 und CH11 die aus S1 und S2 entstanden sind
 schreiben wir in die Globalen Variablen GV3, GV4 rein. Das geht in den Spezialfunktionen
 Adjust GV3 von CH10 (aus S1 für Expo) Adjust GV4 von CH11 (aus S2 für Gewichtung)

SF11	----	Override CH1	0	<input type="checkbox"/> EIN
SF12	EIN	Adjust GV 3	Quelle CH10	<input checked="" type="checkbox"/> EIN
SF13	EIN	Adjust GV 4	Quelle CH11	<input checked="" type="checkbox"/> EIN
SF14	----	Override CH1	0	<input type="checkbox"/> EIN

3. GV anwenden

Jetzt können wir in den Mixern für Querruder und Höheruder statt mit Festwerten mit einstellbaren
 variablen Werten arbeiten.

Gewichtung: GV4 50 bis 100% mit S2 Expo: GV3 0 bis 40% mit S1

CH1	Gas Gewichtung (+100%)
CH2	Que Gewichtung (GV4) Expo (GV3)
CH3	Höh Gewichtung (GV4) Expo (GV3)

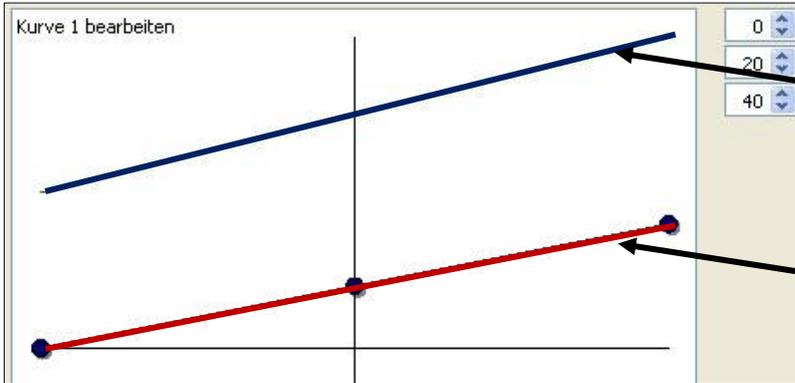
Alternative: S1 und S2 mit Kurven vorverarbeiten

S1 und S2 nicht per Vorbereitung selber direkt in den Bereichen einschränken, sondern S1 mit Kurve1 von 0% bis 40% und S2 mit Kurve2 50% bis 100% verarbeiten.

S1, S2 liefern ganz normal -100% bis +100%.

das geht bei den Mischerberechnungen Ch14, CH15 durch die Kurven und es kommt raus:

Ch14 0% bis 40% (S1= Expowerte) und CH15 +50% bis +100% (S2= Gewichtung)



S2 mit Kurve 2
50% bis 100% in CH15 als GV4
für Gewichtung

S1 mit Kurve 1
0% bis 40% in CH14
als GV3 für Expo

-100% 0% +100% S1, S2-Potis von -100% +100%

1. vorberechnen: S1, S2 mit den Kurven in freien Mischern CH14, Ch15 verarbeiten

CH13		
CH14	S1 Gewichtung (+100%)	Kurve (1)
CH15	S2 Gewichtung (+100%)	Kurve (2)
CH16		

2. GV zuweisen: GV3 aus Ch14 von S1 GV4 aus Ch15 von S2.

SF14	EIN	Adjust GV 4	Quelle	CH15	<input checked="" type="checkbox"/> EIN
SF15	EIN	Adjust GV 3	Quelle	CH14	<input checked="" type="checkbox"/> EIN
SF16	----	Override CH1	0		<input type="checkbox"/> EIN

3. GV verarbeiten: GV3 als Expo von S1 GV4 als Gewichtung von S2

CH1	Gas Gewichtung (+100%)
CH2	Que Gewichtung (GV4) Expo (GV3)
CH3	Höh Gewichtung (GV4) Expo (GV3)
CH4	

→Das Ergebnis ist exakt das Gleiche!

Logische Schalter L1 ... L32

(10/13)

Achtung nicht verwirren lassen, je nach Softwarestand steht da:

Logische Schalter Lx, Programmierbare Schalter PSx oder Custom Switch CSx

LOGIK SCHALTER		Var2		10/12		
L1	Puls	SD↑	000-1.7]	---	2.6	N/U
L2	Takt	1.9	1.0	---	---	---
L3	SRFF	SA↑	SA↑	---	---	---
L4	SRFF	SB↑	SB↑	---	---	---
L5	---	---	0	---	---	---
L6	---	---	0	---	---	---
L7	---	---	0	---	---	---

CUSTOM SWITCHES		9/12	
CS1	v>ofs	Trnr1	06:24
CS2	v>ofs	Power	500W
CS3	v>ofs	Alt	400m
CS4	v1>v2	PPM8	3POS
CS5	v>ofs	Cnsp	900mA
CS6	v<ofs	Cell	3.34v
CS7	v>ofs	Curr	24.0A

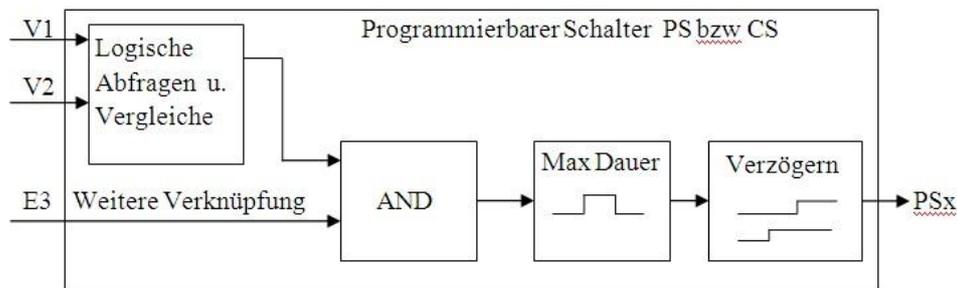
PROG. SCHALTER		Delay		10/13		
PS1	a~x	Que	20	SA↓	2.0	2.0
PS2	a>x	S1	15	---	---	---
PS3	a<x	S1	37	PS2	1.5	4.0
PS4	a<x	Sei	45	---	---	---
PS5	a>x	Sei	25	PS4	---	---
PS6	a~x	LS	-20	---	---	500
PS7	---	---	0	---	---	---

Logische Schalter **Lx**, Programmierbare Schalter **PS**, Custom Switches **CS**. Software Schalter, sind Schalter die durch Bedingungen und Verknüpfungen aktiviert werden und wie richtige Schalter verwendet werden können. Es gibt **32** Logische Schalter (**L1...L32**)
 Wenn sie aktiv werden, d.h. ihre Bedingungen erfüllt ist und „**ON**“ sind, erscheinen sie **Fett** dargestellt, so wie hier bei **L2, L3, (PS4, PS6, CS4, CS6)**. Dann können sie an vielen anderen Stellen Aktionen auslösen. Mischerzeilen aktivieren, Flugphasen umschalten, Spezialfunktinen starten
Logische. Schalter reagieren auf Bedignungen und können damit Aktionen auslösen!

Ein log. Schalter verhält sich wie ein normaler physikalischer Schalter

Ein log. Schalter als Mischerquelle liefert automatisch -100% oder 100%

Ein log. Schalter als Mischerschalter aktiviert, deaktiviert eine Inputzeile / Mischerzeile



Bedeutung der 6 Eingabe-Spalten:

Zuerst legen wir die Bedingungen fest, dann die Vergleichswerte, dann weitere Verknüpfungen und Zeiten

Spalte 1 die Bedingungen:

Es gibt 4 Arten von Bedingungen

- Vergleich von 1 Variable **a** mit Festwert **x** **a=x, a~x, a>x, a<x, |a|>x, |a|<x,**
- Vergleich von 2 Variablen **a** und **b** **a=b, a!=b a=>b, a=<b, a>b, a<b**
- Vergleich von Differenzwert **d** mit Festwert **x** **d >= x, |d| >= x**
- Logischer Verknüpfungen mit 2 Variablen : **AND, OR, XOR**

Das Tilde-Zeichen ~ a-x bedeutet ungefähr, circa, mit ca 5% Hysterese a-x Que 30
 damit kann ein Wert in einem schmalen Bereich sicher abgefragt werden.
 Geber- oder Potistellungen könnten sonst nur per Bereichsabfrage erfasst werden.
 Mit einer circa „~“ Abfrage geht das viel einfacher.
 Soll nur der Betrag, absolut (ohne Vorzeichen) berücksichtigt werden dann gilt
 "| a |" bzw. "| d |" (aus -10 wird dann 10, d.h. immer nur positive Werte)

Dann gibt es noch 4 fertige Funktionen:

- Progr. Taktgenerator: **Takt** Takt 0.3 1.2 0,3s ON, 1,2s OFF Taktzeit damit 1.5s
- Einstellbaren Einzelimpuls: **Puls** mit div. Startbedingungen, Verzögerung und Dauer
- Ein Set/Reset Flip-Flop: **SRFF** mit div. Startbedingungen, Verzögerung und Dauer
- Eine Bereichsabfrage: ~~**Range** mit Untergrenze und Obergrenze (kommt noch)~~
- Modulo Divison** **Mod** der Rest wird ermittelt 100 Mod 15= 6 **Rest 10**

Spalte 2 und 3 die eigentlichen Vergleichswerte, Variablen oder Festwerte

Die Variable a b kann alles sein: Sticks, Potis, PPMx Input, ein Ausgangskanal (CHxx), Timer (TMR1, TMR2) oder ein Telemetrie Eingangswert

Der Festwert x ist ein Zahlenwert der mit der Variablen a verglichen wird.

Beispiele:

- L1 a> x S1 10** L1 wird aktiv wenn der Potiwert S1>10 ist
- L2 | a |> x S1 10** L2 wird aktiv wenn der Potiwert S1 größer +10 oder kleiner als -10 ist (wegen den Betragsstrichen!)
- L3 d> x S1 10** L3 wird aktiv wenn die Differenz zu S1 >10 ist
- L4 | d |> x S1 10** L4 wird aktiv wenn die absolute Differenz S1 > 10 ist

Spalte 4 enthält Freigabeschalter bzw eine weitere UND Verknüpfung

In der Spalte 4 gibt es noch eine weitere **UND/AND** Verknüpfung **zu anderen (prog.) Schaltern** um prog. Schalter selber wieder untereinander verknüpfen zu können. Damit lassen sich alle Arten von Freigaben/Sperren und Bereichs-Abfragen und Bereichs Fensterungen machen.



Beispiel Bereichsabfrage

- L2 a>x S1 15**
- L3 a<x S1 37 UND L2 (AND Switch)**

Das bedeutet: L3 wird aktiv wenn S1 im Bereich von 15 bis 37 liegt

Spalte 5 und 6 sind einstellbare Zeitdauern und Verzögerungen

Spalte 5 Einschaltdauer für die dieser log. Schalter aktiv ist, auch wenn die Startbedingung schon wieder weg ist, läuft diese Zeit ab wenn sie mal gestartet wurde. Das ist wie eine Impulsverlängerungszeit

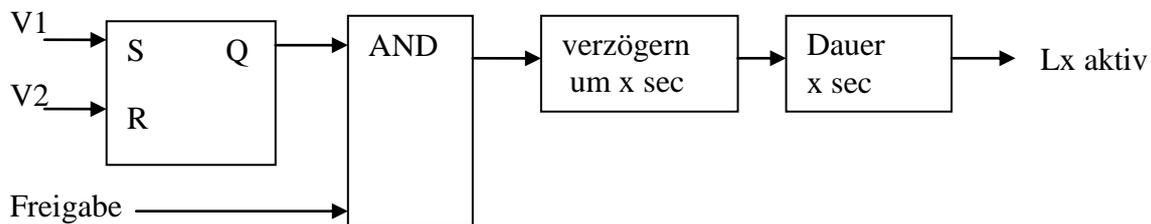
Spalte 6 Verzögerung bis dieser log. Schalter aktiv wird. Während der Verzögerung muss die Startbedingung gültig bleiben, sonst kommt kein nachfolgendes EIN-Signal.

Stehen hier keine Werte so ist keine Zeitüberwachung aktiv!
d.h. die Funktion ist solange aktiv, solange die Bedingungen selbst erfüllt sind.

**Achtung nicht verwirren lassen, je nach Softwarestand steht da:
Logische Schalter LS, Programmierbare Schalter PS oder Custom Switch CS**

Neue Funktionen für Logische Schalter ab opentx2.0

SRFF Ein SR-FlipFlop mit Bedingungen



SRFF ist eine neue universelle Flip Flop Funktion, die mehr kann als die bisherige Toggle-Funktion mit dem vorangestellten „t“ und ersetzt ihn.

Das Flip-Flop wird durch einen kurzen Impuls gesetzt und durch einen anderen kurzen Impuls wieder rückgesetzt. V1= Setz, V2= Rücksetz (Reset)

Das Flip-Flop kann noch durch ein Freigabesignal gesperrt/freigegeben werden
Der Ausgang Lx ist so lange aktiv bis das FlipFlop einen Resetimpuls erhält, oder die Freigabe weggenommen wird (führt auch zu einem Reset).

Falls eine Verzögerung und/oder Dauer eingegeben werden folgt:

Der Ausgang Lx kann um bis zu 15 Sekunden verzögert werden bis er aktiv wird

Die Dauer kann auf bis 15 s Sekunden eingestellt werden,

Ist die Dauer abgelaufen wird das Flip- Flop automatisch zurückgesetzt (Vorrang)

Setzen und Rücksetzen kann auch mit dem gleichen Impulsgeber erfolgen
damit haben wir ein T-Flip-Flop (Toggle Flip-Flop)

Als Impulseingabe und Freigabe können alle Arten von Schaltern und Schalterstellungen verwendet werden. (Physikalischer Schalter, Logische Schalter auch mit 3 Stellungen).

Liegen Setz- und Rücksetz gleichzeitig an hat Rücksetz den Vorrang.

L2 SRFF SB↓ SC↓

L2 wird durch SB gesetzt und SC rückgesetzt

L3 SRFF SH↓ Dauer 5s

L3 wird durch SH gesetzt und nach 5s autom. rückgesetzt

L4 SRFF SH↓ Dauer 3s Verzög 2s

L4 wird durch SH gesetzt, muss aber min 2s anstehen, ist dann 3s an und wird dann autom. Rückgesetzt

Ersatz der „t“ toggle Funktion: Erfolgt jetzt über einen Log.Schalter

L1 SRFF SA↓ SA↓

L1 wird durch SA gesetzt und rückgesetzt **Toggle-Funktion**

Puls Einen einmaligen Impuls erzeugen (Edge), wie ein Monoflop

Ein einmaliger Impuls kann erzeugt werden mit der Funktion Puls (Edge)
 Das ersetzt z. B. die Short und Long Funktion des SH-Tasters
 Am Beispiel Log. Schalter LS1:

L1 Puls [0,0 : 0,7] SA↓ Dauer 5,0

Schalter SA wird für max 0,7s betätigt, dann wird ein einmaliger Impuls für 5 s Dauer erzeugt.

L2 Puls [1,0 : 1,0] SH↓ Taster SH muss min 1 sec betätigt sein

L3 Puls [1,0 : 2,5] SH↓ Taster SH muss zwischen 1 und 2,5sec betätigt sein

L4 Puls [0,0 : 0,6] SH↓ Taster SH darf nur max 0,6 sec betätigt sein

L5 Puls [2,0 : 0,0] SH↓ da kommt gar nichts raus!

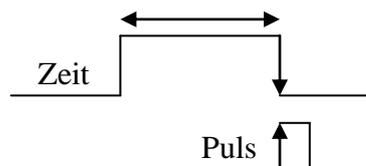
Wird keine Dauer angegeben erfolgt nur ein sehr kurzer Impuls (1Rechenzyklus ca 10-20ms)

SH↓ SH↓s long und short ersetzen:

L5 Puls [0,0 : 0,4] SH↓ das ersetzt den SH↓s short mit max 0,4s

L6 Puls [0,8 : 0,8] SH↓ das ersetzt den SH↓l long mit min 0,8s

Der Puls wird bei fallender Flanke ausgelöst,
 denn die Zeitbedingung muss erst abgelaufen
 und erfüllt sein.



Takt Ein einstellbarer Taktgenerator

Ein Taktgenerator mit einstellbarem ON- und OFF-Zeiten **Takt** (bzw TIM)

L3 SB↓ Takt 0,5 0,2 einstellbares Taktverhältnis 0,5 s Ein + 0,2s Aus = 0,7s Periode

Range Einen Analogwert als Bereich abfragen (kommt erst noch)

Ein Analogwert kann in einem Bereich abgefragt werden

L2 S2 Range -35 +45

Bereichsabfrage (Alternative zu Range) kann man aber auch selber machen

L2 a>x S1 -25

L3 a<x S1 40 UND L2

Somit: L3 wird aktiv im Bereich von -25 bis 40

Modulo (ab openTx V2.1)

Mod Restwert einer Division, gut um bestimmte Werte (Telemetrie) abzufragen

L2 Mod S1 S2

$$100 \text{ Mod } 25 = 0 \quad (4 \cdot 25 = 100 + \text{Rest } 0)$$

$$100 \text{ Mod } 15 = 10 \quad (6 \cdot 15 = 90 + \text{Rest } 10)$$

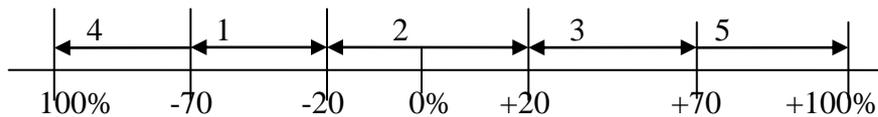
$$-100 \text{ Mod } 15 = 5 \quad (-7 \cdot 15 = -105 + \text{Rest } 5)$$

$$-100 \text{ Mod } -15 = -10 \quad (6 \cdot -15 = -90 + \text{Rest } -10)$$

Beispiel: Fensterbereiche, Range-Bereich definieren (hier per Gasstellung Throttle)

Hier werden fast alle Möglichkeiten einer Bereichsabfrage/Fensterung an 5 Beispielen erklärt:

Bereich 1:	$-70 < a < -20$	L1	$a < x$	Thr - 20	L2	$a > x$	Thr -70	
Bereich 2:	$-20 < a < +20$	L3	$ a < x$	Thr +20				wg Symmetrie zu Null
Bereich 3:	$+20 < a < +70$	L4	$a > x$	Thr +20	L5	$a < x$	Thr +70	
Bereich 4:	$a < -70$	L6	$a < x$	Thr - 70				alles unterhalb -70
Bereich 5:	$a > +70$	L7	$a > x$	Thr +70				alles oberhalb +70



Entweder: Mit zusätzlichen Lx Verknüpfung und Abfragen in Spalte 2, 3

L8 L1 **AND** L2 genau im Fenster Bereich 1 **AND** in Spalte 2 L8 wird aktiv

L9 L4 **AND** L5 genau im Fenster Bereich 3 **AND** in Spalte 3 L9 wird aktiv

Oder: Mit direkter **AND** Verknüpfung in Spalte 4

L2 **AND** L1 L2 wird aktiv wenn L1 und L2

L5 **AND** L4 L5 wird aktiv wenn L4 und L5

L2 wg Symmetrie zu Nullpunkt als Betrag möglich

L10 **NOT** L2 **NOT** in Spalte 2, alles außerhalb von Fenster Bereich 2

L10 $|a| > x$ Thr +20 eine Alternative für alles außerhalb Bereich 2

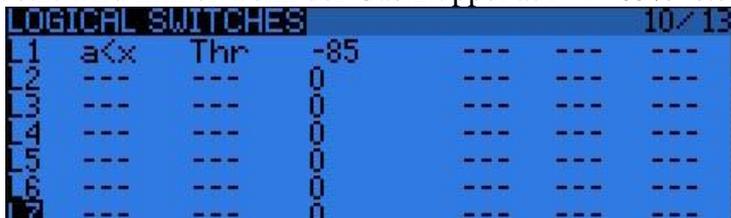
Beispiel: Glühkerzenheizung automatisch aktiveren wenn Gas fast auf Leerlauf

Eine Glühkerzenheizung soll immer dann automatisch angehen wenn der Gasknüppel fast ganz unten ist (bei < -85%).

Das geht mit einem logischen Schalter so:

L1 a<x Thr -85 und schon ist der log. Schalter **L1** definiert.

Jetzt wird immer wenn der Gasknüppel auf < -85% steht L1 aktiv,



Vergleiche a<x Gasknüppelwert Gas mit Festwert -85%

Jetzt können wir **L1** im Mixer Menü verwenden.



Dazu belegen wir den Kanal der die Glühkerzenheizung einschaltet z.B. CH8 und tragen dort ein, als Quelle „MAX“ (ist ein Festwert +100%) und als Schalter **L1**

In Mixer für Kanal 8 wird der Ausgang auf 100% (MAX) gesetzt wenn **L1** aktiv wird und damit die Glühkerzenheizung eingeschaltet.



Ganz einfach.

Übersicht der Variablen für Logische Schalter

Variable	Bedeutung	Wertebereich
Rud	Value of rudder channel	-125 to +125
Ele	Value of elevator channel	-125 to +125
Thr	Value of throttle channel	-125 to +125
Ail	Value of Aileron channel	-125 to +125
P1	Value of Pot 1 (top left)	-125 to +125
P2	Value of Pot 2 (top right)	-125 to +125
P3	Value of Pot 3 (front left)	-125 to +125
Rea	Rotary Encoder A (if installed) Dimmed if not available	-100 to +100
Reb	Rotary Encoder B (if installed)	-100 to +100
TrmR	Rudder trim switch value	-100 to +100
TrmE	Elevator trim switch value	-100 to +100
TrmT	Throttle trim switch value	-100 to +100
TrmA	Aileron trim switch value	-100 to +100
MAX	Constant maximum value determined by weight	-125 to +125
3POS	3 position switch. End points determined by weight setting	-Weight or 0 or +Weight
CYC1	Hubschrauber Taumelscheiben Mischwerte	
CYC2	Hubschrauber Taumelscheiben Mischwerte	
CYC3	Hubschrauber Taumelscheiben Mischwerte	
PPM 1 - 8	Eingänge am DSC Stecker	
Ch 1 - 32	Value of Channel 1 - 32	-125 to +125
Timer 1	Timer 1	Measured in seconds
Timer 2	Timer 2	Measured in seconds
TX	Transmitter RSSI	between 0 and 100
RX	Receiver RSSI	between 0 and 100
A1	Analog port1 on Frsky receivers	
A2	Analog port2 on Frsky receivers	
Alt	Altitude from FrSky altitude sensor	Metric or imperial based on
Rpm	RPM optical Frsky sensor	Set number of prop bladed on
Fuel	FrSky Fuel Sensor	Percentage
T1	GPS Sat Nrs. // Temperature 1 from Frsky temp sensor 1	??
T2	GPS Fix-Typ //Temperature 2 from Frsky temperature 2	??
Speed	Speed From Frsky GPS	Metric or imperial based on
Dist	Distance from origin From Frsky GPS	Metric or imperial based on
GPS Alt	Altitude From Frsky GPS	Metric or imperial based on
Cell	Lowest Cell on FLVS	volts
Cels	Sum of all cells on FLVS	volts
Vfas	Voltage detected by FAS100 or FAS40	volts
Curr	Amperage FAS or analog configured on telemetry page	mA
CNsp	Total mAh used	mAh
Powr	Power, voltage source used is configured on telemetry page, current as above	Watts

Liste ist nicht vollständig!

Spezial Funktionen SF (11/13)



Mit **[Ent Long]** ein Untermenü öffnen

Hier kann man Reaktionen, Funktionen und Abläufe starten, Ansagen aufrufen, Werte ansagen die dann ausgeführt werden wenn ein beliebiger Schalter (physikalisch oder logisch) aktiv wird. z.B. wird der Schalter **SE↑** aktiviert, dann starten die Variotöne. Oder ein Sicherheitsschalter der den Gas Kanal sperrt, damit nicht aus Versehen der Elektromotor anläuft, einen Timer reseten oder für die Trainer-Funktion Kanäle sperrt oder freigibt.

Mit Sicherheitsschaltern kann man eine höhere Stufe der Sicherheit einbauen und verhindern, dass etwas ungewollt anläuft oder sich bewegt.

Alle Typen von Schaltern als Bedingungen sind möglich

1. Alle physischen Schalter (**SA-SG**) in allen Varianten und Stellungen
Auch invertiert, also zum Beispiel "Schalter SB nicht in Mittelstellung" (**!SB-**)
2. Taster **SH** (beide Positionen)
3. Logische Schalter **L1-L32** EIN (+100%) oder AUS (-100%).
4. Alle acht Trimmwähler (**tSl, tSr, tHd, tHu, tGd, tGu, tQl, tQr**) (als links, rechts up,down)
5. Flugphasen (**FP0-FP8**)
6. Beim Aktivieren eines Models, z.B. für die Modellansage (**ONE**)
7. Einmal ansagen, aber nicht beim aktivieren des Modell (**!ONE**)
8. Immer aktiv (**EIN**)

Vordefinierte Funktionen:

1. **Sicher bzw Override CH1 .. CH32** Kanäle mit einem Sicherheitsschalter freigeben.
Ein Wert (-100 bis +100) kann übergeben werden und eine **ON/OFF** Checkbox erscheint wenn man Werte verändert.
2. **Trainer** alle 4 Kanäle zusammen übergeben oder
3. **Trainer** jeden Kanal einzeln (**Rud / Ele / Thr / Ail**) übergeben
4. **Instant Trim** Nette Funktion um das Modell ganz schnell zu trimmen. Es werden bei Betätigen des Schalters die Knüppelstellungen und die Trimmwerte von Ele, Ail, Rud, (nicht aber die Werte von Thr/Gas) als aktuelle Trimmwerte in die Subtrim/Offset von Limits7/12 übernommen. Dann Knüppel loslassen und damit ist das Model fertig getrimmt. Falls der Bereich von + -25% nicht ausreichen kann man mir extended Trim die Werte auf + -100% erweitern, aber dann ist eh was faul am Flieger.
5. **Spiel Töne, Sag Text, Sag Wert** einen Sound abspielen oder einen Werte ansagen
6. **Vario** für ein Variometer das Audiosignal freischalten

openTx für Taranis Anleitung Deutsch

7. **Reset**. Je nachdem Timer1, Timer2, Telemetrie. (Telemetrie) oder Alles.
8. **Haptic** Vibrator Alarm
9. **Beep** Einen kurzen Piepser auslösen
10. **Backlight** Hintergrundbeleuchtung. Ein/Aus
11. **Adjust GV1 - Adjust GV9**. Mit Adjust GVx werden den globalen Variablen Werte zugewiesen und können eingestellt werden (Beispiel: SC↑ Adjust GV3 P3)
Das können sein: Festwerte, beliebige Analogwerte, beliebige Kanäle,
andere GVARS und +1 / -1 increment/decrement
Mit [**Enter Long**] umschalten von Zahlen nach Variablen!
12. **Start Log** alle Flugdaten auf SD Karte als Log-Datei im CSV-Format aufnehmen.
Datei steht im Verzeichnis /LOGS mit Datum und Uhrzeit
13. **Screenshot** des LCD-Schirms im BMP-Format im Verzeichnis /SCREENSHOTS

Und es gibt noch viele weitere vorbelegte Funktionen

Telemetrie und Werte ansagen
Werte und Texte ansagen
Hintergrundmusik abspielen/stoppen
Timer reseten, stoppen, voreinstellen
Div. Töne und Warnungen ausgeben
Variotöne ausgeben

Unter companionV2 sieht man die Auswahlliste

Tip:

Mit dem Freigabe-Häckchen kann man auf einfache Weise eine Spezialfunktion sperren oder freigeben ohne sie löschen zu müssen.

Ganz rechts Wiederholraten (5 = alle 5s wird die Ansage wiederholt)

Beispiel: Telemetrie Grenzwerte setzen und Warntöne erzeugen

Dazu muss man 2 Dinge tun:

1. Mit einem **Logischen Schalter Lx** die **Aktion definieren**
(d.h. unter welchen Bedingungen soll der Schalter aktiv werden)
2. Mit einer **Spezialfunktionen Sfx** die **Reaktion auslösen** (d.h. was soll dann passieren).

Am Beispiel: via Telemetrie wird der Strom übertragen und im Sender daraus die verbrauchte Akkukapazität Cnsp berechnet.

Wenn mehr als 1200mAh verbraucht sind soll ein Warnton kommen.

Die Aktion: **L4** a>x Cnsp 1200mAh (**L4** wird bei >1200mAh aktiv)

Die Reaktion: **SF1 L4** Play Sound Warn1 (wenn **L4** aktiv ist löst **SF1** den Ton aus)

Beispiel: Variometer umschalten und Telemetrie-Daten alle 0,1s aufzeichnen

Schalter	Funktion	Parameter	aktivieren
CF1	SD- Vario		
CF2	SD↓ Play Value	Alt+ 10s	
CF3	SF↓ Start Log	0,1	<input checked="" type="checkbox"/> ON

Beispiel: Ansagen, Werte und Töne in den Spezialfunktionen auslösen

Alles was mit Ansagen, Werte ansagen, Tönen zu tun hat wird in den Spezialfunktionen ausgelöst, aufgerufen, programmiert.

Sag Text: Eine Wav-Datei wird aufgerufen, Verzeichnis **Sounds/de**

Spiel Töne: Diverse Töne und Geräusche, die sind fertig vorprogrammiert

Sag Wert: Der Wert einer Variablen, Geberstellungen, Telemetriewerte, usw. wird angesagt

Es können Wiederholraten in Sekunden eingegeben werden, 1 - 99s

1x nur einmal wenn ausgelöst

!1x nur einmal wenn ausgelöst, aber nicht schon beim Modellsaufruf

Einfaches Beispiel mit dem Schalter SB

Bitte damit etwas spielen und ändern, Handbuch nutzen!

SPEZIAL-FUNKTIONEN				11/13
SF1	SB↑	Sag Text	attero	9s
SF2	SB-	Spiel Töne	Piep3	3s
SF3	SB↓	Sag Wert	Gas	5s
SF4	---			
SF5	---			
SF6	---			
SF7	---			

Dann gibt es noch Ansagen und Töne die automatisch kommen.
z.B.. Akuuspannungswarnungen, Schalterstellungswarnungen, Countdown usw.
Das stellt man in Grundeinstellungen und den Modelleinstellungen ein.

Das Dateiverzeichnis auf der SD-Karte muss exakt eingestellt sein **Sounds/de/System**
Die Dateinamen dürfen nur 7-8 Zeichen lang sein

Unter **Sounds /de** kann man die eigenen Dateien ablegen, mit eigenem Inhalt und eigenem Namen.
Bitte nur hier die eigenen Ansagen reinstellen!

Unter **Sounds/de/System** sind die Systemmeldungen, Name darf nicht umbenannt werden, Inhalt schon

Das kann man sich auch alles einfach am Sender anhören, Datei auswählen und **[Enter]**

Eigene Ansagen erzeugen ist auch sehr leicht , das kann man online machen,
dann Datei downloaden , umbenennen und dann ins Verzeichnis **Sounds/de** reinkopieren.

Hier: <http://212.59.78.77/taranis-sounds-beta1/index.php>

Tip: Mit den Kommas ,, kann man etwas Zeit zwischen den Worten einfügen

LUA Custom Scripts Interpreter Sprache (12/13)

LUA ist eine Interpreter-Programmiersprache die im Sender selbst kleine Programme starten und ausführen kann. Dazu wurde LUA 5.2 um openTx-Funktionen erweitert die senderspezifisch sind. Diese kleinen Programme sind Text-Dateien die in ganz bestimmten Bereichen auf der SD-Karte stehen müssen, damit sie ausgeführt werden.

Grundsätzlich gibt es:

Programme die nur einmal ausgeführt werden.

Programme die neue Modelle halbautomatisch erzeugen, mit Kontextmenüführung

Programme die dauernd ausgeführt werden, zyklisch alle ca 20-30ms

Programme die Telemetriewerte bearbeiten und am Bildschirm darstellen (bis zu 7 Bildschirme)

Programme für Spezialfunktionen

Einzige Begrenzung ist das RAM des Prozessors

Auf der SD-Karte müssen diese Unterverzeichnisse für LUA Scripts anlegen

/SCRIPTS/

/SCRIPTS/BMP/

Alle Bilder die für LUA verwendet werden

/SCRIPTS/WIZARD/

Alle LUA Scripte+ Bilder für Modellgenerator, neue Modelle erzeug.

/SCRIPTS/TEMPLATES/

/SCRIPTS/MIXES/

/SCRIPTS/FUNCTIONS

für die Spezial Funktionen

/SCRIPTS/<MODELNAME>/telemXX.lua für Telemetrianzeigen.

/SCRIPTS/TELEMETRY für Telemetriescripte (ab openTxV2.10)

Das LUA-System muss man zur Zeit noch von Hand einrichten, damit es läuft.

Mehr info zu LUA : <http://www.open-tx.org/lua-instructions.htm>

LUA- Modellgenerator zum download: <http://www.open-tx.org/2014/06/02/lua-wizard/>

Das **wizard.zip** ins Unterverzeichnis /SCRIPTS/WIZARD reinkopieren, dort entpacken

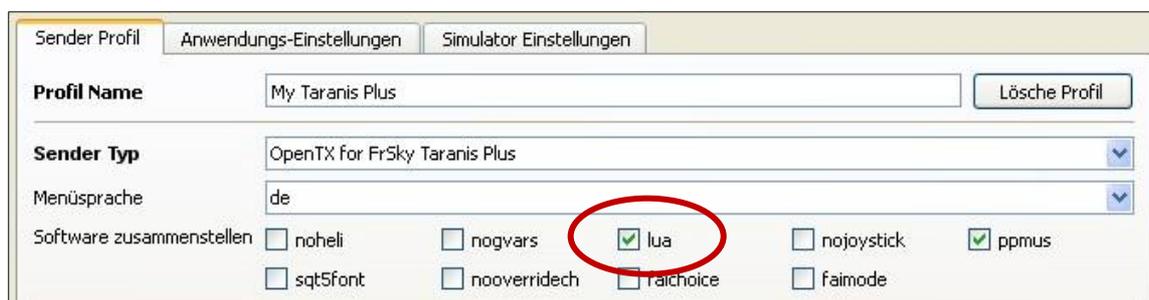
Dann startet der Modellgenerator automatisch wenn ein neues Modell angelegt wird.

Bisher gibt es nur LUA Scripte für Modelle neu anlegen, ist aber noch nicht vollständig.



Beispiel: LUA auf dem Sender und am PC einrichten

Damit LUA-Scripte laufen können, muss der Interpreter in openTx eingebunden werden
LUA Interpreter unter Companion auswählen, dann Sender-Firmware downloaden und flashen

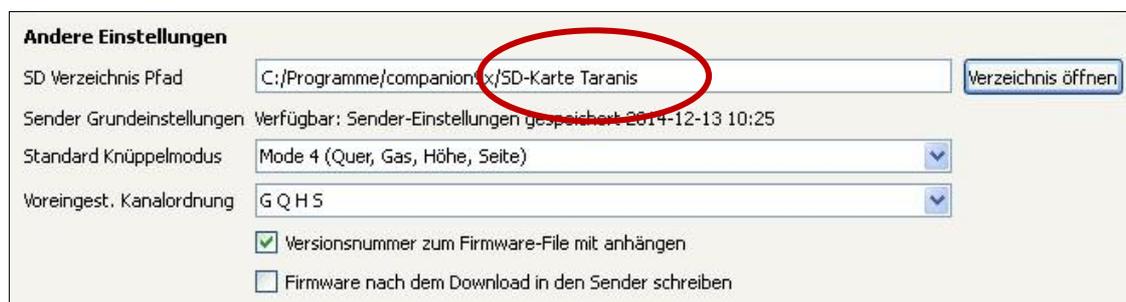


1. Auf der SD-Karte müssen die LUA-Verzeichnisse exakt so heißen wie oben dargestellt, nicht anders, oder irgendwie so ähnlich, oder so wie es mir gerade gefällt. So ist es halt mal.
OpenTx muss das ganze Zeug finden, zusammenstellen und zuordnen können.

2. Am PC muss im Senderprofil ein Pfad zu einem Verzeichnis eingerichtet sein auf dem ein Abbild der SD-Karte des Senders liegt.

Konkret: wir kopieren dann auf den PC in dieses Verzeichnis die SD-Karte rein, denn nur dann kann Companion und der Simulator dort auch zugreifen.

Wir arbeiten nicht mit einem angeschlossenen Sender!



3. Als <Modellname> eben nicht "Modellname" sondern den Name des Modells und zwar ganz genauso wie er am LCD dargestellt und abgespeichert ist.

z.B. ASW21, Extra_300, Tornado, MODELL1, Fun_Flyer, MODELL9 usw.

Häufiger Fehler: MODELL7 statt MODEL7 oder Modell 7 statt Model_7

Der "Modellname" darf nur max 12 Zeichen lang sein!

Keine Leerzeichen, Keine Sonderzeichen, sonst eben den Namen abändern!

4. Die Namen für Telemetrie LUA-Scripte müssen so heißen:

telem1.lua

telem2.lua

telem3.lua

usw. bis telem7.lua

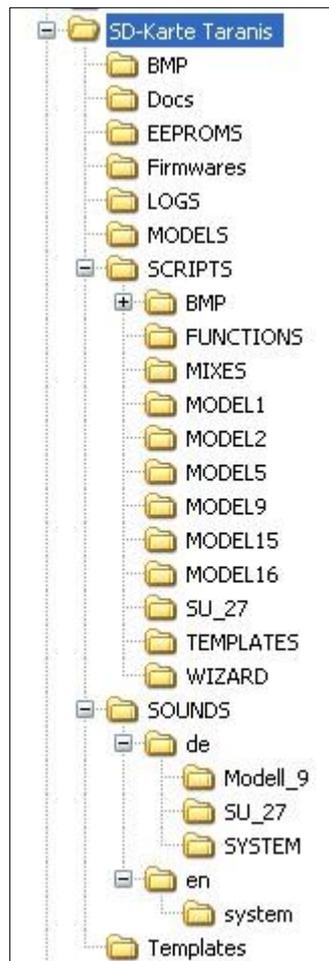
Beispiel: /SCRIPTS/Fun_Flyer/telem1.lua

In dieser Reihenfolge werden sie dann mit **[PAGE Long]** auch **automatisch** aufgerufen!

Die komplette Verzeichnisstruktur der SD Karte als Kopie am PC

Die normalen Funktionen der SD-Karte

Modelle, Firmware, Startbilder, Modellbilder
Logs, EEPROMs



Alles für die LUA-Scripte

Bilder für die LUA-Scripte
Modelle mit LUA- Telemetrie-Scripte
Wizard für neue Modelle erzeugen

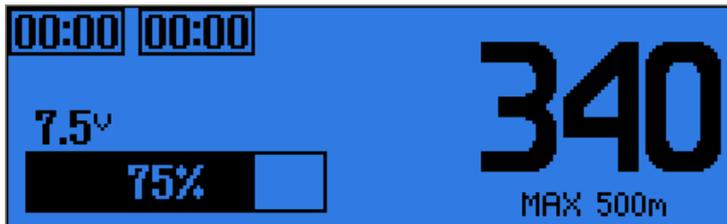
Die Sounds in Deutsch und Englisch

fixe Sounds freie Ansagen, Zahlen

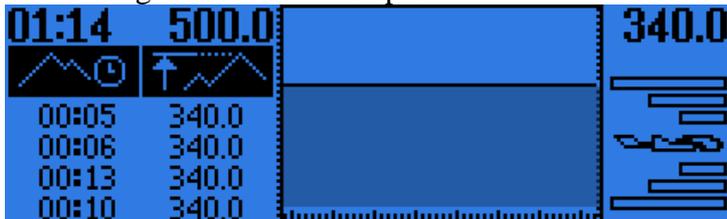
Beispiel: Ein sehr kurzen LUA-Script: telem2.lua <http://www.open-tx.org/downloads.html>

```
Test002.lua x de.h.txt x telem1.lua x telem2.lua x
1 local function run(event)
2   lcd.drawNumber(210, 10, getValue("altitude"), XXLsize)
3   lcd.drawText(150, 54, "MAX", 0)
4   lcd.drawChannel(172, 54, "altitude-max", LEFT)
5   local timer = model.getTimer(0)
6   lcd.drawTimer(2, 1, timer.value, MIDSIZED)
7   lcd.drawRectangle(0, 0, 34, 14)
8   timer = model.getTimer(1)
9   lcd.drawTimer(40, 1, timer.value, MIDSIZED)
10  lcd.drawRectangle(38, 0, 34, 14)
11  lcd.drawChannel(11, 29, "tx-voltage", LEFT+MIDSIZED)
12  local settings = getGeneralSettings()
13  local percent = (getValue("tx-voltage")-settings.battMin) * 100
14  lcd.drawNumber(35, 45, percent, LEFT+MIDSIZED)
15  lcd.drawText(lcd.getLastPos(), 45, "%", MIDSIZED)
16  lcd.drawGauge(5, 42, 88, 18, percent, 100)
17 end
18
19 return { run=run }
```

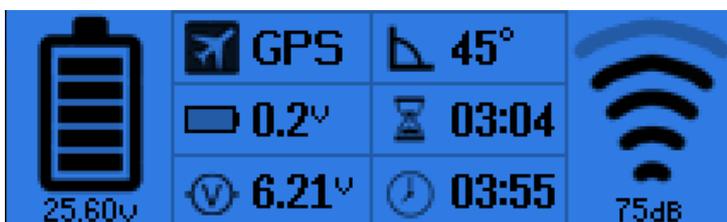
telem2.lua



Ein etwas größeres LUA Beispiel



Ein anderes umfangreiches LUA Beispiel:



Wo bekommt man fertige LUA Scripte her:

Dutzende fertige LUA-Scripte zum Testen und Downloaden

rcsettings: <http://rcsettings.com/index.php/view...13-lua-scripts>

rcgroups LUA-page: <http://www.rcgroups.com/forums/showt...180477&page=76>

fpv-community: <http://fpv-community.de/showthread.php?47985-LUA-scripts-zum-testen>

Einfache Anleitung: <http://open-txu.org/home/grad-school...ations-of-lua/>

OpenTx LUA Programmiersprache Referenz Guide

<http://www.open-tx.org/lua-instructions.html>

<http://www.open-tx.org/documents.html>

OpenTx LUA Feldbeschreibungen

http://jenkins.open-tx.org/nightly-20/11_15_2014/lua_fields.txt

Kleines Entwicklungssystem für LUA

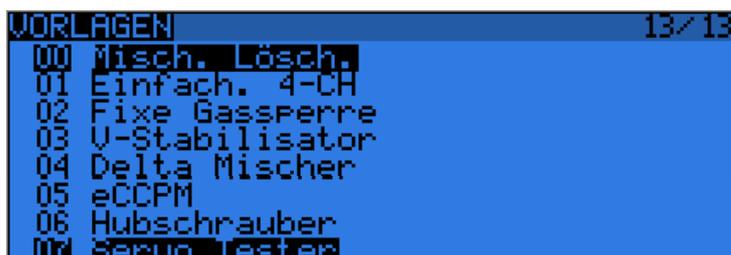
Das ZeroBraneStudio ist kostenlos, mit einem zusätzlichen OpenTx Emulator

Ein echtes eigenständiges LUA Entwicklungssystem mit zusätzlichem openTx Emulator

<http://www.rcgroups.com/forums/showt...180477&page=73>

Fertige Voreinstellungen, Templates, Th9x, 9XR, 9XRPro(13/13)

Achtung: Für Taranis openTx2.0 siehe Teil E, Templates, Wizzard, LUA-Scripte



Das sind fertige Voreinstellungen für bestimmte Modellarten, die man aus der Liste auswählen kann. Mit den Cursor auswählen und dann mit [MENU LONG] bestätigen.

Dann werden diese fertigen Mischer-Funktionen **im aktuellen aktiven** Modell eingefügt.

Die Zuordnung der Kanäle und Mischer erfolgt in der Reihenfolge wie sie in den Sendergrundeinstellungen 1/6, Kanalzuordnungen (RX Channel Order) festgelegt wurde.

z.B. **GQHS** (TAER)

Ganz oben in der Liste gibt es die Funktion: Clear Mixer. Mit [MENU LONG] werden dann alle Mischerwerte für das **aktuelle, aktive** Modell gelöscht.

Folgende fertige Voreinstellungen gibt es:

1. **Simple 4-CH**: ein einfaches 4 Kanal Flugmodell.
2. **T-Cut**: Damit wird ein Gas Sicherheitsschalter dazugemischt. Das ist etwas aufwändig programmiert, da die Gas-Leerlaufstellung **und** eine Schalterstellung überwacht werden.
3. **V-Tail**: Mischer für ein V-Leitwerk.
4. **Elevon / Delta**: Delta-Mischer für Höhenruder und Querruder gemischt.
5. **eCCPM**: Allgemeiner einfacher Heli-Mischer für elektrisches collectives Pitch eCCPM mit 3 Servos
6. **Heli Setup**: Erweiterte Mischer für eCCPM, resetet die Mischer und Kurven des einfachen eCCPM
7. **Servo Test**: Erzeugt auf Kanal 32 ein Servotestsignal das langsam von -100% auf +100% und zurück läuft und per PS1 aktiviert wird. Das kann man dann z.B. auf Empfänger Kanal8 legen und ein Servo anschließen. (Source für Mischer CH8 ist CH32)

Sehr viele weitere Templates findet man im 9xforums hier: <http://9xforums.com/forum/> und unter: <http://rcsettings.com/>

Dort gibt es sehr viele Hubschraubereinstellungen, Quadrocopter, Segler, Spezialfunktionen für Flächenmodelle, Doorsequenzer, Spezialfahrzeuge, alles Mögliche an Klappensteuerungen usw.

9xforums ist das zentrale Forum für die Th9x, 9XR und Taranis und deren Softwarevarianten.

Telemetrie-einstellungen (13/13)

Telemetriedaten können angezeigt werden. Dazu hat der Sender schon ein telemetriefähiges XJT Sendemodul oder es kann zusätzlich ein externes Sendemodul XJT oder DJT verwendet werden. Es muss ein telemetriefähiger Empfänger, **X-Typen im X16, D16-Mode** z.B. X8R, X6R, X4R oder **D-Typen im D8 Mode** D8R-II verwendet werden, der die Daten sendet.

Für die Telemetrieübertragung mit allen Smart-Port Sensoren muss das XJT-HF Modul im D16 (X16) Modus betrieben werden. Der D8 Modus ist für die D-Empfänger und Hub-Sensoren

Die FrSky-Sensoren gibt es (noch) in 2 Ausführungen

- für den **bisherigen (alten) Frsky-Sensor Hub** (alle Sensoren werden zentral angeschlossen)
- für die **neue Smart-Port** Schnittstelle (alle Sensoren werden in Reihe hintereinander angeschlossen)



Bisheriger (alter) Fr-Sky Sensor-Hub

hier werden die Sensoren zentral angeschlossen

Eine super Seite: http://www.eflightwiki.com/eflightwiki/index.php?title=FrSky_Telemetry

Telemetriedaten parametrisieren für die Anzeige am Sender

Diese Telemetrie-Darstellungen gelten bis openTx V2.017

Ab openTx V2.10 gibt es ein stark erweitertes Verfahren und viel Mehrmöglichkeiten!

Das ist nur mal ein Auszug der Möglichkeiten Telemetriedaten zu konfigurieren.

```
TELEMETRIE 12/13
A1 Kanal 0.77v
Skala 13.20v
Offset 0.00v
Low Alarm 9.31v
Kritisch Alarm 8.90v
A2 Kanal 3.68v
Skala 10.00v
```

```
TELEMETRIE 12/13
Variometer
Quelle Alti
Limite -10 -0.5 0.5 10
Bild 1 Wert
Batt Uhr1 SWR
RSSI A1 A2
Höhe Umdr Stof
```

```
TELEMETRIE 12/13
A2 Kanal 3.68v
Skala 10.00v
Offset 2.50v
Low Alarm 2.50v
Kritisch Alarm 2.50v
RSSI
Low Alarm 500
```

```
TELEMETRIE 12/13
Bild 1 Wert
Batt Uhr1 SWR
RSSI A1 A2
Höhe Umdr Stof
T1 T2 Gesc
Bild 2 Wert
Höh GHöh Zell
```

```
TELEMETRIE 12/13
RSSI
Low Alarm 41
Kritisch Alarm 39
Daten
Blätter 2
Spann. FAS
Strom 2.18
```

```
TELEMETRIE 12/13
Bild 2 Balken
RSSI 35 48
Strm 0.0A 25.0A
Verb 0mAh 250mAh
Bild 3 Werte
```

```
TELEMETRIE 12/13
Verb Leis BesX
BesY BesZ Rich
Bild 3 Balken
Verb 0mAh 5100mAh
Rich 0° 360°
UGes 0.00 2.55
Höh- -500m 16200m
```

```
TELEMETRIE 12/13
Daten
Blätter 2
Spann. A1
Strom A2
Variometer
Quelle Alti
Limite 100 -0.5 0.5 10
```

Hier werden alle Einstellungen für die Telemetriedaten vom FrSky-Modul angepasst, normiert und die Alarme gesetzt. Der FrSky-Empfänger haben div. Eingänge um Signale zu verarbeiten und als Telemetriedaten an den Sender zu übertragen. 2 Analoge Eingänge A1, A2, einen serielle Eingang für einen Telemetrie-Hub oder S-Port und interne Temperatur- und Spannungsmessungen

Analoge Eingänge A1 A4 Bereiche anpassen je nach Empfänger

Selbst wenn man keine Telemtriesensoren angeschlossen hat, **RSSI** und **A1** werden immer übertragen und können zur Anzeige gebracht werden.

Für A1, A2 gilt, der Messbereich im Empfänger intern ist immer 3,3V und wird via Spannungsteiler 1:4 auf 13,2V angepasst, dazu gibt es fertige einstellbare Schaltungen.

Nicht alle Empfänger haben die 2 Analogeingänge A1 und A2

Der **X8R Empfänger** hat gar keinen Analogeingang herausgeführt. Er überträgt aber immer die

Empfängerakkuspannung (4-10V) als **A1** an den Sender. **Intern** hat der **X8R** einen festen 1:4 Teiler, somit ist er auf 13,2V eingestellt. Deshalb auch am Sender in der Telemetrie A1 auf 13,2V einstellen!

Für jeden Eingang A1 bis A4 kann man einstellen:

1. **Skala** : der Messbereich der (auch in der Balkenanzeige) angezeigt wird (0V bis 204V)
2. **Offset**: eine Verschiebung und Anpassung damit die richtigen Werte angezeigt werden
3. **Alarme**: die mit im FrSky-Modul abgespeichert werden
 - Alarm Level (---, **Gelb**, **Orange**, **Rot**) bzw. Orange=low , Rot=kritisch
 - Alarmrichtung, ob der Alarm bei > oder < Schwellwert kommt
 - Ansprechschwelle, Schwellwert

Wenn ein Wert verändert wird, wird er sofort ins FrSky Modul übertragen und abgespeichert.

Empfangsfeldstärke RSSI des Empfängers

Das gleiche Prinzip wird für die Empfangsfeldstärken angewendet:

- Ansprechschwellen, Schwellwerte Orange, low, auf ca 41dBm einstellen
- Alarmstufen (---, **Gelb**, **Orange**, **Rot**) Rot, kritisch, auf ca 39dBm einstellen

Wenn ein Wert verändert wird, wird er sofort ins FrSky Modul übertragen und abgespeichert. Ab ca 28dBm treten dann Ausfälle auf und das Failsafe greift ein.

Sendeantenne-Funktionsüberwachung SWR

Werte: 0-2 alles ok, Antenne mit der Hand ganz umfasst 6 -18 **ab > 51 Alarm Antenne defekt!**

Das Format des seriellen Empfangsprotokoll (UsrData):

1. **Proto**: verwendetes serielles Protokoll vom Telemetrie Empfänger. Die Optionen sind:
None, keines, nicht verwendet, falls Smart-Port-Sensoren verwendet werden
Hub für das FrSky-Hub Modul oder
WSHHigh für das Winged Shadow How High (gibt sehr genau die Höhe an)
2. **Blades**: Anzahl der Propellerblätter die am Drehzahlmesser angezeigt werden(2-3-4-5-6-Blatt)

Konfiguration der Balkenanzeigen für Telemetrie:



Es können bis zu 4 Anzegebalken (**Bars**) pro Bildschirm dargestellt werden:

Dazu braucht es 3 Parameter:

1. **Source**: die Datenquelle, was soll angezeigt werden
2. **Min**: den Minimalwert links
3. **Max**: den Maximalwert rechts

openTx für Taranis Anleitung Deutsch

Das Erreichen von Grenzwerte (z.B. Höhen, Spannungen, Drehzahlen usw.) kann automatisch angezeigt werden, wenn sie von Alarmen des FrSky Telemetrie-Modul kommen oder von den virtuellen Schaltern. So kann man einen virtuellen Schalter auf z.B. 400m Höhen (Altitude) setzen und bei Erreichen von 400m wird eine Meldung/Ton/Ansagetext erzeugt.

→XJT-HF-Modul im D16 (X16) Modus betreiben!

Für die Telemetrieübertragung mit allen Smart-Port Sensoren muss das XJT-HF Modul im D16 (X16) Modus betrieben werden.

Dazu sind auch die X-Empfänger nötig X8R, X6R, X4R

Der D8 Modus des XJT-HF-Moduls ist für die D-Empfänger D8R-II, D8R-II Plus, D8R-II XP und die alten Hub-Sensoren wg der Kompatibilität noch vorhanden.

Beispiel: Telemetrie-Grenzwerte setzen und Warnton auslösen

Dazu muss man 2 Dinge tun:

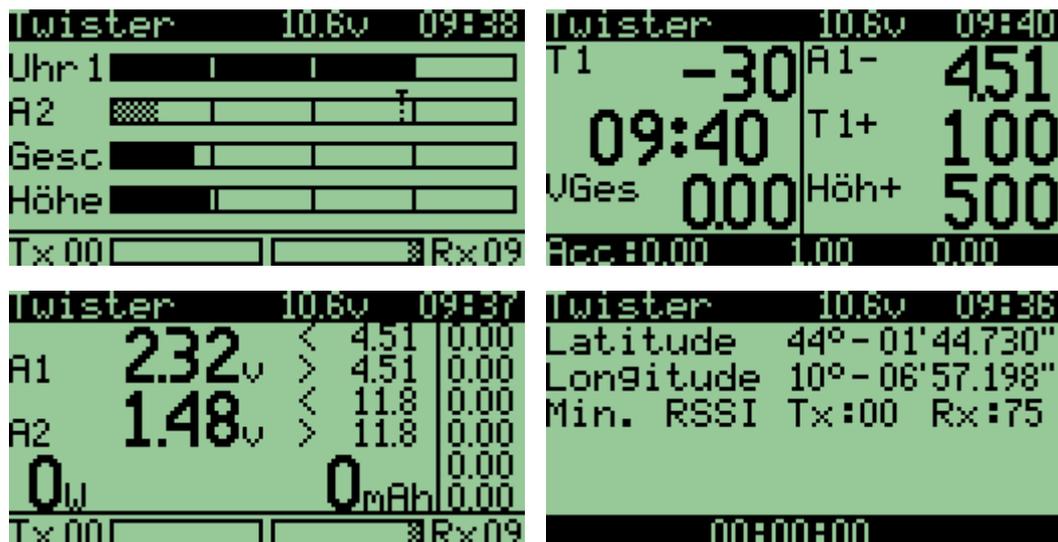
1. Mit einem **Logischen Schalter Lx** die Aktion definieren (d.h. wann soll der Schalter aktiv werden)
2. Mit einer **Spezialfunktionen SFx** die Reaktion auslösen. (d.h. was soll dann passieren)

Am Beispiel: via Telemetrie wird die verbrauchte Akkukapazität Cnsp übertragen. Wenn mehr als 1200mAh verbraucht sind soll ein Warnton kommen.

Die Aktion: **L4** a>x Cnsp 1200mAh (**L4** wird bei >1200mAh aktiv)

Die Reaktion: **SF1 PS4** Play Sound Warn1 (wenn **L4** aktiv ist löst **SF1** den Ton aus)

Anzeige der Telemetriedaten am Sender je nach Einstellungen



Die Telemetrie Anzeigen werden mit [PAGE LONG] aus dem Hauptmenu aufgerufen. Die Anzeigen sind abhängig von den Daten die man empfängt und konfiguriert hat. Von Screen zu Screen kommt man mit [PAGE] Mit [ENTER Long] erscheint eine Auswahlmenü dort kann man Telemetriedaten reseten. Mit [EXIT] kommt man wieder ins Hauptmenu des Senders.

Balkenanzeigen mit Schwellwerten, Ansprechschwellen



Eingänge A1 und A2 mit Min, Max Max, und LiPo-Zellen



Höhenmesser, Geschwindigkeit, Temperaturen ...

```
Twister 10.8V 00:00  
RPM:0 Fuel:75%  
T°1:0° T°2:0°  
Alt:500m  
Acc:0.00 0.00 0.00
```

GPS Daten

```
Twister 10.8V 00:00  
Lat:44°01.7455-  
Lon:10°06.9533-  
Alt:0m Dst:0m  
Spd:0kts Max:0kts  
2000-00-00 00:00:00
```

Hier werden Längengrade, Breitengrade, Höhe und Distanzen angezeigt.
Nach dem reseten der Telemetriedaten werden die ersten empfangenen GPS-Daten als Startwert für alle weiteren Berechnungen verwendet.

Telemetrie Alarme, Warnungen und Ansagen

Es gibt 3 Arten von Alarmierungen:

1. **Alarme** aus dem Frsky HF-Telemetrie-Modul (Gelb/Orange/Rot - 1/2/3 beeps)
2. **Warnungen** das sind System-Alarme/Warnungen für alle Arten von Parameter
3. **Ansagetexte** und **Töne** z.B. vom Variometer Sensor

Alarme vom Frsky-Modul (DJT, XJT)

Das sind Eingangssignale die von den A1/A2/ RSSI Signalen des Empfängers kommen und im FrSky Modul ausgewertet werden. Sie werden durch Ansprechschellen die im Frsky Sender-Modul hinterlegt sind ausgelöst. Wenn aber keine Telemetriedaten mehr vom Empfänger ankommen, kommt auch kein Alarm! Oder anders ausgedrückt, wenn vorher Daten da waren, werden die letzten Werte für Alarm oder kein Alarm verwendet.

Im openTx Telemetrie Screen kann man diese Art von Alarm einstellen, Gelb, Orange, Rot und die Schwellwerte dazu. Die Schwellwerte werden als kleine senkrechte Pfeile in den Balkenanzeigen, **aber nur für A1/A2 /RSSI**, angezeigt. Werte unterhalb werden als gepunkteter Balken dargestellt (siehe unten).

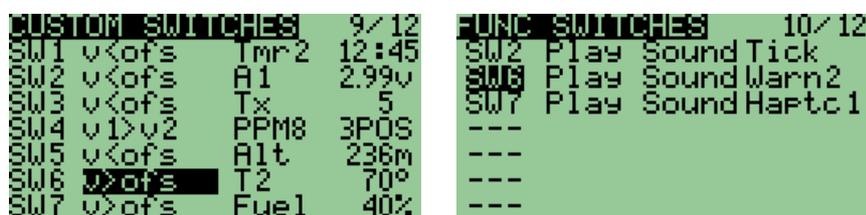


Die Balkenanzeigen für die Temperaturen 1 und 2 werden unterhalb der Schwelle fett dargestellt, da der Wert noch nicht überschritten ist. Sie werden erst oberhalb gepunktet.

Warnungen

Das sind System-Alarme die von den Funktions- Schaltern (Spezial-Funktionen) 10/12 ausgelöst werden. Die Ansprechschwellen dazu werden in den Programmierbaren Schaltern (Custom Switch) 9/10 eingestellt.

Diese Art von System-Alarmen kann für alle möglichen Ereignisse programmiert werden. (Frsky HUB Sensor, Timer, PPM, Stick, Kanal Werte und auch für A1/A2 und RSSI).



Variometer einstellen

OpenTx kann auch für verschieden Variometer Ansagen und Töne ausgeben um Thermik zu finden. Es werden 4 Hersteller von Variometer unterstützt.

- Das Thermal Scout Produkt von Winged Shadow <http://www.wingedshadow.com/>
- Das normale FrSky Variometer am FrSky Hub <http://www.frsky-rc.com/>
- Das Halcyon Project von eine Forumsmitglied <http://code.google.com/p/halcyon/>
- **Das openXvario Projekt mit Arduino** <http://code.google.com/p/openxvario/>

Die Konfiguration des Variometers geschieht wie folgt:

Im Telemetrie Screen mit den Cursor nach unten und „Vario“ auswählen.

Dann braucht man die Signalquelle wo das Variometer angeschlossen ist.

Zur Auswahl hat man **VSpd, BaroV2, A1, A2**,

VSpd für das Frsky Variometer für vertikale Geschwindigkeit, steigen/sinken

BaroV2 für das Halcyon systems

A1/A2 für das Thermal Scout System, openXvario je nach Eingang A1 oder A2.

Anmerkung: Wenn man A1/A2 verwendet muss man auch die Alarme für die A1/A2 Kanäle im Telemetrie Screen freigeben.

Der Rangebereich ist hier auf 3.2m/s und einen Offset von -1,6m/s eingestellt. Das wurde gemacht weil das Messsystem bei 1,6V = kein Steigen (Null) und bei 3.2V starkes Steigen liefert.

Es ist nicht nötig Alarme (**Gelb, Orange, Rot**) für Kanäle A1/ A2 zu definieren



Wenn man das Vario einstellt gibt es noch 2 Optionen von Grenzwerten „Limits“ einzustellen. Das ist etwas trickreich damit das Vario genau richtig auf Thermik reagiert und „Nullschieber“ ausblendet.

Die zwei Grenzwerte sind:

1. Minimale negative Sinkrate, damit das Vario mit einem Ton/Ansage beginnt.
OFF – kein Signalton für negative Sinkrate, Einstellbereich von -10.0 ... 0.0
2. Minimale positive Steigrade damit das Vario mit einem Ton/Ansage beginnt.
-1.0 ..2.0 diese -1.0 scheint etwas komisch, aber ein Beispiel macht das schnell klar:

Wenn man -0.7 im zweiten Limitfeld für die positive Seigrate einträgt so bedeutet das:

Ein normaler Segler hat eine Sinkrate von -1.0m/s und hat jetzt nur noch eine Sinkrate von -0.7m/s . Er hat also eine leichte Thermik von +0,3m/s gefunden, denn seine Sinkrate ist jetzt kleiner geworden, er sinkt noch, aber viel weniger als vorher.

Bei einem winged shadow system sind guten Anfangswerte für Range und Offset 10.16m/s und -5.08

Bei Taranis gibt es ein eingebautes Soundsystem für Variometertöne das in Tonfrequenz und Tondauer Sinken, Nullschieberbereich, Steigen signalisiert.

→ Siehe Sender Grundeinstellungen



Zum Schluss muss man das Vario noch freigeben/sperrern, damit es Sound oder Ansagen machen kann.

Das machen wir ganz einfach in den Spezial Funktionen 11/13 in dem wir mit einen Schalter z.B.

SA↓ Vario =Töne und **SA→ Sag Wert** für Ansage Höhe

(nicht verwechseln mit Höh= Knüppel) aktivieren.

Und mit Schalter **SB↓** kann man auch noch die Aufzeichnung auf die SD-Karte starten.



Beispiel: Frsky Variometer am X8R anschließen und Bereiche einstellen

Das Frsky Vario wird am **S-Port** (nicht verwechseln mit **S-Bus**!) angeschlossen und liefert die Höhen (Alt) und Steig- und Sinkraten (Vertical Speed, Vspd).

Immer das FrSky Vario High Precision mit der neuesten Software verwenden!

Um das Vario einzustellen sind 3 Schritte nötig

1. Vario Töne einstellen

Im Sender, Grundeinstellungen, wird der Tonbereich und Wiederholrate der VarioTöne eingestellt
zB. 500Hz 1200 Hz 800ms

```
SENDER GRUNDEINSTELLUNGEN 1/8
---Variometer---
Lautstärke          == ==
Pitch at Zero       700Hz
Pitch at Max        1700Hz
Repeat at Zero      500ms
---Alarm wenn---
Akku Spg kleiner   6.5v
```

2. Vario Bereiche einstellen

Im Modell, Telemetrie, werden 3 Bereiche für Sinken, Nullschieber, Steigen definiert.

- Sinken: mit Dauerton
- Nullschieber: mit konstantem Kurzen Pieps
- Steigen: mit veränderlichem Piepston

Sink Min / Climb Max gute Werte +/-3m/s

Der mittlere Bereich, Sink Min/ Climb Min, liefert einen kurzen Piepsston

Da werden die „Nullschieber-Werte“ z.B. von -0,5 -0,1 eingestellt,

(Ab openTx V2.10 kann man auch schon hier die Mitte ausblenden, ruhig, ohne Töne)

```
TELEMETRIE 12/13
Daten
Blätter      2
Spann.       A1
Strom        ---
Variometer
Quelle        Vario
Limite       -3 -0.2 0.2
```

Achtung: wg. Rauschsignalen um die 0,0 immer etwas unterhalb von 0,0 bleiben.

Damit hat man 2 Dinge: Einen ruhigen gleichmäßigen Ton für den Nullschieber und eine sehr schnelle Reaktion auf kleinste Thermik.



Gute Einstellungen sind zB.

- 3 -0,5 -0,1 +3
- 3 -0,2 -0,1 +3

Weniger gut sind Bereiche die 0,0 einschließen

- 3 -0,2 +0,1 +3

3. Vario freigeben

Das Vario muss in den Spezialfunktionen auch noch freigegeben werden (siehe unten, Funktion Vario)
Das kann man dann auch gleich umschalten von Ton auf Ansage.

3a. Vario umschalten von Ton auf Ansage, Daten aufzeichnen, Daten Loggerfunktionen

Mit einem 3 Stufen-Schalter kann man die Variotöne und Höhenansagen umschalten/wegschalten
Höhenansagen (alle 10 s) **SD↓**, Varitöne **SD--**, oder ganz weg wenn **SD↑**
Mit **SF↓** kann man die Telemetrie-Daten auf die SD-Karte aufzeichnen, Auflösung 0,1s



Tip:

Bestimmte Bereiche/ Töne automatisch ausblenden:

Das Vario gibt in allen 3 Bereichen **immer** unterschiedliche Töne ab.

Will man einen Bereich haben in dem **automatisch** keine Töne kommen, z.B. -0,1m bis + 0,1m so kann man diesen Bereich mit Log.Schaltern ausblenden und in den Spez. Funktionen Play Vario freigeben.

Logische Schalter: **L1 a>x Vario +0,1**
L2 a<x Vario -0,1
L3 OR L1 L2 AND SD— blendet den Bereich um +/-0,1 aus

Spez.Funktionen: **SF1 L3 Play Vario**

mit SD— in den log Schaltern wird L3 freigegeben und damit in den SF das Play Vario.

Das ist die flexibelste Art für jeden Benutzer.

Der eine will keinen Ton beim Nullschieber, der ander keinen Ton beim Sinken oder erst ab einem bestimmten Bereich usw.

Praktische Erweiterung:

Das Vario soll ganz aus sein, wenn der Motor läuft.

Ein Log. Schalter fragt den Gas Kanal (hier Kanal 1) auf ca. Nullstellung ab,
das wird noch verknüpft mit L3 von oben und damit erst das Vario automatisch freigegeben.

Log. Schalter: **L1 a>x Vario +0,1**
L2 a<x Vario - 0,1
L3 OR L1 L2 blendet den Bereich um +/-0,1 aus
L4 a<x CH 1 -98% AND L3 überwacht auf Motor Aus und Bereich

Spez. Funktionen: **SF1 L4 Play Vario** gibt die Variotöne frei

Das Vario gibt also nur Töne aus, wenn der Motor aus ist und Vspeed ausserhalb von +/-0,1 ist.
Das könnte man auch noch mit einem Schalter (wie oben) verknüpfen um
Von Ton auf Ansage umzuschalten.

Stromsensor / Spannungssensor einstellen

FrSky Stromsensoren gibt es mit (alter) Hub-Schnittstelle und mit neuer S-Port-Schnittstelle. Dann gibt es Stromsensoren von Fremdherstellern, die Ihre Daten an A1 und A2 liefern.

Mit einem Stromsensor kann man den aktuellen Stromverbrauch (A) und die Akkuspannung (V) messen und damit die aktuelle Leistung (W) und die verbrauchte Kapazität (mAh) ermitteln.

Leistung und Verbrauch errechnet die Taranis intern. Deshalb muss man unter **Daten berechnen aus:** die **richtige Sensorquelle** hier **FAS** für Strom und Spannung angeben sonst wundert man sich wenn falsche Werte errechnet werden.



Es gibt 2 Möglichkeiten um Stromsensoren anzuschließen:

1. **FrSky FAS-40 bzw 100** der 40A bzw 100A Stromsensor der am **S-Port** des Empfänger oder am **FrSky-Hub** angeschlossen wird.
Aber immer **Sensorquelle FAS einstellen!**

2. **Externer Sensor** am A1/A2 Eingang des Empfänger

~~Für beide Arten von Sensor muss man im openTx Telemetrie Screen die **UsrData** einstellen.~~

1. **Proto** auf **None** oder **Hub** abhängig davon ob man A1, A2 für Spannungseingänge verwendet und den Stromsensor **FAS** am FrSky Hub anschließt.
2. **Voltage** auf **A1, A2, FAS** oder **Cel** abhängig davon wo der Spannungssensor angeschlossen ist. A1/A2 ist ein externer Spannungssensoren der direkt am entsprechenden Eingang angeschlossen ist. FAS ist der Spannungssensor am FAS-40 und Cel ist die Spannungsmessung für den Akku-Zellensensor FAS-01
3. **Current** an **A1, A2, FAS** abhängig davon wo der Stromsensor angeschlossen ist. A1/A2 ist ein externer Stromsensor der direkt am entsprechenden Eingang angeschlossen ist. FAS ist der Stromsensor FAS-40 der am Hub angeschlossen ist.
4. **FAS Offset** ist ein Korrekturwert für die Strommessung, um genauere Werte zu werden. Wenn z.B in Ruhe bereits 100mA fließen.

FAS-40 und FAS-100 Stromsensoren liefern Strom und Spannung

Für die FAS-40 und FAS-100 Stromsensoren muss man **FAS** als Telemetrie-Quelle einzutragen.

S-Port Stromsensor



Hub-Stromsensor



Externe Spannungs- und Stromsensoren an A1 und A2

Wenn man einen externen /fremden Spannungs- oder Stromsensor verwendet braucht man zusätzlich zu den UserData Einstellungen auch jeweils die Einstellungen für die A1 und A2 Kanäle. Einen Messbereich (Range) und einen Offsetwert für Spannung und Strom .

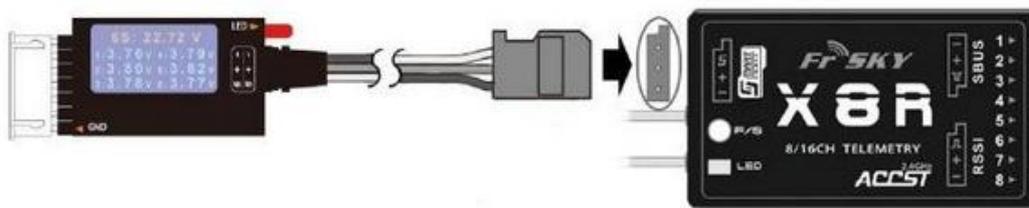
```
TELEMETRY 11/12 TELEMETRY 11/12
Alarm Red < 50 Range 12.6v
UserData Offset 4.24v
Proto None Alarm --- < 4.24v
Blades 2 Alarm --- < 4.24v
Voltage A1 A2 channel 0.00A
Current A2 Range 5000A
Vario Offset 0.00A
```

Spannungssensor FLVSS mit Smart-Port Anschluss

Mit farbigem OLED Display für die Anzeige von bis zu 6 Zellen
 Gesamtspannung: Zellen und Einzelspannungen: Zelle



Smart-Port Anschluss



Telemetrie-einstellungen am Sender für FLVSS und FAS

Spannungsquelle: FVLSS bzw Zellen (Cells)

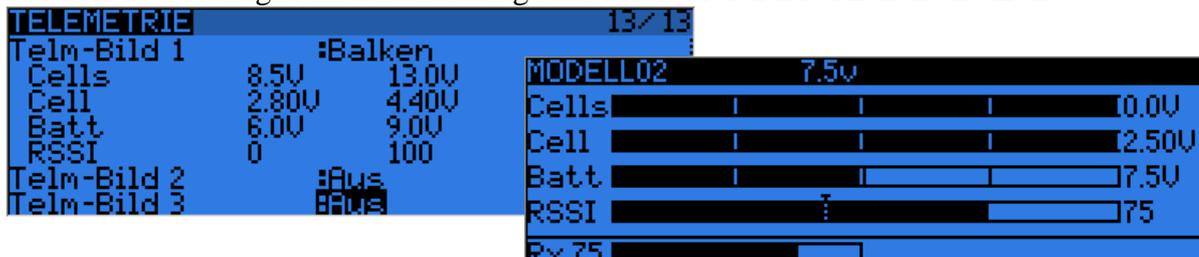
Stromquelle: FAS falls gleichzeitig noch ein FAS40 Stromsensor angeschlossen ist



Anzeige am Sender Telemetrie-Bildschirm (erfolgt dann automatisch, ohne weitere Einstellungen)



Oder Telemetrie-Einstellungen als Balkenanzeige mit Messbereichen für Zellen und Zelle



Kleinste Gesamtspannung: Zellen- (Cells-) Kleinste Einzelspannung: Zelle- (Cell-)

Übersicht der Telemetriewerte Stand: opentx V2.07

Deutsch	Englisch	Bedeutung	Sensor
TX-Akku	Batt	Spannung Sender Akku	Sender
Zeit	Time	aufgsummierte Zeit	
Stoppuhr 1	Timer1	Stoppuhr 1	
Stoppuhr 2	Timer2	Stoppuhr 2	
SWR	SWR	Sender HF-Abstrahlung, Überwachung der Antenne	
RSSI TX	RSSI TX		
RSSI RX	RSSI RX	Empfänger Signal Feldstärke	Empfänger
A1	A1	Analogwert einstellbar	X8R
A2	A2	Analogwert einstellbar	
A3	A3	Analogwert einstellbar	
A4	A4	Analogwert einstellbar	
Alt	Alt	Höhe aus dem Vario-Sensor	Vario
RPM	RPM	Drehzahl	
Fuel	Fuel	Füllstand	
T1	T1	GPS Satelitten Anzahl	
T2	T2	GPS Fix Typ 0,2,3,4	
Geschw.	Speed	Geschwindigkeit	Vario
Dist	Dist	Abstand	
GPS Alt	GPS Alt	GPS Höhe	
Zelle	Cell	Akku Einzelzellenspannung	FVLSS
Zellen	Cells	Akku Gesamtspannung	FVLSS
Vfas	Vfas	Spannung vom Sensor FAS	FAS
Strom	Current	Strom vom Sensor FAS	FAS
Verbrauch	Cnsp	Verbrauch in mAh (wird im Sender errechnet)	
Leistung	Powr	Leistung in Watt (wird im Sender errechnet)	
AccX	AccX	Beschleunigungssensor in X-Achse	
AccY	AccY	Beschleunigungssensor in Y-Achse	
AccZ	AccZ	Beschleunigungssensor in Z-Achse	
HDG	HDG	Heading, Richtung in Grad 0-360°	
V-Speed	V-Speed	Vertikale Geschw Steigen/Sinken Variosensor	Vario
AirSpeed	AirSpeed	Geschwindigkeit aus Drucksensor	
dte	dte	Gesamtenergie Variosensor	
A1-	A1-	Kleinster Wert	
A2-	A2-	Kleinster Wert	
A3-	A3-	Kleinster Wert	
A4-	A4-	Kleinster Wert	
Alt-	Alt-	Kleinste Höhe	
Alt+	Alt+	Größte Höhe	
RPM+	RPM+	Größte Drehzahl	
T1+	T1+	Größte Temperatur	
T2+	T2+	Größte Temperatur	

openTx für Taranis Anleitung Deutsch

Geschw+	Speed+	Größte Geschwindigkeit	
Dist+	Dist+	Größte Distanz	
AirSpeed+	AirSpeed+	Größte Geschwindigkeit	
Zelle-	Cell-	Kleinste Zellenspannung	
Zellen-	Cells-	Kleinste Akkuspannung	
Vfas-	Vfas-	Kleinste Spannung	
Strom+	Curr+	Größter Strom	
Leistung+	Powr+	Größte Leistung	

Die Neue Telemetrie ab openTx V2.10

Überblick:

Ab OpenTx V2.10 ist die Telemetrie komplett überarbeitet und erweitert worden. Was zuerst etwas aufwändig erscheint ist der Preis für die Universalität.

Alles kann mit allem gemacht und an beliebigen Stellen wieder verwendet werden.

Das heist ich kann jeden Telemetriewert so verwenden die einen Knüppelwert oder Schalter, Abfragen und Auswertungen machen, Werte verechnen und in Inputs und Mischern anwenden.

Mit fertigen LUA-Scripten muss man aber sehr aufpassen!

Alle angeschlossenen **FrSky-Sensoren** werden **automatisch** mit ihrer ID erkannt und zugeordnet.

Dabei bleiben auch die bisherigen fertigen Telemetrie - Namen erhalten!

Damit sollte man mit fertigen fremden LUA Scripten keine Probleme haben.

Aber:

Man kann auch die Telem- Namen selber vergeben / ändern, max 4 Zeichen.

Dann wird aber ein fertiges fremdens LUA-Scripts diese Namen nicht kennen und kann damit nicht laufen.

Deshalb gibt es eine LUA-Abfragefunktion die erkennt ob der Name vorhanden ist und was er bedeutet. Somit muss man dies Abfrage an ein fertiges LAU-Script voranstellen.

-----INFO wg Scriptabfrage ab openTX V2.10-----

For anyone else looking into scripts on 2.1, Telemetry scripts SD card path has now changed (as of today's build), and is now: SCRIPTS\TELEMETRY

thanks to bsongis (Dev) It's now clear that if you're using a telemetry field in a script, you need to "catch" a non-existent field:

In 2.0, all fields were always available (ie, alt, vspd, a2, etc etc)

In 2.1, only the fields used are available, so if the field is not available,

your script won't run (rather than just show 0) here is the recommended catch,

Instead of just using this Code:

```
Alt = getFieldInfo("Alt").id
```

You should employ a function with a catch Code:

```
local function getTelemetryId(name)
    field = getFieldInfo(name)
    if field then
        return field.id
    else
        return -1
    end
end
```

You can then call this Code:

```
Alt = getTelemetryId("Alt")
```

Note that telemetry field names used are whatever you've set them up to on the sensors page.

Telemetriedaten

Dass Daten kommen erkennt man an einem Sternchen in der Sensorzeile.

Der aktuelle Sensorwert wird dort auch direkt angezeigt.

Fremdsensoren müssen eine 2 Byte Phys.-ID liefern und Daten-ID's wenn sie mehrere Werte liefern (z.B. SM-Unisens-E liefert bis zu 7 Werte)

```

TELEMETRIE 13/13
RSSI
Vor-Alarm bei 41
Kritisch-Alarm 38
---Sensoren--- Wert ID
1: RSSI [75] 25
2: SWR [5] 25
3: Temp [100°C] 5
    
```

```

TELEMETRIE 13/13
---Sensoren--- Wert ID
1: RSSI [75] 25
2: SWR [5] 25
3: Temp [100°C] 5
4: Temp [200°C] 5
5: Alt [10.00m] 1
6: Cels [8.20V] 2
    
```

```

TELEMETRIE 13/13
8: RPM [3600rpm] 5
9: Fuel [10.00mL] 11
10: LowZ --- 0
11: Kapa [8mAh]
Ein Sensor hinzufügen
---Variometer---
Quelle Alt
    
```

Im Telemetrie-Eingabe Screen kann man die Sensoren definieren und in allen Parametern anpassen. **Ein Name (4 Zeichen) muss vorhanden und eindeutig sein**, denn darauf kann beliebig zugegriffen werden.

Sensorwerte können mehrfach benutzt und unterschiedlich verarbeitet werden.

```

SENSOR12 0
Name
Type SensorTyp
ID 0000 0
Einheit -
Präzision 0.--
Gewichtung -
Offset 0
    
```

```

SENSOR5 10.00m
Präzision 0.00
Bereich -
Offset 0.00
Auto Offset 
Filter 
Permanent 
Log Daten 
    
```

Sensorwerte können untereinander verrechnet werden um weitere Werte zu erhalten.

Addieren, Multipliziert, Mittelwerte, Min, Max, Gesamt, höchster/ niedrigster Wert usw.

(Strom *Spannung = Leistung in W) (Strom als Zeitintegral = Verbrauch in mAh) usw.

```

SENSOR12 0
Name
Type Berechnung
Formel Addieren
Einheit -
Präzision 0.--
Quelle1 ---
Quelle2 ---
    
```

```

SENSOR12 0.00V
Quelle1 ---
Quelle2 ---
Quelle3 ---
Quelle4 ---
Auto Offset 
Filter 
Permanent speich. 
    
```

```

SENSOR10 4.10V
Name LowZ
Type Berechnung
Formel Zelle
Zellen Sensor Cels
Zellen Wert Niedrigst
Log Daten 
    
```

Sensorwerte können aufgezeichnet werden als Log Daten, permanent gespeichert, oder auch gefiltert werden. Die Präzision kann eingestellt werden 0 0.0 0.00

openTx für Taranis Anleitung Deutsch

In der Summe sind 32 Sensorberechnungszeilen möglich.

Jeder Wert kann einzeln in der Logdatei gespeichert, resettet werden oder/und bleibt permanent erhalten.

Auf jeden Sensorwert oder Berechnungswert (**der Name mit max 4 Zeichen muss eindeutig sein!**) kann in den logischen Schalter, Spezialfunktionen, Inputs zugegriffen, abgefragt und verwendet werden.

Auch gleiche Sensoren mit unterschiedlicher ID sind möglich.

Damit z.B. 2 Stk 6 Zellen Spannungswächter mit Einzel-Zellenüberwachung für 12 Zellen.

Die Telemetriescreens an Sender kann man wie bisher auch als Werte oder Balken frei zuordnen und dann per [PAGE Long] anzeigen.

Beispiel: Anzeige der Einzel-Zellenspannungen vom FLVSS-Sensor

Als Sensor wird der FLVSS für 6 Zellen verwendet. Er wird automatisch erkannt mit ID 2 und liefert erst mal als Grundwert die Gesamtspannung als Cels [8,2V]

Er liefert aber auch die möglichen 6 Einzelspannungen.

Die müssen wir aber erst aufrufen wenn wir sie auch verwenden wollen.

```
TELEMETRIE 13/13
RSSI
Vor-Alarm bei 45
Kritisch-Alarm 42
----Sensoren---- Wert ID
1: RSSI [75] 25
2: SWR [5] 25
3: Temp [100°C] 5
5: Alt [10.00m] 1
6: Cels [8.20V] 2
7: Curr [1.0A] 3
8: RPM [3600rpm] 5
9: Fuel [10.00ml] 11
```

Jetzt wollen wir auch die Einzelzellenspannungen sehen
Also müssen wir eine „berechneten“ Sensor hinzufügen

Ein Sensor hinzufügen

Name vergeben: Z1 nur 4 Zeichen möglich, muss eindeutig sein!

Typ: Berechnung

Formel: Zelle

von Sensor: Cels

Zellenwert: 1.Zelle

```
SENSOR10 0.00V
Name Z1
Type Berechnung
Formel Zelle
Zellen Sensor Cels
Zellen Wert 1. Zelle
Log Daten 
```

Und schon haben wir die 1.Zelle als Einzelwert.

Da kann man jetzt wiederholen für die 2. 3... 6. Zelle

```
10: Z1 [0.00V]
11: Z2 [0.00V]
12: Z3 [0.00V]
```

Zur Anzeige bringen wir die Zellenwerte in den Telemetriebildschirmen

Das kann dann sein: der **aktueller Wert** Cels, **größter Wert** Cels+, **kleinster Wert** Cels -
Analogo dazu für Z1 Max Z1+ oder Min Z1- Z2 als Max Z2+ oder Min Z2- usw.

```
TELEMETRIE 13/13
----Infozeile----
Spannungsquelle ---
Höhe ---
Telm-Bild 1 Werte
PWR PWR+ PWR-
Verb Verb+
Z1 Z2 23
MODELL01 7.5V Tmr1
PWR 0.0 PWR+ 0.0 PWR- 0.0
Verb 0 Verb+ 0
Z1 0.00 Z2 0.00 Z3 0.00
```

Beispiel: Verbrauchte Kapazität in mAh und Leistung in W ermitteln

Als Sensoren haben wir den Stromsensor FCS-40A mit ID 03 und den Zellsensor FLVSS mit ID 02 verbaut.

Die verbrauchte Kapazität in mAh erhalten wir über das Strom vom FCS-40 und integrieren.

Die Leistung per Multiplikation von 2 Werten $P = U \cdot I$ Spannung vom FLVSS, Strom vom FCS-40

Verbrauchte Kapazität ermitteln:

Ein Sensor hinzufügen

Name vergeben: Verb Muss eindeutig sein!

Typ: Berechnung:

Formel: Verbrauch

Stromsensor auswählen: Curr

```

SENSOR13      0mAh
Name          Verb
Type         Berechnung
Formel       Verbrauch
Strom Sensor Curr
Permanent    
Log Daten    
    
```

Leistung ermitteln $P = U \cdot I$

Ein Sensor hinzufügen

Name vergeben: PWR Muss eindeutig sein!

Typ: Berechnung

Formel: Multiplikation

Einheiten : W für Watt

Präzision: 0.0 für die Genauigkeit 0 0.0 0.00

Quelle1: Cels Sensor ist der FLVSS

Quelle2: Curr Sensor ist der FCS-40

```

SENSOR14      0.0W
Name          PWR
Type         Berechnung
Formel       Multiplizier
Einheit      W
Präzision    0.0
Quelle1      Cels
Quelle2      Curr
    
```

Und so sehen dann die zusätzlichen Sensoren aus

```

10: Z1      [0.00V]
11: Z2      [0.00V]
12: Z3      [0.00V]
13: Verb    [0mAh]
14: PWR     [0.0W]
    
```

Auch bei der Leistung kann man wieder auswählen

Aktueller Wert: PWR, Max Wert: PWR+ Min Wert: PWR-

```

Telem-Bild 1  #Werte
PWR          PWR+      PWR-
Verb         Verb+     ---
Z1           Z2       Z3
    
```

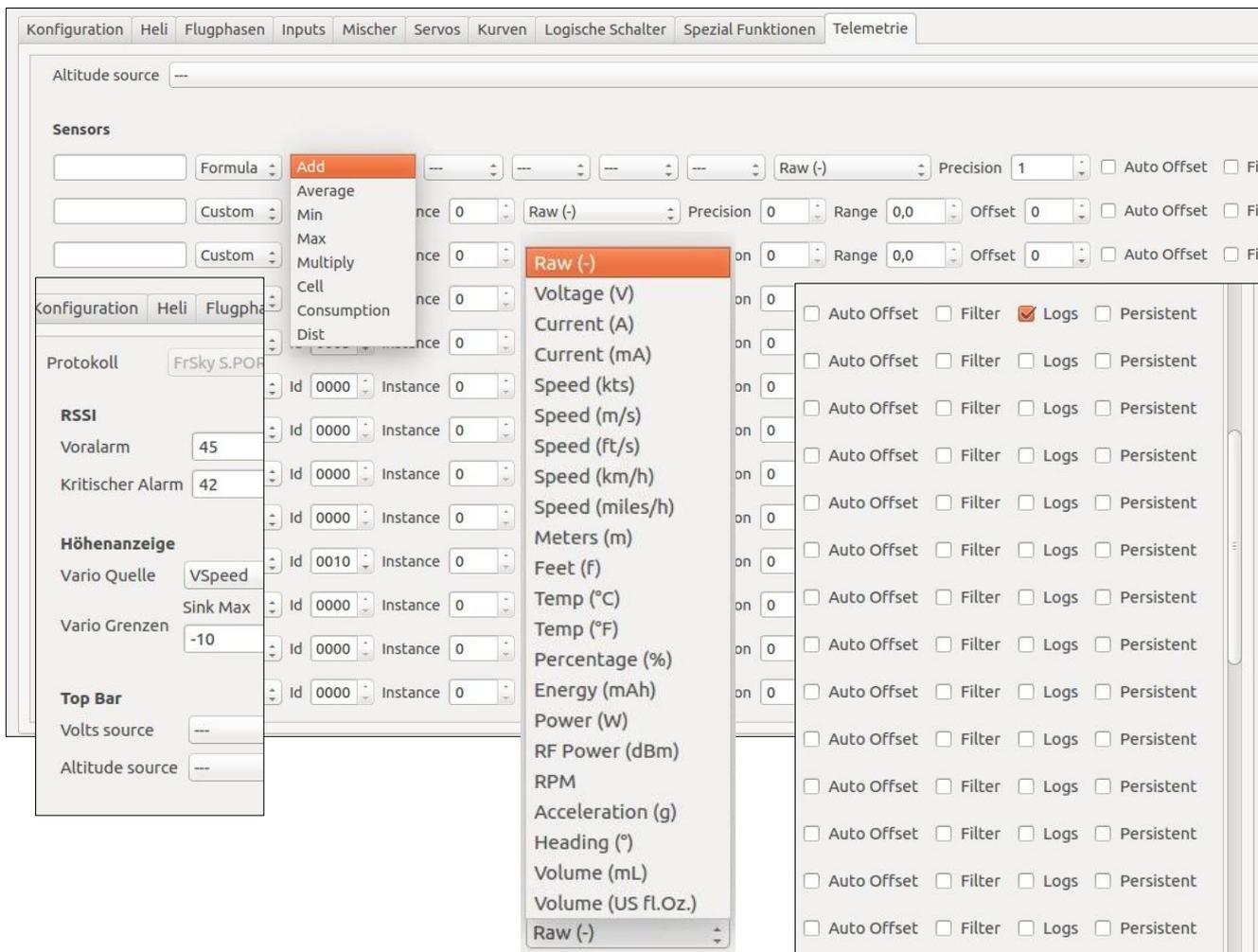
Anzeige im Telemtrieschirm

```

MODELL01      7.5V Tmp1
PWR           00 PWR+ 00 PWR- 00
Verb         0 Verb+ 0
    
```

Beispiel: Telemetrie unter Companion V2.1 konfigurieren

Unter Companion V2.1 ist der Telemetrie Bildschirm sehr umfangreich geworden. Neben den normalen Werten wie RSSI, SWR und Vario ist jetzt für jeden Sensor bzw. Sensorberechnung eine eigene Zeile verfügbar.



Die Telemetriewerte sind dynamisch!

Nur was als Telemetriewert definiert und auch als Telemetriewert reinkommt, kann auch an anderer Stelle mit dem 4-stelligen Namen weiterverarbeitet werden.

Also erst die Telemetrie definieren und einen Namen vergeben, dann erst erscheint dieser Name in den anderen Menüs als Auswahl z.B. in den Spezial Funktionen, Inputs, Mischern, log Schaltern, usw.

Übersicht aller Telemetrie ID-Bereiche 2Byte und 1Byte**Bisherige FRSky-Sensoren: neue 2-Byte ID / alte 1-Byte ID**

Name	neue ID	alte ID
RSSI	F101	25
SWR	F105	25
Alt	0100	01 FVAS-02 Variometer
Cels	0300	02 FLVSS
Curr	0200	03 FCS-40
Temp1	0400	05
Temp2	0410	05
RPM	0500	05
Fuel	0A10	11

Übersicht aller möglichen FrSky Phys. ID-Bereiche als 2 Byte Version

RSSI_ID	F101	Empfängersignalstärke RSSI
ADC1_ID	F102	Analogwert AD1 vom Empfänger
ADC2_ID	F103	Analogwert AD2 vom Empfänger
BATT_ID	F104	Sender Akku-Spannung
SWR_ID	F105	Sender SWR-Werte Antennenüberwachung
T1_FIRST_ID	0400	Temperatursensoren
T1_LAST_ID	040F	
T2_FIRST_ID	0410	
T2_LAST_ID	041F	
RPM_FIRST_ID	0500	Drehzahlsensoren
RPM_LAST_ID	050F	
FUEL_FIRST_ID	0600	Füllständssensoren
FUEL_LAST_ID	060F	
ALT_FIRST_ID	0100	Höhen des Vario
ALT_LAST_ID	010F	
VARIO_FIRST_ID	0110	Steigen/Sinken des Vario
VARIO_LAST_ID	011F	
ACCX_FIRST_ID	0700	Beschleunigungssensoren
ACCX_LAST_ID	070F	
ACCY_FIRST_ID	0710	
ACCY_LAST_ID	071F	
ACCZ_FIRST_ID	0720	
ACCZ_LAST_ID	072F	
CURR_FIRST_ID	0200	Stromsensor im FCS-40
CURR_LAST_ID	020F	
VFAS_FIRST_ID	0210	Spannungssensor im FCS-40
VFAS_LAST_ID	021F	
CELLS_FIRST_ID	0300	Spannungssensoren im FLVSS mit Gesamt und Einzelzellen
CELLS_LAST_ID	030F	

openTx für Taranis Anleitung Deutsch

GPS_LONG_LATI_FIRST_ID	0800	GPS Koordinaten
GPS_LONG_LATI_LAST_ID	080F	
GPS_ALT_FIRST_ID	0820	GPS Höhe
GPS_ALT_LAST_ID	082F	
GPS_SPEED_FIRST_ID	0830	GPS Geschwindigkeit
GPS_SPEED_LAST_ID	083F	
GPS_COURS_FIRST_ID	0840	GPS Kurs
GPS_COURS_LAST_ID	084F	
GPS_TIME_DATE_FIRST_ID	0850	GPS Zeit
GPS_TIME_DATE_LAST_ID	085F	

//FrSky wrong ID ?

BETA_VARIO_ID	8030
BETA_BARO_ALT_ID	8010

Wenn man diese ID-Werte kennt, kann man unter Companion gleich mal vorprogrammieren, ohne vorher am Sender die tatsächliche Telemetrie zu kennen.

Ein weiteres gutes Sensorsystem mit div Steckern, das auch am FrSky S-Port läuft:

<http://www.sm-modellbau.de/pdf/SM%20Anleitung%20UniSens-E%20v1.11.pdf>

SM Modellbau in 82407 Wielenbach

Unisens-E liefert schon 7-8 Werte

Die ID zum einstellen von Unisens-E

Name	Wert	ID	Sub-ID
Curr	A	0200	20
VFAS	V	0210	20
Fuel	%	0600	20
VSpd	m/s	0110	20
ALT	m	0100	20
RPM	rpm	0500	20
Temp	°C	0410	20
VerB	mAh	Berechnung Verbrauch via openTx	



Etwas mehr ins Detail der Telemetrie:

Die Telemetrie ab openTx V2.10 ist stark erweitert und kann viel mehr als bisher.

- Jeder Sensor hat **EINE** eindeutige Phys-ID und **MEHRERE** Daten-ID (bis zu 16) mit den eigentlichen Datenwerten
(Diese ID's sind in der Datei oXs_config.h definiert und können dort geändert werden.)
- Es können max $28 - 2 = 26$ Phys. Sensoren am S-Port des Empfängers angeschlossen werden, Jeder Sensor selbst kann wieder 1 bis 16 Daten mit den Daten-ID liefern.
- Es können mehrere gleiche Sensoren angeschlossen werden, dann müssen sie aber unterschiedliche Phys-ID haben. d.h. die Sensor eigene Phys-ID muss man ändern können. Also darauf achten beim Kauf, wenn an mehrere gleiche Sensoren betreiben will.
- Die Sensoren liefern mit ca 2-5 Hz ihre Daten an den Empfänger ab (keine 10 Hz)
- openTx erkennt automatisch welche Sensoren angeschlossen sind, und liefert sofort die Werte und ID, das ist also ein dynamischer Vorgang, nicht statisch.
- jeder Sensor-Datenwert mit dem man auch tatsächlich arbeiten will **muss** unter openTx einen **Namen** erhalten (4 Zeichen) z.B. Zel1, Zel2, Zel3, Str, Pwr, Verb, Spg, usw.
siehe unter: **Neuen Sensor definieren**
- Sensorwerte kann man in der Telemetrieseite gleich verrechnen / Berechnungen anstellen.
z.B Stromverbrauch, Leistung, Min Max, Mittelwert, roh und gefiltert. usw.
Auch diese brauchen einen eigenen **Namen** (4 Zeichen)
- Auf diese Namen kann man dann **ÜBERALL** zugreifen und sie verarbeiten
z.B in den Spezialfunktionen mit diesem Namen etwas tun!
z.B in den Inputs und in den Mischern!

Mit openTX können 32 Datenwerte (=32 slots) pro Modell gleichzeitig ausgewertet und verarbeitet / verrechnet werden.

Beispiel:

X-Empfänger liefern immer RSSI und A1 (Empfängerakku) (2 slots)

Der Frsky RPM-Sensor liefert 3 Datenwerte 1xRPM und 2x Temperatur (3 slots)

Der Frsky 40A Stromsensor liefert 2 Datenwerte 1x Strom und 1xSpannung (2 slots)

Der Frsky Spannungssensor liefert 7 Datenwerte 1x Gesamtspannung und 6x Einzellzellen (7 slots)

Das Vario liefert 2 Datenwerte: Höhe und vertikales Steigen/Ssinken (2 slots)

Wenn ich davon alles verwenden will, sind damit 16 Slots belegt.

Ein Unisens-E liefert 6 oder 7 Datenwerte (7 slots)

Das GPS-Sensor liefert 5-9 Datenwerte (Länge, Breite, Zeit, Kurs, Höhe,) (5-9 slots)

Neue Werte aus der Telemetrie berechnen.

Aus vorhandenen Telemetriewerten können neue zusätzliche Werte Berechnet werden.
Verbrauch in mAh, aktuelle Leistung in Watt, höchste Leistung, niedrigste Spannung,
Dabei wird auch je 1slot verbraucht.

SWR und Sender-Akku-Spannung kommen vom Sender,
das sind zwar keine echten Telemetriewerte, belegen aber je 1 slot wenn sie verwendet werden.

Unter Companion V2.10 die Telemetrieseite:

Dort sind 32 Zeilen mit denen man diese 32 Telemetriewerte definieren kann.
Dazu gehören auch die eigenen Berechnungswerte z.B. Verbrauch oder Leistung

Alles zusammen eben 32 Datenwerte = 32 Slots

Diese 32 Slots können auch gleichzeitig auf die SD-Karte aufgezeichnet (gelogt) werden.

Companion V2.10 Telemetrie Werte definieren und Berechnungen machen

Sensor ID-Werte definieren und erweiterte Berechnungen machen (vergleiche ID-Liste)
 Temperatur, Drehzahl, Vario, Strom, Spannung, Leistung, Verbrauch, Zellenspannungen



OpenTx V2.1x Telemetriewerte in Anzeigewerte umrechnen im Detail

Ratio: In Abhängigkeit ob und was man in Ratio einträgt wird unterschiedlich gerechnet:

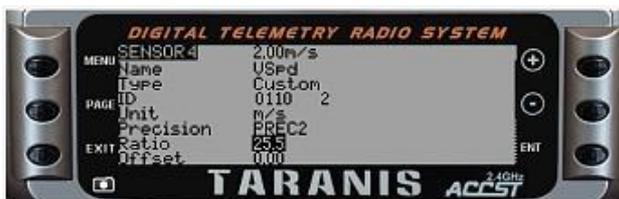
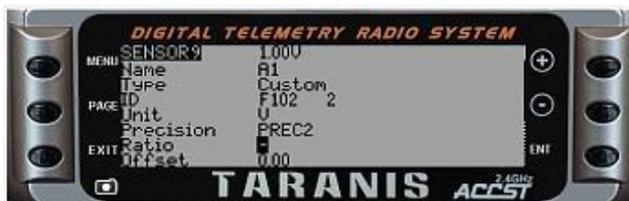
Ratio = "-" Dann wird der Telemetriewert einfach durch 10 geteilt, sonst nichts weiter.
 $[(\text{Telemetriewert} / 10)] + \text{Offset} = \text{Anzeigewert}$
 Beispiel: $[(1000/10)] + 0 = 100$

Ratio nicht "-" Dann wird der Telemetriewert durch 256 geteilt, dann mit Ratio multipliziert.
 $[(\text{Telemetriewert} / 256) * \text{Ratio}] + \text{Offset} = \text{Anzeigewert}$
 Beispiel: $[(1000 / 256) * 256] + 0 = 1000$

Damit kann man alles passend umrechnen !

Beispiel: Bei deinem SM-Vario der Steigen in cm/s ausgibt, Steigen mit 200cm/s also 2,00m/s
 $(200/255)*2,5 = 1,96$ (genauer geht es, nicht da bei Ratio 2,55 nicht möglich ist)

Unter Companion, Simulator, mit F4 und Telemetriesimulation aktiv, kann man das schön ansehen.

Merke:

Alle Telemetrie-Werte die von einem S-Port Sensor kommen und deren ID's im FrSky-System bekannt sind, kommen in den richtigen Dimensionen und Werten rein, mit einer Auflösung von 1,0 oder 0,1. Trotzdem kann man sie anpassen und umrechnen.

Ergänzung zu den Analog Eingangswerten A1 bis A4:

Dort können ja beliebige physikalische Werte als Spannungswert anstehen.
z.B. Ein 3 Zellen Lipo-Akku $3 \cdot 4,2V = 12,6V$ sollen auch als 12,6V angezeigt werden.

Verrechnung am A/D Wandler:

Am A/D Wandler des Prozessor dürfen max nur 3,3V anliegen.
Diese werden mit 10 Bit digitalisiert (=1024), dann um 2 Bit verschoben wg LSB-Rauschen.
Damit stehen für 3,3V effektiv 8 Bit = 256 (0 bis 255) zur Verfügung

Jetzt greift wieder die Umrechnung mit Ratio in bekannter Weise
[(Telemetriewert / 256) * Ratio] + Offset = Anzeigewert
 $3,3V \implies$ umgesetzt in $\implies 255bit / 256 * 3,3Ratio \implies 3,3V$ als Anzeige

Ratio Anzeige

1,0 = 1,00V
2,0 = 2,00V
3,3 = 3,3V
13,2 = 13,2V (hier kommt dieses ominösen 13,2 her, $3,3V \cdot 4 = 13,2V$)

Was bedeutet das für die Companion, Telemetrie-Werte-Simulation mit F4

Unter F4 bei A1-A4 nur Werte von 0 bis 255 eingeben,
(also nicht 3,3V oder 33 oder 330 eingeben)

Beispiele 3-6 Zellen Akkuspannungen richtig anzeigen:

3 Zellen Lipo-Akkuspannung: $3 \cdot 4,2V = 12,6V$

A/D Wandler Eingang kann max 3,3V $3,3V \cdot 3 = 9,9V$ zu wenig für 12,6V $3,3V \cdot 4 = 13,2V$ passt
3 : 1 ist ein Spannungsteilerfaktor von 4 1 Teil am A/D-Wandler-Pin, 3 Teile am Vorwiderstand
Spannungsteiler 3 : 1 = $3 \cdot 1k$ zu 1k

4 Zellen Lipo: $4 \cdot 4,2V = 16,8V$

$5 \cdot 3,3V = 16,5V$ zu wenig für 16,8V!, $6 \cdot 3,3V = 19,8V$ passt also
Spannungsteiler 5 : 1 = $5 \cdot 1k$ zu 1k

5 Zellen Lipo: $5 \cdot 4,2 = 21V$

A/D Wandler = $3,3V \cdot 7 = 23,1V$ passt also
Spannungsteiler 6 : 1 = $6 \cdot 1k$ zu 1k

6 Zellen Lipo: $6 \cdot 4,2V = 25,2V$

A/D Wandler Eingang kann 3,3V
 $7 \cdot 3,3V = 23,1V$ etwas zu klein für den Akku mit 25,2V! $8 \cdot 3,3V = 26,4V$ passt also
Spannungsteiler 7 : 1 = $7 \cdot 1k$ zu 1k

Die Spannungsteiler-Widerstände muss man rausmessen und zusammenstellen
Man sollte auf Fehler kleiner 0,3% kommen. ($100\% / 255 = 0,39\%$)

Parameterübersicht für Telemetrierrechnung

Präzision	= Gibt nur die Nachkommastellen an 0 oder 0,0 oder 0,00
Ratio	= Multiplikator von 1000 bis 0,1 (0,1 ist Teiler mit 10)
Offset	= reine Linearverschiebung
Permanent	= letzter Wert merken auch wenn Sender ausgeschaltet wird, z.B. Verbrauch mAh
Filter	= gleitender Mittelwert der 5 (10) letzten Werte, weniger Sprünge, Signalberuhigung
Log	= Aufzeichnen auf die SD-Karte
AutoOffset	= erster Wert wird als Offset genommen, z.B. Alt Höhe automatisch als Null

OpenTx Update am Sender mit der USB Schnittstelle, SD-Karte

Achtung: Ab openTx V2.00 wurde vieles umgestellt und vereinfacht.

- 1. Alle Sender die noch nicht auf opentx V2.00 umgestellt wurden brauchen noch den Zadig-Treiber um die Firmware auf den Sender zu flashen.**
- 2. Ab OpenTx V2.00 braucht man das nicht mehr, denn dort kann man die Firmware einfach auf die SD-Karte kopieren und von dort direkt flashen. Kapitel E**
- 3. Das Verhalten der USB-Schnittstelle ist je nach Softwarestand unterschiedlich Also aufpassen wann man das USB-Kabel einsteckt und den Sender einschaltet.**

Hinweis für CompanionV1.52, openTx r2940 und OpenTx V2.00

Ich habe bewußt den nachfolgenden Installations-Teil für Companion9x V1.52 und opentx r2940 aus dieser Anleitung nicht entfernt, nur durchgestrichen, weil viele Sender noch nicht auf opentx V2.00 umgestellt sind und man die Funktionen des Zadig-Treibers, DFU-Util zumindest einmal braucht, um den neuen Bootloader für opentx V2.00 auf den Sender zu bringen.

**Kapitel B: Die ersten 2-3 Seiten sind nur für openTx V1.52, der Rest ist weiterhin gültig
Kapitel E: Besonderheiten ab openTx V2.00, Flashen, Bootloader, USB**

Auf dem Prozessor selbst ist immer ein Bootloader **fest installiert**, damit kann man ganz einfach ein neues Firmware-Update von openTx übertragen. Dazu muss aber vorher einmalig ein spezieller Treiber **zadig.exe** installiert werden der den Bootloader des Prozessors ansprechen, lesen und schreiben kann. Die Installation des Teibers macht manchmal Ärger, je nach PC!
Das eigentliche Programm um von und zum Sender zu schreiben und zu lesen heist **dfu-util.exe**

Ab opentx V2.00 ist in der Software selbst ein zusätzlicher Bootloader vorhanden der keinen Zadig-Treiber braucht. Ablauf siehe Kapitel E

Bei Taranis X9D und X9E geht das alles via USB-Schnittstelle.

Varianten des USB Verbindungsaufbau mit Taranis X9D, X9E

- 1. Taranis ausgeschaltet => USB anschließen => STM32 Bootloaderfunktion aktiv (DFU, Zadig)**
- 2. Taranis eingeschaltet => USB anschließen => Joystickfunktion aktiv**
- 3. Taranis Trims halten, einschalten => USB anschließen => SD-Datenträgerfunktion, Update, Boot**

Damit wären alle wesentlichen Dinge der openTx-Software erklärt.

Um aber sehr bequem damit arbeiten zu können sollte man sich das Programm **Companion** laden (das ist Freeware und eine Erweiterung von eePe)

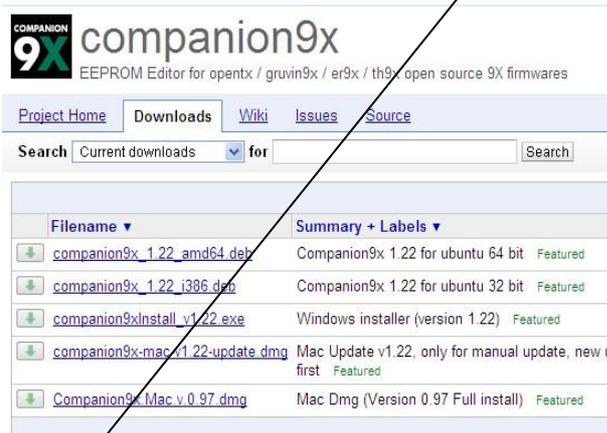
Companion ist mehr als ein reiner Software Simulator, man kann sich damit die Software Optionen für openTx zusammenstellen, Modelle am PC programmieren und simulieren.

Companion greift für die Sender **Th9x, 9XR** auf **AVRdude** zurück. Damit kann man für diese Sender den Flashspeicher und das EEPROM auslesen, überschreiben und neu programmieren.
Siehe deutsches Handbuch open9x für Th9x und 9XR

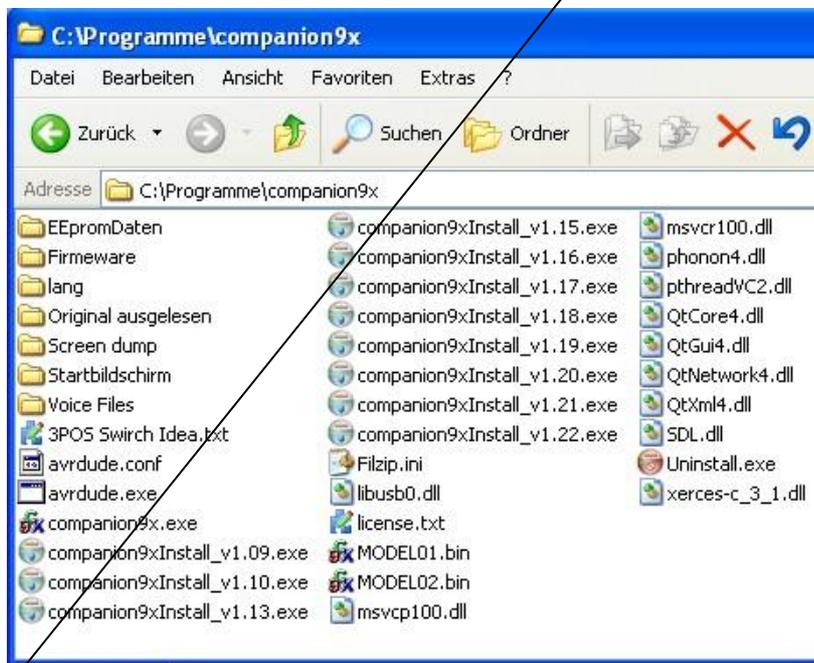
Teil B Companion Einführung Step by Step

→ Für OpenTx V2.00 Siehe Teil E, ansonsten fast gleiche Bedienung!

Companion9x ist Freeware für PC, MAC, LINUX und man lädt es sich aus dem Internet. Dort ist immer die neueste Version verfügbar unter : <http://code.google.com/p/companion9x/>
Unter **Downloads**, die aktuellste Version z.B. **Companion9xinstall_v1.52.exe** laden
Für die Installation braucht man Administratorrechte, da Laufwerke eingerichtet werden.



Beim Installieren erzeugt Companion9x div Verzeichnisse und auf dem Desktop einen Startbutton, zum Beispiel so wie unten dargestellt. Einige Unterverzeichnisse sollten wir dort noch selber erzeugen um unsere Daten besser zu strukturieren.



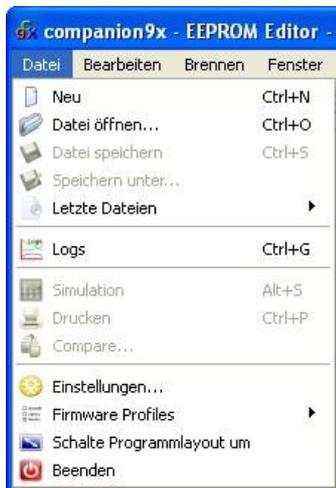
Damit sind wir vorbereitet und können Companion9x starten und sehen diese Programmleiste



openTx für Taranis Anleitung Deutsch

Jetzt müssen wir Companion9x erst mal einrichten damit es den richtigen Sender simuliert, Deutsch als Sprache verwendet und die einzelnen, gewählten Softwaremodule zum Sender passend zusammenstellt. Dazu muss eine Internetverbindung stehen denn das läuft Online.

Wir benötigen die Funktion Einstellungen, unter Datei, Einstellungen zu finden.



Wenn wir Einstellungen anklicken erhalten wir ein großes Fenster mit vielen Möglichkeiten. Um uns nicht zu verrennen passt das meiste schon, aber einfach mal vergleichen.

Entscheidend sind die passenden Verzeichnisse, und die

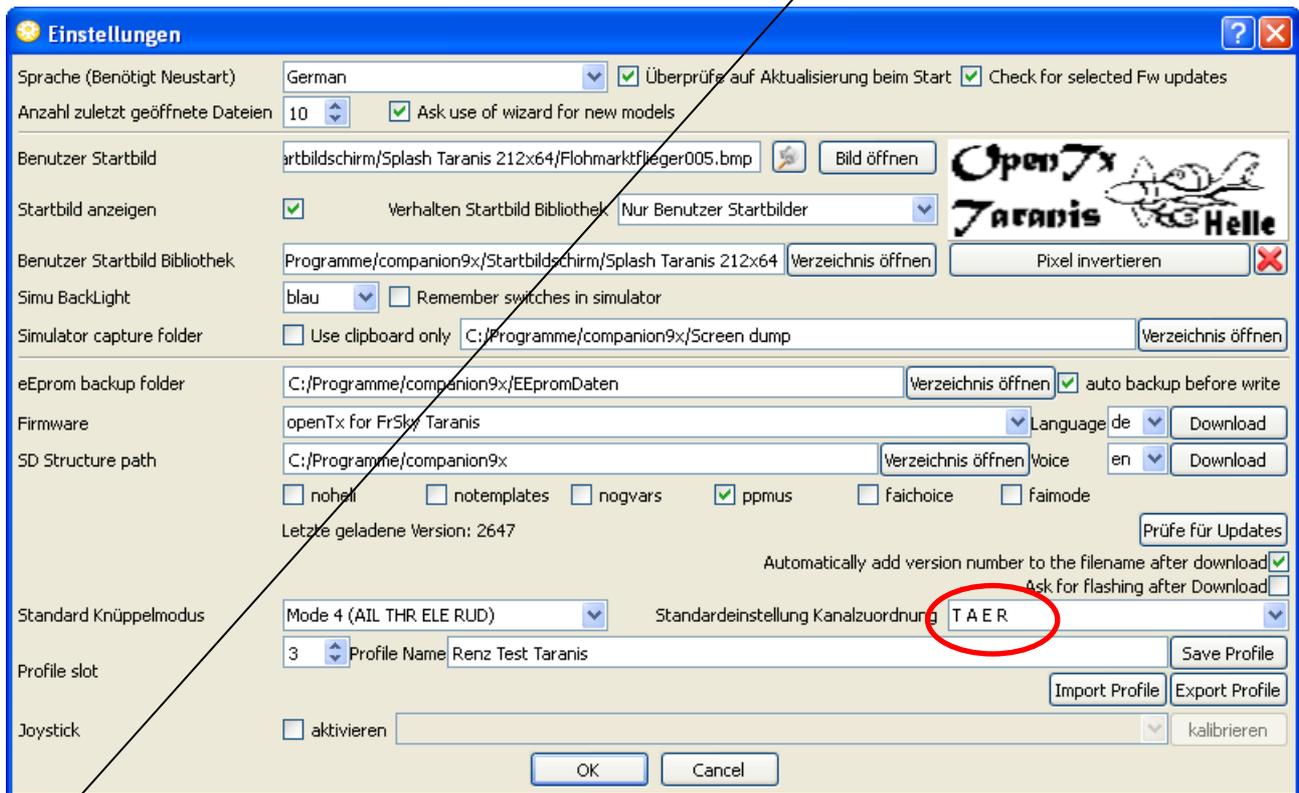
Senderauswahl = Firmware für FrSky Taranis

Für FrSky Taranis sollte/muss das dann so aussehen.

Für andere Sender sieht das etwas anders aus.

Bitte hier **nicht irgendwie rumspielen** wenn man nicht weiß was man tut und welche Funktionen was bedeuten!

Jetzt Zeile Firmware, ganz rechts, **Download** drücken, dann stellt Companion9x die aktuellste Sender-Firmware aus dem Internet zusammen und wir sind bereit.



Alle Eingaben vergleichen und auf eigene Unterverzeichnisse anpassen

Beachte: TAER = GQHS= Gas, Quer, Höhe, Seite auf Kanal 1, 2, 3, 4 vorbelegen

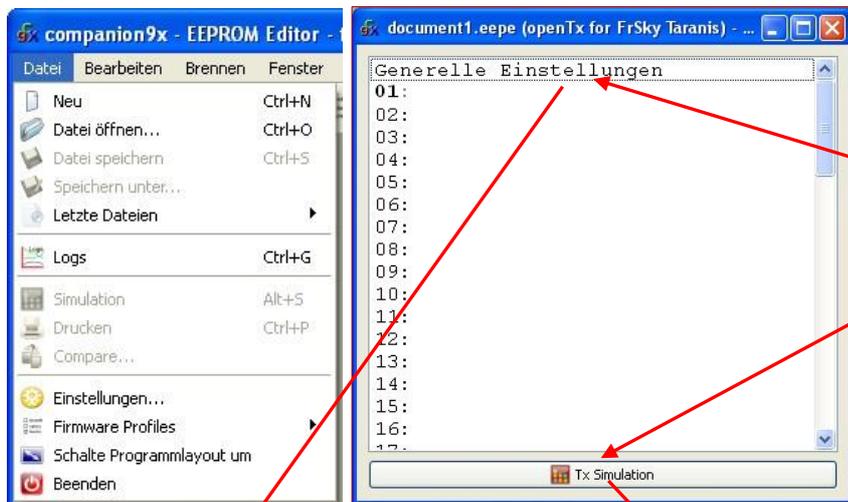
Das ist die Reihenfolge der Kanalvorbelegung für neue Modelle!

Hier Anpassen nach eigenen Bedürfnissen (Graupner, Futaba Multiplex Belegung?)

Soweit erst mal eingestellt und Companion9x ist vorbereitet.

Simulation des Sender, Grundeinstellungen, Modell erzeugen

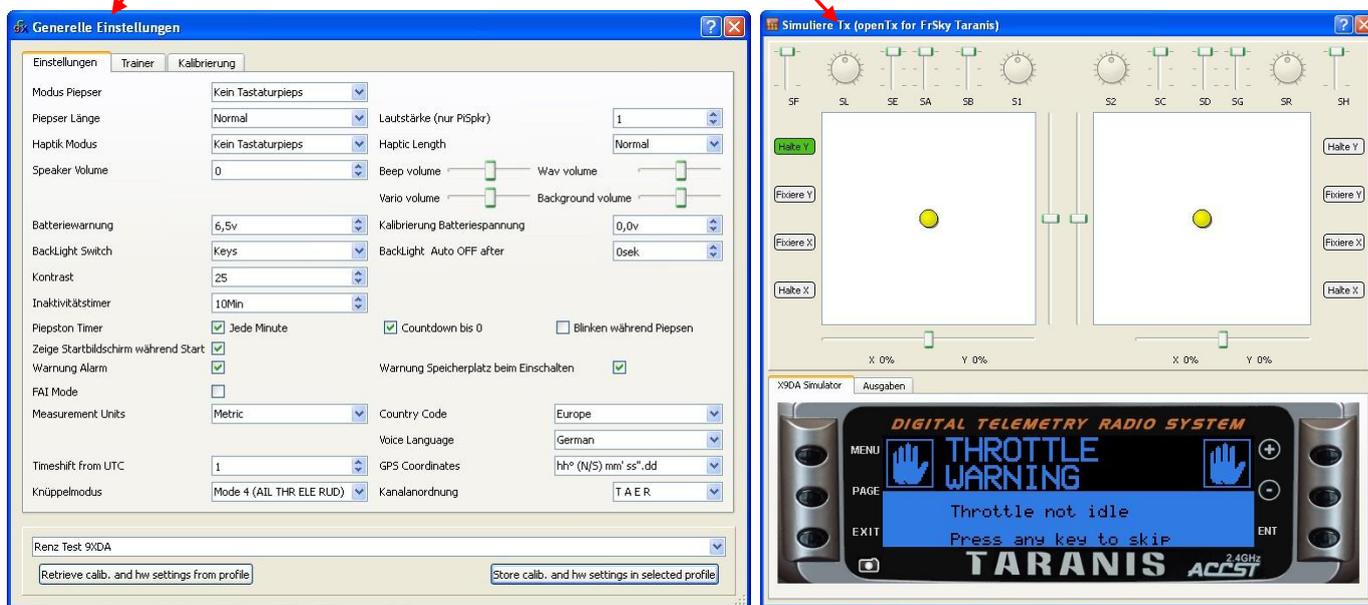
Mit Datei Neu legen wir Modelle an und erhalten das Fenster des „Modellspeichers“



Ganz oben steht:
Generelle Einstellungen,
das sind die
Sender -Grundeinstellungen!

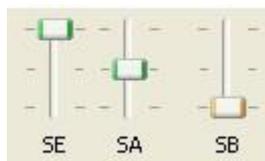
Ganz unten steht:
TX-Simulation, damit wird der
Sender 1:1 nachgebildet und wir
müssen auch die Tasten am Sender
genauso bedienen und mit der Maus
1:1 drücken.

Das sieht dann so aus und auch am Sender wird alles ganz genau so 1:1 dargestellt
Sender Grundeinstellungen **TX Simulation**



Diese beiden Fenster machen wir gleich wieder zu, denn da machen wir erst mal gar nichts!

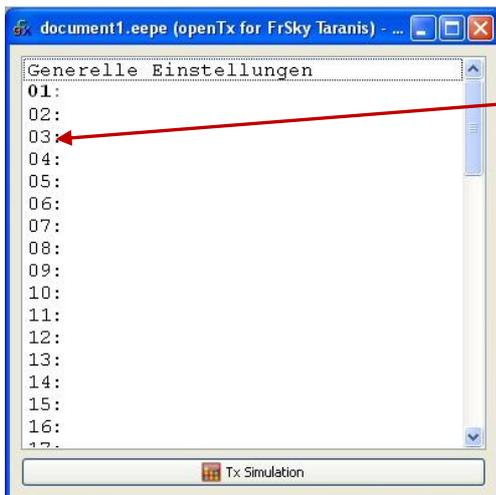
Vergleich Schalterstellungen in Companion, PC-Simulation und am Sender



SE ↑ Down- Stellung, Schalter am Sender zeigt von mir weg, nach unten
SA — die Mittelstellung ist klar
SB ↓ Up- Stellung, Schalter am Sender zeigt zu mir her, nach oben

openTx für Taranis Anleitung Deutsch

Wir gehen zurück in das Fenster „Modellspeicher“ und legen ein neues Modell an.

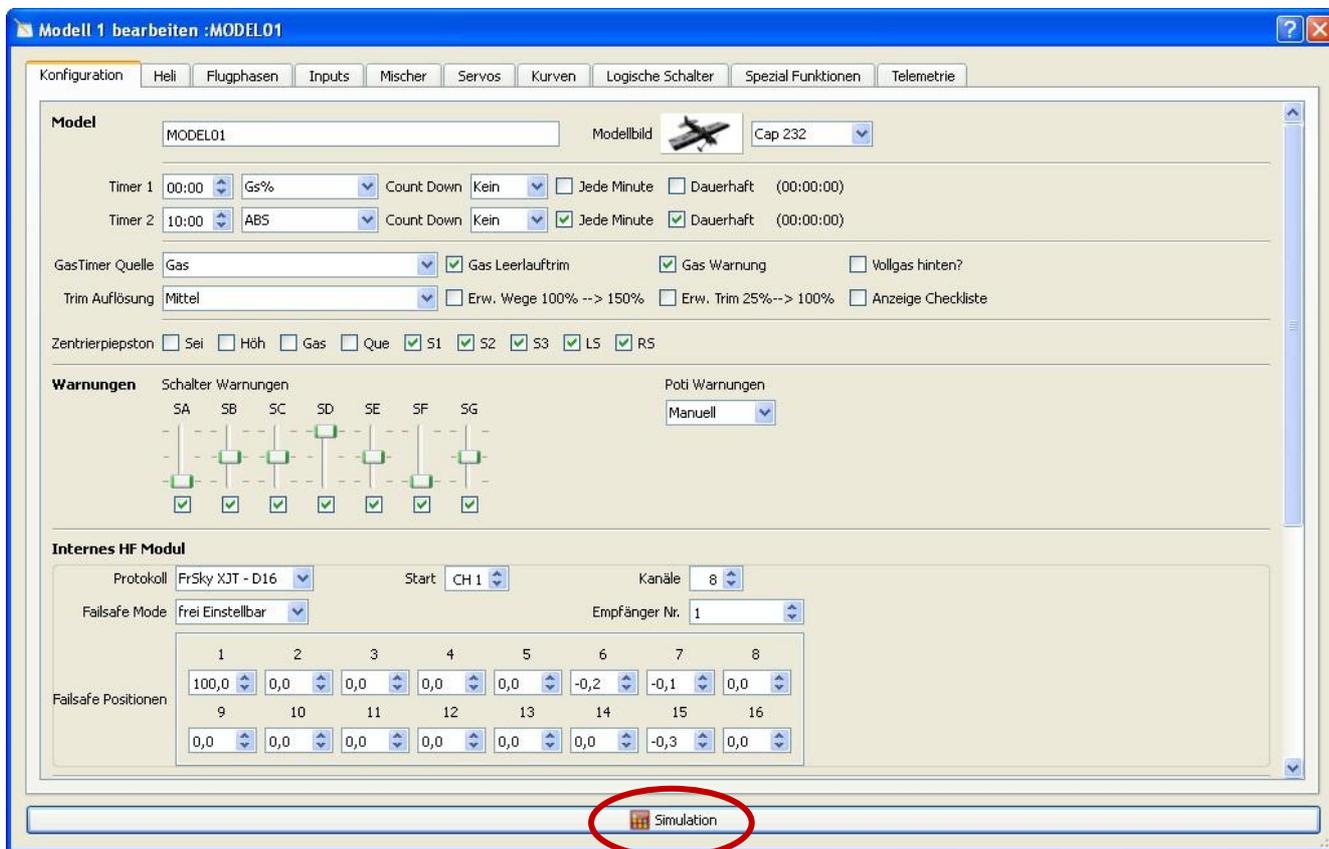


Zum Beispiel im Speicher 03:
Dort einen **Doppelklick** und wir kommen in die
„Modell-Einstellungen“

Hier wird das Modell eingestellt und programmiert

Zuerst aber linker Reiter: Grund-Konfiguration des Modell
und Modellname eintragen

Das sieht dann so aus und jetzt sollten wir mal etwas spielen und
im Handbuch openTX lesen.



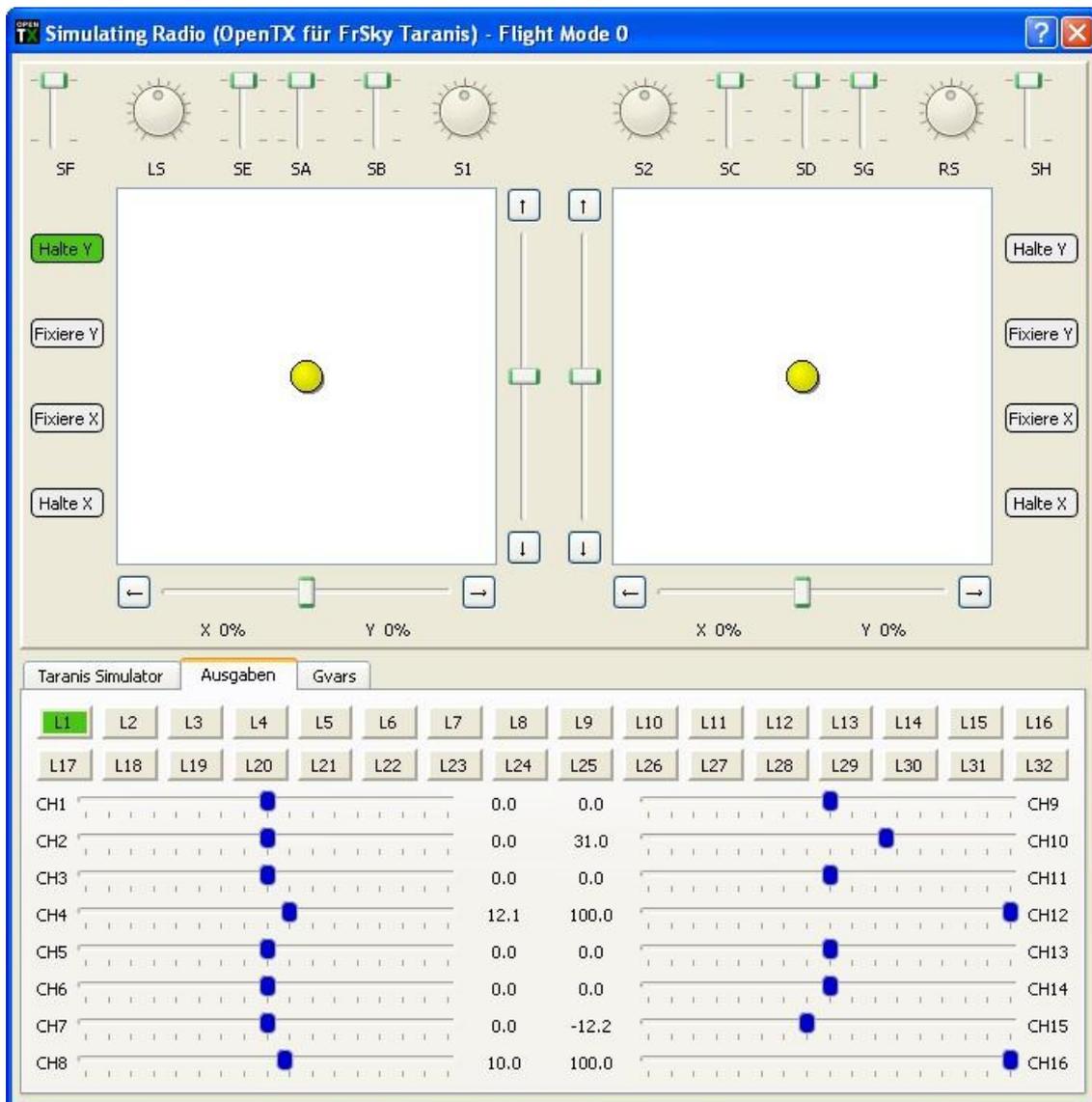
Modellname, Stoppuhren, Trimmung usw., bloß was bedeutet das alles im Detail?

Ganz unten in diesem Fenster steht wieder **Simulation**, aber **nicht** TX-Simulation!

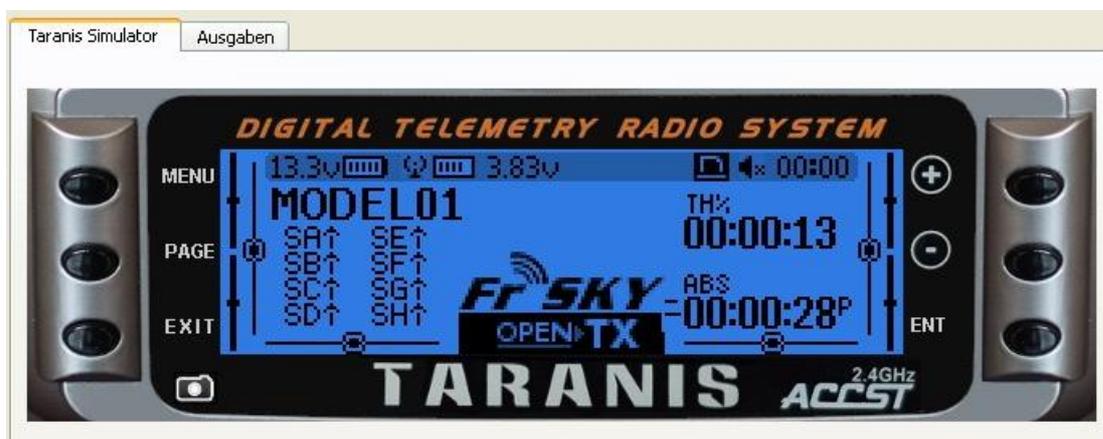
Das ist dann die **Softwaresimulation** mit der wir (fast) immer arbeiten.

Das ist einfacher, da wir hier nicht den Sender 1:1 in den Tasten drücken nachbilden müssen.

Sender-Simulation, Taranis Simulator, Ausgabe-Simulation



Softwaresimulation als Kanalsimulator



Sendersimulation mit allen Menüs und Anzeigen (immer in Englisch)

Das erste Modell „Programmieren“ Schritt für Schritt

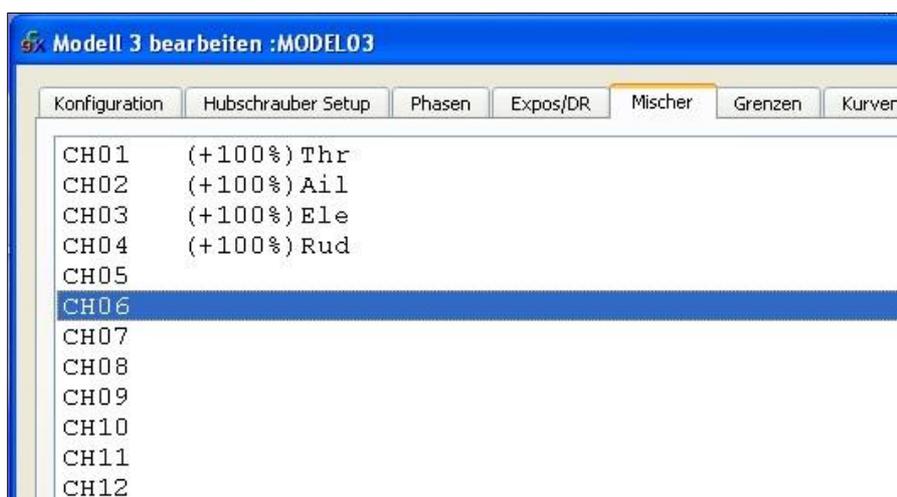
Die Reihenfolge der Reiter entspricht der Reihenfolge der Hauptfunktionen der Software, vergleiche mit Softwarestrukturblatt Modelleinst. aus der Kurzanleitung und das Handbuch.

Konfig, Hubi, Flugphasen, DR/Expo, Mischer, Servowege, Kurven, Schalter, Funktionen, Telemetrie, Voreinst.



Das zentrale Element bei openTX sind die Mischer. Alles läuft über Mischer, hier wird alles andere zusammengeführt, verrechnet und den Kanälen zugeordnet.

Da bei openTX alles per Mischer läuft, den Reiter Mischer anwählen und schon sind wir hier.



Und da steht auch schon was drinnen, beim Aufruf des neuen Modell automatisch erzeugt. Für Kanal1 bis Kanal4 sind schon die 4 Knüppel zugeordnet mit 100% Weg.

Thr = Gas

Ail = Querruder

Ele = Höhenruder

Rud = Seitenruder

TAER = **GQHS** das ist die Kanalvorbelegung die wir vorher festgelegt haben.

Es gibt keine festen Zuordnungen zu Kanälen, man kann alles frei belegen!

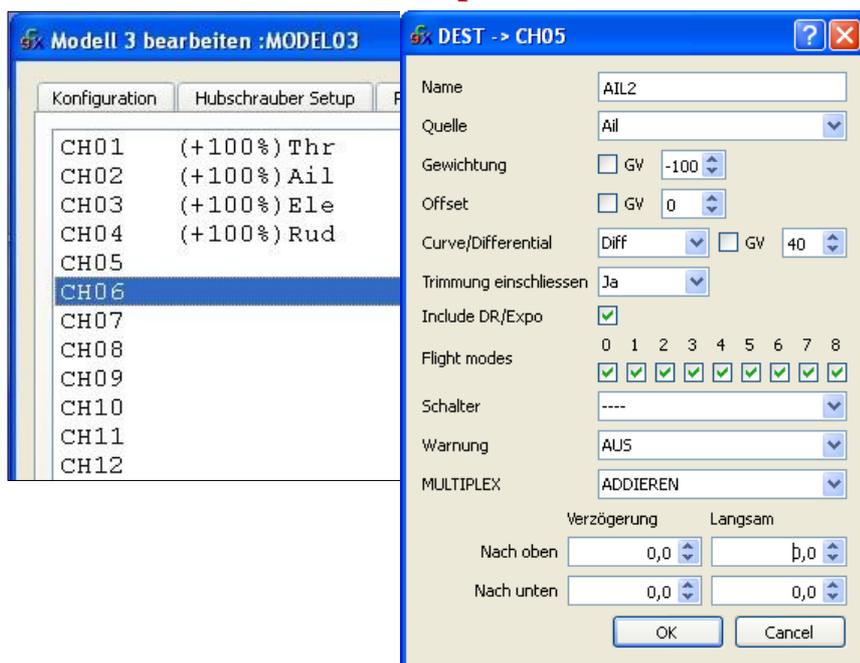
Wer Futaba hat muss sich an eine feste Kanalbelegung halten, genauso bei Graupner und all die anderen. Das ist hier nicht so, jeder Kanal kann für jede Funktion frei belegt werden!

Die Voreinstellungen der Kanalreihenfolge für die Templates findet man unter:

Standarteinstellungen Kanaluordnung Gefunden? Genau, ganz, am Anfang als wir für Companion9x die Einstellungen machten, rechts unten → **TAER** Deutsch: **GQHS**

Eigentlich wären wir jetzt schon fertig, ein einfaches 4-Kanal „Trainer“ Modell fertig zum einfliegen. Das hat aber noch nichts mit programmieren zu tun.

Das Mischer Fenster am Beispiel Kanal 6



Wir wollen mal Kanal 6 mit etwas belegen, **also Doppelklick drauf und das Mischer-Eingabe-Fenster** erscheint mit all seinen vielen Möglichkeiten

GV= Globale Variable GVAR
Statt Festwerte eine Variable zur Verrechnung verwenden

Ab openTx V2.0:
Mischerberechnung =
[(Quelle *Gewichtung)+ Offset]

Name: Ein Klarname z.B. QuerLi., Fahrwerk

Quelle: Da kommen die Werte her, Analogwerte, Digitalwerte, Verknüpfungen, andere Mischer usw.

Gewicht: die Quelle wird mit Gewichtung verrechnet oder mit dem Wert der Globalen Variable

Offset: per Offset kann die Quelle noch verschoben werden oder eine GVAR verwenden

Kurve/Diff: sollen Kurven oder Ruderdifferenzierungen verwendet werden oder eine GVAR

Trimm: Trimmungen verwenden ja /nein, welche denn und wie

DR/Expo: Dualrate/Expo Werte anwenden oder Kurven oder GVARS verwenden

Flugphasen: bei welchen Flugphasen soll dieser Mischer aktiv sein.

Schalter: welche Schalter geben den Mischer frei/sperren ihn und wie

Warnungen: Welche Meldungen sollen ausgegeben werden

Multiplex: Wie soll der Mischer zum Kanal verrechnet werden

Pro Kanal braucht man oft mehrere Mischerzeile.

Diese Mischerzeilen müssen miteinander arbeiten.

Sie können Addiert werden, Multipliziert werden oder eine Zeile kann alle Zeilen darüber ersetzen (**Replace**) Dabei tauchen verschieden Zeichen auf **Add += Mult *= Replace :=**

Zeiten:

Verzögerungen: (Einschalt- und Ausschaltverzögerungen getrennt einstellbar)

Nach oben: Zeit bis der Mischer aktiv wird, ab dem er freigeschaltet ist (Einschaltverzögerung)

Nach unten: Zeit bis er wieder inaktiv wird, ab dem er wegschaltet ist (Ausschaltverzögerung)

Langsam: (langsamer Lauf, in beide Richtungen getrennt einstellbar)

Nach oben: Geschwindigkeit des Mischer in positive Richtung

Nach unten: Geschwindigkeit des Mischer in negative Richtung

Denkt man an langsam laufende Ruder nach oben oder unten, oder langsame Servogeschwindigkeiten oder an langsames Ein/Ausfahren von Fahrwerken wird es klarer.

Erst laufen die Zeit-Verzögerungen ab, dann erst das langsame Fahren von Servos.

Damit kann man auch Door-Sequenzen programmieren!

Die Mischer-Verrechnung geht im Prinzip so

[(Quelle * Gewichtung) + Offset] => Mischerwert **Diese Grundformel gilt ab openTx V2.00!**

Gewicht und Offset sind % -Werte 125% = 1,25

Quelle kann % Werte, Festwerte, Schalter oder logische Schalter -100% 0% +100% sein

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail (Quer1)
	(-25%) MAX Schalter (SA↓) (Lande1)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD)
CH05	(-100%) Ail (Quer2)
	(-25%) MAX Schalter (SA↓) (Lande2)
CH06	
CH07	

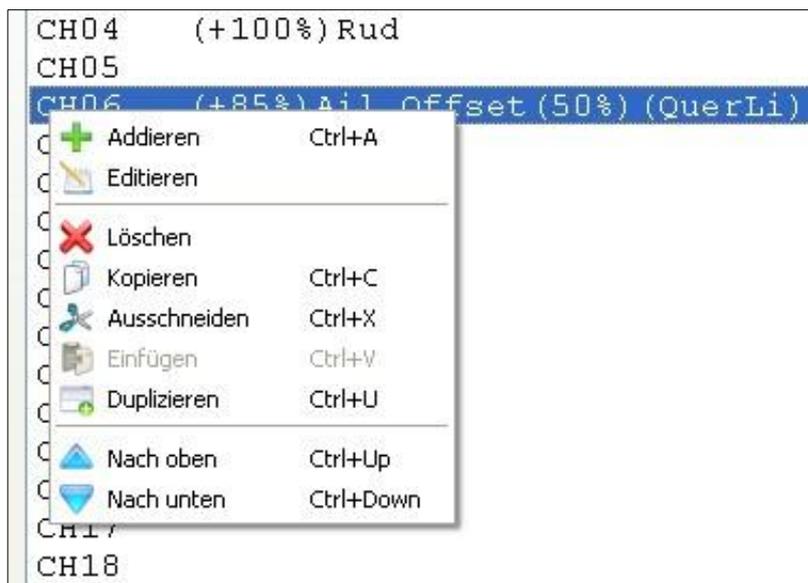
Mehrere Mischerzeilen pro Kanal (bei CH2 und CH5) können miteinander verarbeitet werden.

Sie können Addiert werden **+=**, Multipliziert ***=** werden,

oder eine Mischerzeile kann alle anderen darüberstehende Mischerzeile ersetzen **:=** **R=Replace**

Zusätzliche Mischerzeilen in den Kanal davor/danach einfügen löschen, verschieben,

einfach mit **Rechtsklick**, und schon können wir weitere Mischerzeilen in den Kanal einfügen.



Erste Beispiele einfach mal eingeben und abändern.

Querruderknüppel (AIL) steuert über Kanal 2 das rechte Querruderservo, an Kanal 5 das linke Querruderservos, Ruderdifferenzierung 60%

Beachte: Kanal 2 **positive Gewichtung** (+100%) und Kanal 5 **negative Gewichtung** (-100%)

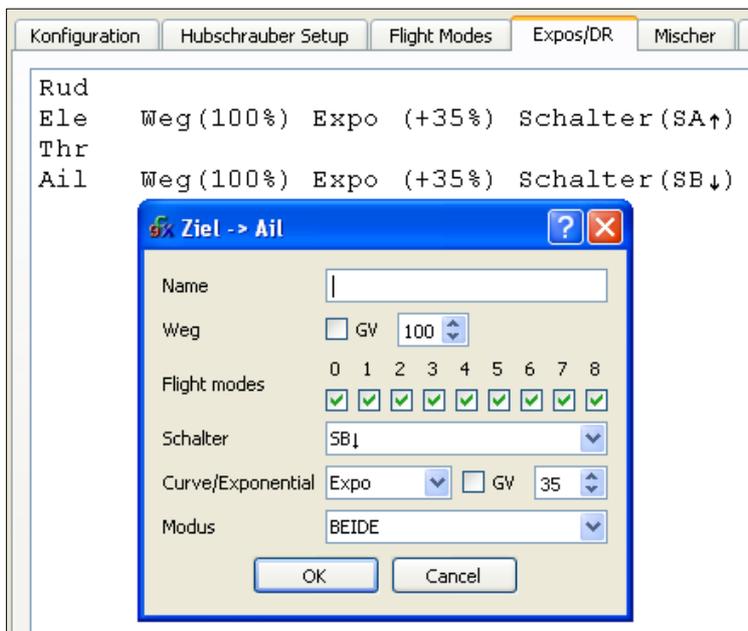
CH01	(+100%) Thr
CH02	(+100%) Ail Diff (60%)
CH03	(+100%) Ele
CH04	(+100%) Rud
CH05	(-100%) Ail Diff (60%)
CH06	

Begründung: CH2 +100% und CH5 -100% warum so und nicht beide auf +100% ?

Wenn der Querruderknüppel nach rechts geht, liefert der Knüppel Ail selbst positive Werte an beide Mischer. Das rechte Querruder CH2 muss nach oben (+100%) **und gleichzeitig** das linke Querruder CH5 nach unten gehen (-100%), dann wird mathematisch richtig gerechnet und auch weitere Mischerverknüpfungen mit diesen 2 Kanäle gehen vorzeichenrichtig ein!!

Jetzt wäre dazu noch Dualrate/Expo auf Höhen und Quer nicht schlecht.
Umschalten auf 80% Weg und ca. 35% Expo mit 2 Schaltern zum aktivieren.

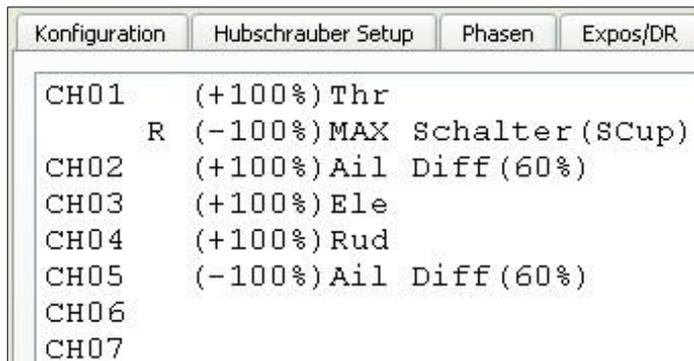
Ok, dann eben ins Menü **Inputs bzw Expos/DR** rein, Doppelklick auf Höhen-Knüppel (Ele) und im Menü die Werte schnell eingeben genauso dann bei Querruder-Knüppel (Ail)



Schalter SA für die Querruder
Schalter SB für das Höhenruder
In beide Richtungen Pos/Neg wirksam
Als Exponentialfunktion
(könnte auch eine beliebige andere Kurve sein)

Statt Festwerte können auch die
Globale Variablen GV1- GV5
verwendet werden

Dann wäre ein Schalter nicht schlecht der verhindert, dass unser Elektromotor losläuft wenn wir versehentlich ans Gas kommen, d.h. also ein Gas Sperr-/Freigabeschalter.



Das macht diese zweite Zeile, die im Mischer für Kanal 1 eingefügt wurde.

R= Replace= Ersetzte alle anderen Zeilen darüber im Kanal1.

Max ist ein Festwert der +100% liefert, mit Gewichtung -100% verrechnet wird und mit Schalter SC auf up aktiviert wird.

CH1 = (Max*Gewichtung) wenn Replace-Zeile per Schalter aktiv ist

Wie geht das schon wieder, ach ja, **Rechtsklick** in der Zeile Kanal 1 [CH01] und Addieren, dann ist eine neue Zeile eingefügt und wir sind wieder im Mischer-Eingabefenster. Dann eingeben:

Quelle = Max Gewichtung= -100% Schalter= SCup Multiplex= Ersetzen und schon ist das ganze fertig.

Konkret: Wenn der Schalter **SC** auf **UP** steht ist diese Mischerzeile, aktiv und liefert dauern -100% an den Kanal1 (Motor-Regler) raus, d.h. der Motor steht, egal wo der Knüppel steht.

Wird **SC** auf **Down** geschaltet ist diese (**R=Replace**)-Zeile nicht aktiv und der Motorregler bekommt seine Werte vom Gasknüppel.

Somit haben wir schon mal ein komplettes Flugmodell programmiert!

5 Kanäle, 4 Servos, 1 Regler, 2 Querruder mit 60% Ruder-Differenzierung

Dualrate umschaltbar 100% auf 80% , Expokurve 0% auf 35% für Quer und Höhe mit 2 Schaltern, Gas Sperre/Freigabe per Schalter.

Damit kann man doch mal anfangen!

Soweit ok, Flieger fertig, nicht ganz! Ein paar Servos laufen irgendwie falsch rum!

Das ist absolut normal, denn je nach Einbaulage, Ruderhornanschluss,

Drehrichtung des Servos selbst, passt es oder eben es passt fast nie. Das ist normal!

Dann mal jetzt noch die Servowege Mittelstellung, Min, Max oder Reverse einstellen.

Ganz einfach ins Menü (Servo)- Grenzen. Upps das ist aber viel und komisch?



Ja, langsam wird es umfangreicher, 32 Kanäle, jeder mit 6 Feldern

Aber das meiste ist ja bekannt Servo-Mitte ist der Offset

Servo-Min, Servo-Max das sind die Servowege, Servo-Drehrichtung/Umkehrung NOR/INV

Der Rest interessiert noch nicht, Symmetrisches Verhalten, PPM-Center einstellen usw.

Das muss aber am fertig aufgebauten Flieger gemacht werden!

Jeden Knüppel einzeln so bewegen dass er zu **positiven Mischer-Berechnungen** für den jeweiligen Kanal führt. Dann das zugehörige Servo per Servo-Invers so umschalten dass das Ruder auch in Richtung positiv, d.h. nach **oben** bzw. **rechts** ausschlägt. Jedes Ruder einzeln abarbeiten.
Höhe ziehen, Seite rechts geben Quer rechts geben Quer links geben.

Hintergrund:

Alle (Mischer)-Berechnungen bei allen Fernsteuerungen dieser Welt sind so ausgelegt, dass positive (Mischer)-Werte zu einer Ruderbewegung nach oben oder rechts führen sollen!

Und zwar egal wie das Servo selber physikalisch eingebaut ist, wo das Gestänge am Ruderhorn eingehängt ist und es deshalb vielleicht noch „falsch rum“ läuft.

Denke an Querruder Kanal2 +100% nach oben **und gleichzeitig** Kanal5 -100% nach unten.

Erst mit den Servoeinstellungen wird die tatsächliche physikalische Laufrichtung so korrigiert, dass sie dem errechneten positiven Mischer-Wert am Ruder entspricht!

**Nicht schon in den Mixern korrigieren, dass es „richtig“ läuft!
Das ist grundlegend falsch!**

Jetzt können wir das auch mal an der Taranis simulieren.

Also wie vorher, Fenster Modellauswahl aufrufen und unten TX-Simulation den “Sender“ starten, das Sender Simulationsfenster erscheint.

Jetzt müssen alle Tasten an der Taranis per Maus 1:1 bedient werden wie am echten Sender.

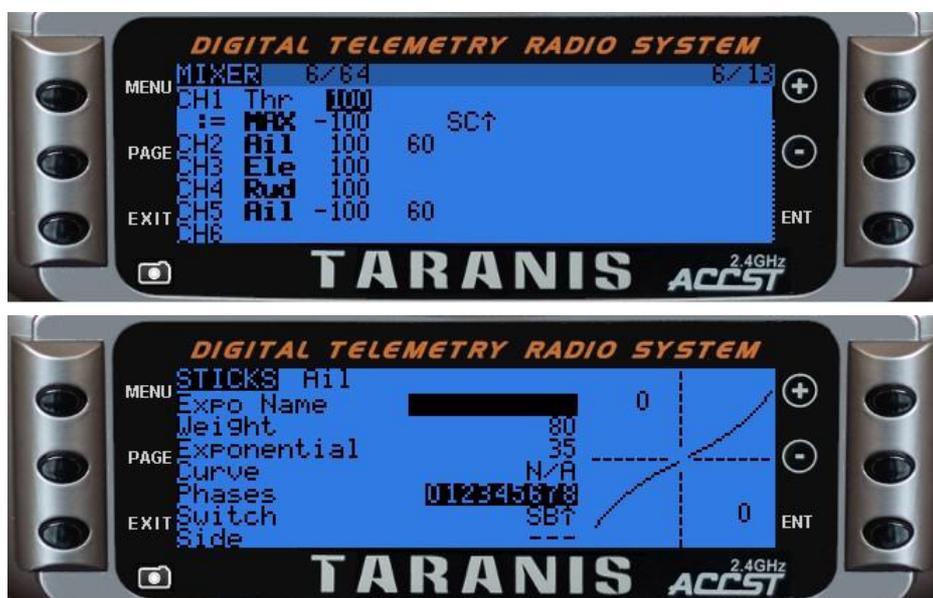
Wichtig, erst mal etwas ausführlich damit spielen, damit man sich daran gewöhnt, Tasten kurz und Tasten lang drücken.

Dazu das Blatt mit den Tastenbelegungen und das Blatt mit der Softwarestruktur herrichten.

Irgendwann mal dann Modell 3 auswählen, denn dort haben wir das obige Beispiel eingegeben.

Jetzt können wir alles am “Sender“ ansehen z.B. via Schalter die Dualrate und die Expokurve umschalten, den Gas-Sperrschalter ausprobieren und am Sender den Kanalmonitor ansehen.

Ach ja, am „Sender“ gibt es unterhalb der EXIT-Taste ein Kamerasymbol, damit kann man von der LCD-Anzeige des Simu-Senders einen Screen-Shot machen der abgespeichert wird.

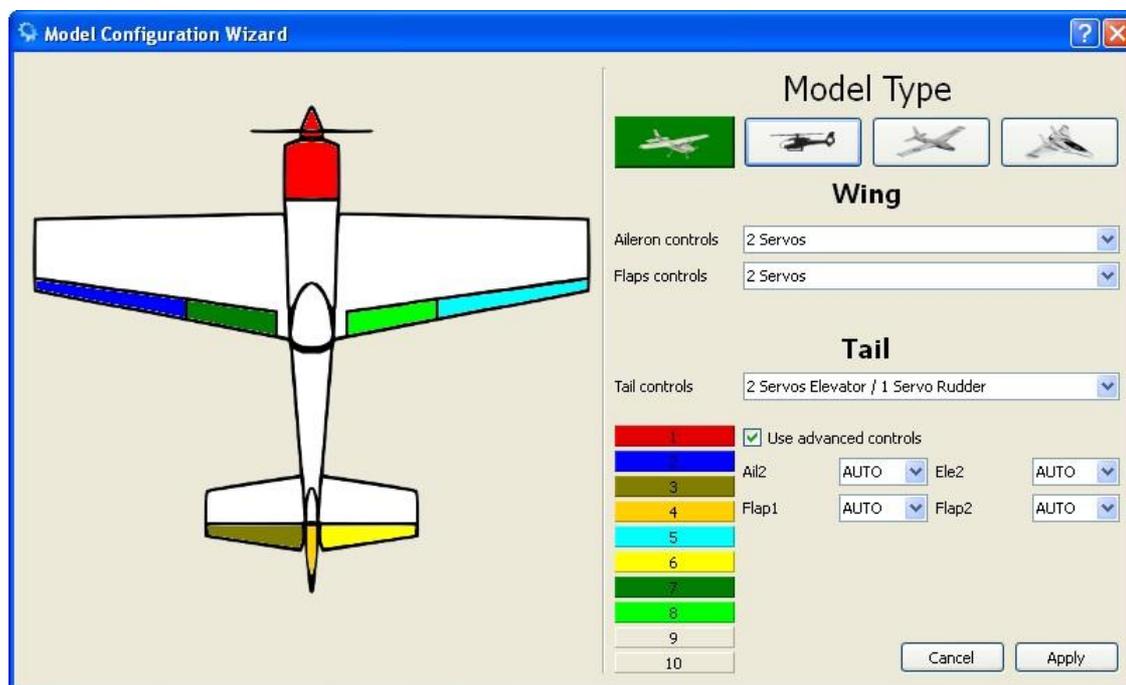
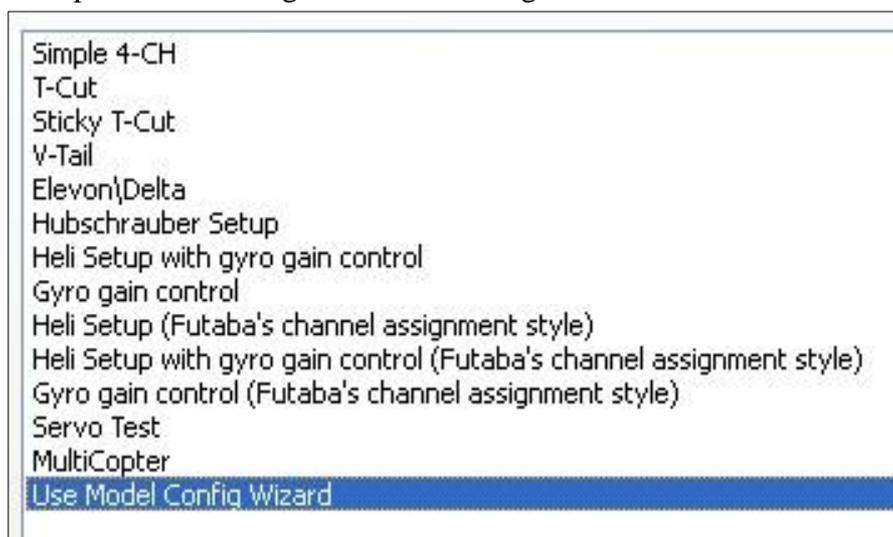


Templates und Modellkonfiguration

Viele fertige Beispiele zum Runterladen gibt es in der Library hier: <http://openrcforums.com/forum/viewforum.php?f=38> und hier : <http://rcsettings.com/>

Im Handbuch mal die Beispiel durcharbeiten, da kommt dann bald der Aha- Effekt

Mit Companion9x Vorlagen Modell Konfigurationen zusammenstellen



Der Modellkonfiguration Wizard

Damit kann man sich Modelle zusammenstellen, die Ruder und Servos zuordnen und dann eine Mischer-Voreinstellung laden.

Die Ruder und Gas Anordnung kann komplett frei, oder als Futaba oder JR/Graupner Anordnung ausgewählt werden.

Daten von und zum Sender übertragen

Diese Möglichkeit bleibt auch ab openTx V2.00 weiterhin bestehen!

Die Treiber müssen vorher richtig geladen sein! **zadig.exe** als STM32Bootloader und **dfu_util.exe** als Brennprogramm (Details siehe weiter hinter)

Unter Companion9x gibt es die Funktionen Brennen für EEPROM und Flashen



Im Flashspeicher wird die Sender-Firmware gespeichert

Schreibe Flash Speicher heißt, **überschreibe** das Betriebssystem openTx im Sender

Lese Flash Speicher heißt, das Sender Betriebssystem openTx auslesen und im PC speichern

Dazu darf der Sender nicht eingeschaltet sein!

Im EEPROM werden die Modelldaten gespeichert

Schreibe aus dem PC die Modelle in den Sender

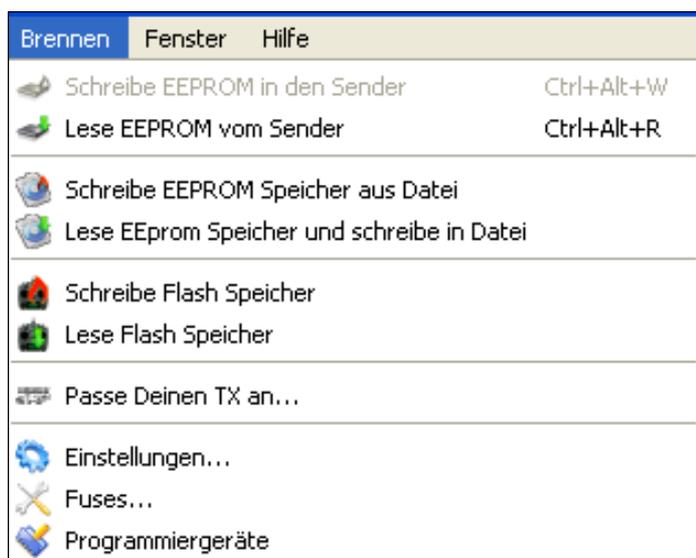
Lese aus dem Sender die Modelle aus und übertrage sie in den PC

Dazu muss der Sender eingeschaltet sein, bevor das USB angesteckt wird!

Die SD-Karte dient als Modell-Archiv

mit Backup und Restore ins EEPROM des Senders und hat mindestens 6-8

Unterverzeichnisse für Modelle, Sounds, Log, Bmp, usw.



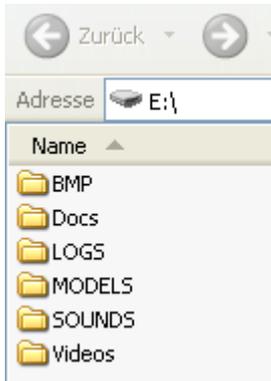
Modelle vom PC in den Sender übertragen und zurück

Sender einschalten, Gas und Schalter quittieren, dann erst USB verbinden, nicht vorher!

Am PC melden sich 2 Laufwerke, das ist ok, die brauchen wir aber jetzt nicht, wieder schließen.

E: das ist die SD-Karte mit den div. Unterverzeichnissen

F: das ist das EEPROM dort sieht man eine einzelne TARANIS.BIN Datei. Das ist die Modelldatei!

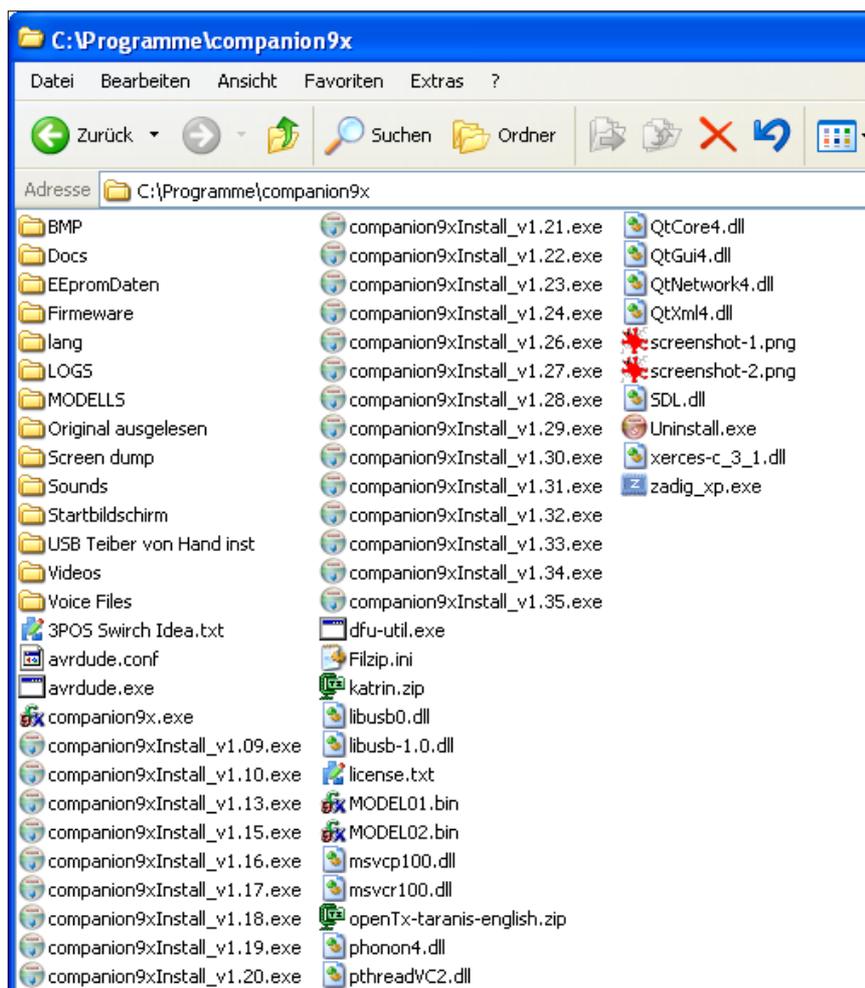


Jetzt aber nicht versuchen diese *.bin Datei einfach zu kopieren, denn damit kann niemand was anfangen!

Nur companion9x kann das EEPROM-Format V212.....V217 richtig interpretieren!

Die Modelldatei im Sender muss TARANIS.BIN heißen

Companion9x starten, zur Kontrolle, das sind meine Unterverzeichnisse unter Companion9x



Hintergrund: Modelldateien von und zu Sender übertragen

Achtung: Ab openTx2.0 mit einem Bootloader geht das anders, siehe Teil E

Zuerst Sender einschalten, Gas und Schalter Warnung weg drücken, bis Sender ganz normal voll hochgelaufen ist.

Erst jetzt das USB Kabel einstecken, nicht vorher!

Dann melden sich 2 Laufwerke z. B. mit E\:\ und F\:\ oder F\:\ und G\:\ oder G\:\ und H\:\ je nach PC

Diese 2 LW schließen wir, die brauchen wir vorab mal nicht

Aber damit ist der Sender als z.B. LW E\:\ und F\:\ dem PC intern bekannt, sonst geht gar nichts!

Erst jetzt kommen wir mit Companion9x auf den Sender und können auf die Modelle zugreifen!

Ein Laufwerk enthält nur eine einzige Datei, TARANIS.BIN, da sind die Modelle enthalten!

Das andere Laufwerk ist die SD-Karte mit den Unterverzeichnissen BMP, SOUNDS, LOG, usw.

Companion9x verwaltet *.eepe Dateien liest und schreibt in den Sender diese TARANIS.BIN d.h. Companion9x erzeugt ein für den Sender lesbares Format!

Dann kann man aber auch direkt ein Backup aus dem Sender machen.

Das liest diese TARANIS.BIN aus und legt sie unter einem freien Namen auf die Festplatte des PC Auch diese Datei kann Companion9x wieder lesen und anzeigen.

Und es muss die EEPROM Version 215 (bzw schon 216) sein damit das unter companion9x vom und zum Sender richtig läuft.

Companion9x kann auch selbständig die EEPROM Formate umschreiben von V212, V213, V214 auf die aktuelle V215 bzw V216

Hinweis: Option Joystick in der Firmware aktiviert / deaktiviert

- Ohne Option Joystick, kann man direkt per USB Kabel verbinden ohne vorher im Sender in den Bootloadermodus zu gehen und es melden sich die 2 Laufwerke.

- Mit Option Joystick, **muss** im Sender der Bootloadermodus aktiviert werden, dann erst per USB verbinden und es melden sich die 2 Laufwerke.

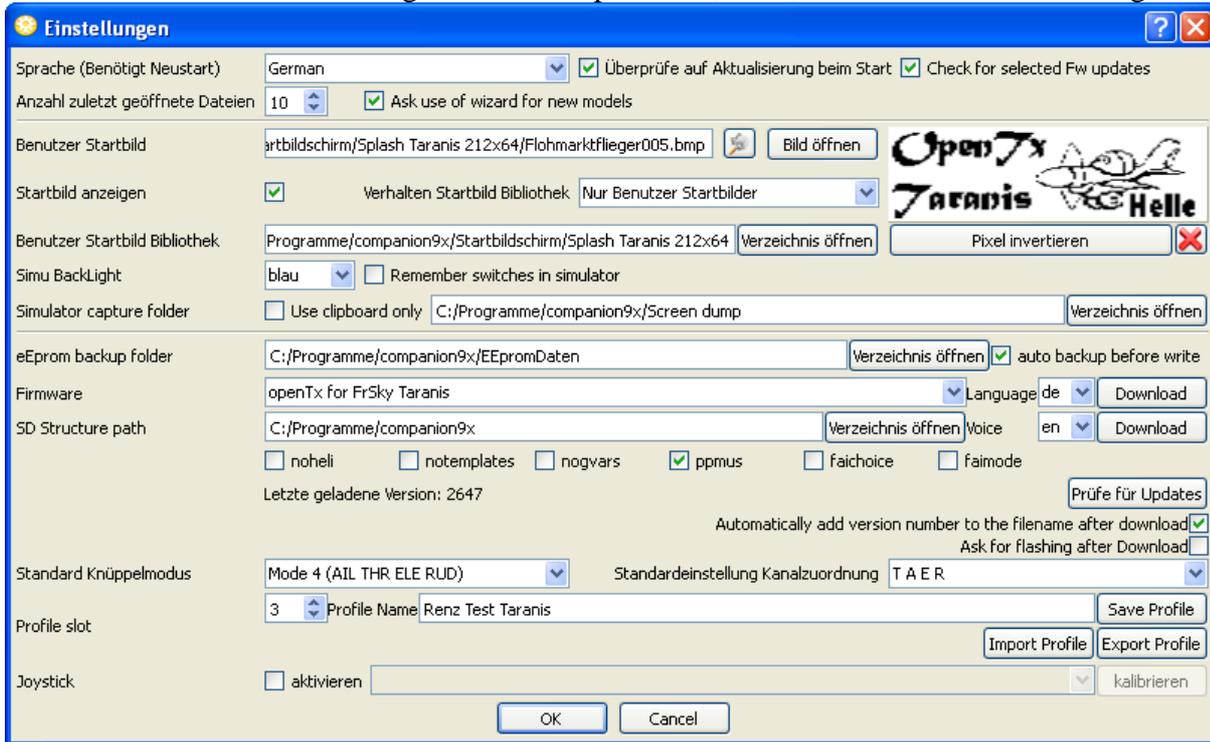
Varianten des USB Verbindungsaufbau mit Taranis X9D, X9E

1. Taranis ausgeschaltet => USB anschließen => STM32 Bootloaderfunktion aktiv (DFU, Zadig)

2. Taranis eingeschaltet => USB anschließen => Joystickfunktion aktiv

3. Taranis Trims halten, einschalten => USB anschließen => SD-Datenträgerfunktion, Update, Boot

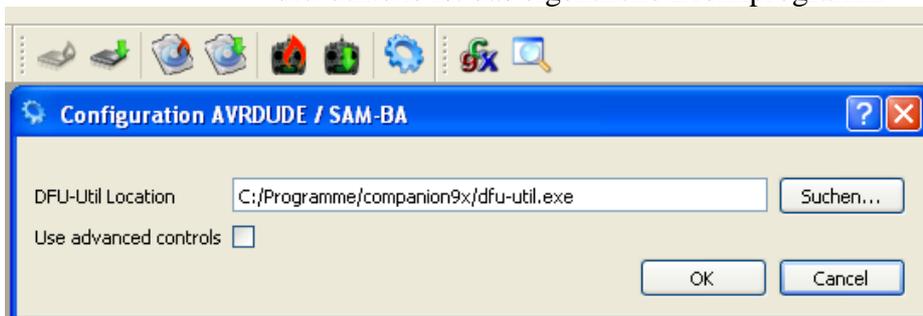
Und das sind meine Einstellungen unter Companion9x, nur mal zur Kontrolle und Vergleich



Bitte mal vergleichen und auf eigene Verzeichnisse anpassen.

Im Blauen Zahnrad muss das eingetragen sein:

dfu-util.exe ist das eigentliche Brennprogramm



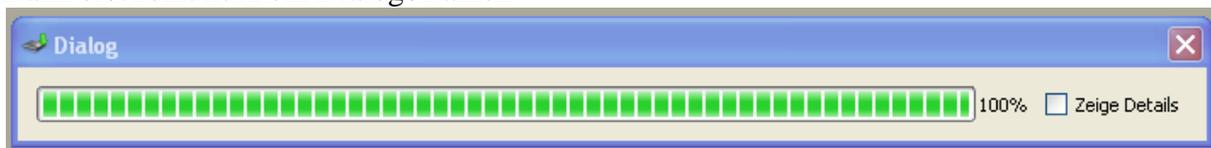
Hinweis:

Gelegentlich hörte ich das der Treiber dfu-util.exe Ärger macht, denn der Virenwarner meldet falsch und schmeiß diese Datei raus. Das ist aber das Brennprogramm!

Dann unter Companion9x in der Zeile **Lese EEPROM vom Sender die Modelle** auslesen



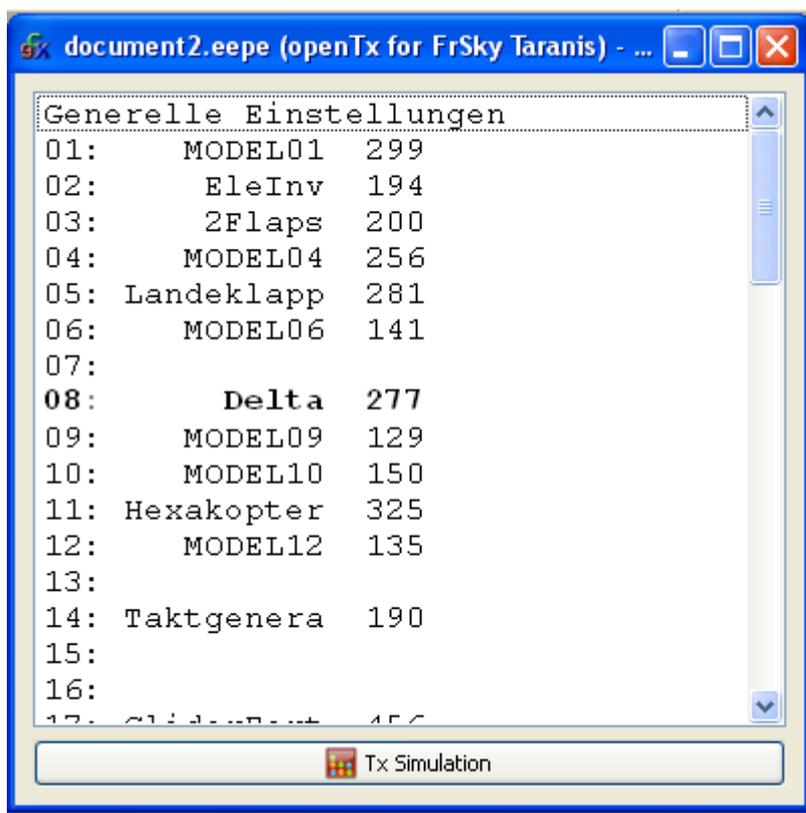
Dann erscheint kurz ein Dialog /Balken



In 2-4s sind die Modelle aus dem Sender ausgelesen.

Und alle Modelle erscheinen im Fenster und als Datei

Bei **Dateinamen etwas aufpassen**, damit sie nicht überschrieben werden.
Da sie umbenannt werden und einen neuen Laufindex erhalten.



Damit hat es sich auch schon, mehr ist nicht.

Rückwärts geht es dann genauso.

Modelle in den Sender Übertragen: **Schreibe die Modelle in das EEPROM des Sender**



Never touch a running System!

Nicht verwechseln mit OpenTx flashen, also Sender Betriebssystem überschreiben!

Das ist was ganz anderes!

Taranis mit andere OpenTx-Firmware überschreiben

(für Windows) das ist weiterhin gültig auch für openTx V2.00

Achtung: Für openTx2.0 siehe Teil E, Firmware und Bootloader installieren

Das Betriebssystem OpenTx im Flashspeicher des Senders kann man überschreiben
Dazu braucht man zuerst einen USB-Treiber: zadig_???.exe den man hier zum Download findet:
Für [Windows XP](#) Zadig_xp.exe oder für [Windows Vista and later](#) Zadig.exe

Wichtig: Der Treiber **muss** ins gleiche Verzeichnis geladen werden wie companion9x
Als normal im C:\Programme\companion9x

Einmalig den Zadig USB-Bootloader-Treiber einrichten

Für die Erstinstallation des Zadig-Treibers muss der Ablauf genau eingehalten werden
damit der Zadig- Treiber den Taranis-Sender sofort erkennt.

Sender ausschalten, nicht einschalten!!

PC und Sender mit USB Kabel verbinden, jetzt erst Programm Zadig_XP.exe ausführen,
dazu sind **Administratorrechte** nötig!

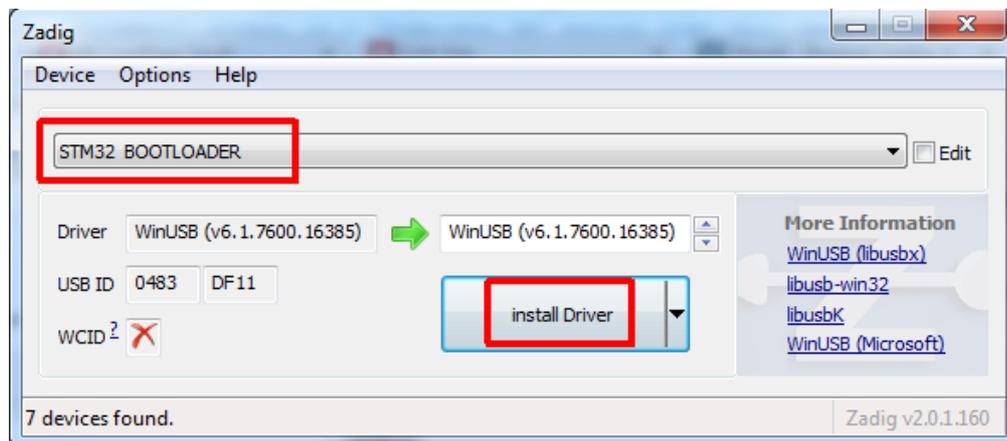
Dann sollte der Zadig-Treiber den STM32 Bootloader im Prozessor des Senders sofort gefunden haben,
ansonsten mal die lange Zeile durchsehen,

→je nach PC kann das aber auch mal 1-2 min dauern!

Mit Install Driver wird die Verbindung fertiggestellt und bestätigt.

Das Programm Zadig kann verlassen werden, braucht man nicht mehr.

Damit ist die USB - Verbindung vom PC zum Sender grundsätzlich mal eingerichtet!



Unter Win XP, Systemsteuerung, System, Hardware, Geräte Manager, USB Device
muss der STM32Bootloader zu finden sein, wenn der Sender am USB angesteckt ist.

Für MAC geht es anderst! Siehe openTx und 9xforums

Neue Firmware in den Sender flashen Schritt für Schritt

Achtung: Für openTx2.0 siehe Teil E, Firmware und Bootloader installieren

Das wird ganz genau so wie bisher bei Th9x oder 9XR mit companion9x ausgeführt.

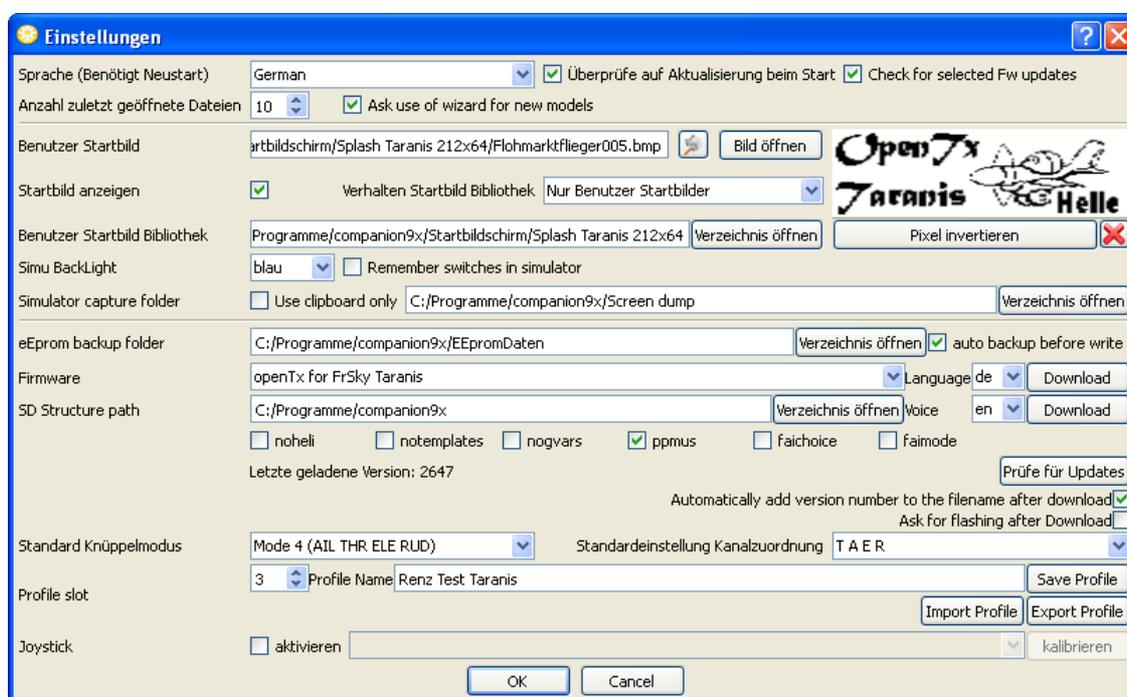
Will man die Sprache ändern, einen anderen Startbildschirm, eine andere Version mit zusätzliche Funktionen von openTx in den Sender laden, dann muss der **Flashspeicher** des Sender **überschrieben** werden.

Das hat nichts mit den 60 Modellspeichern zu tun, die liegen im **EEProm** und werden gebrannt.

Software vorbereiten mit Companion9x:

Datei, Einstellungen, die passende Firmware OpenTx for FrSky Taranis, die Sprache und weitere Optionen auswählen (ppmus).

Dann Download und companion9x stellt aus dem Internet die passenden Programmteile zusammen und speichert sie ein einem Unterverzeichnis von companion9x.



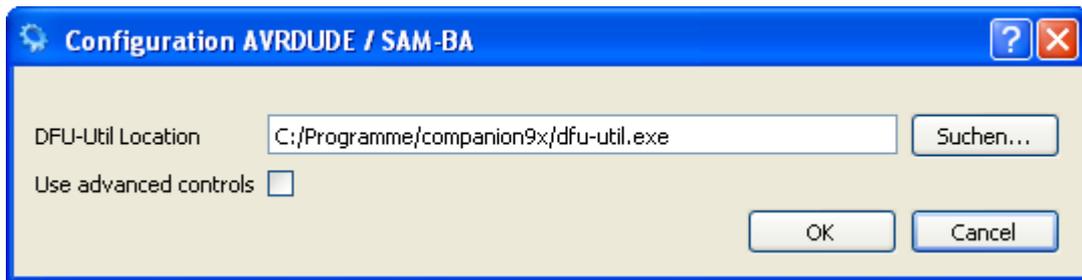
Beispiel für die Einstellungen, aber auf eigene Unterverzeichnisse anpassen!

Dann die Funktion Brennen, Schreibe Flash Speicher aufrufen

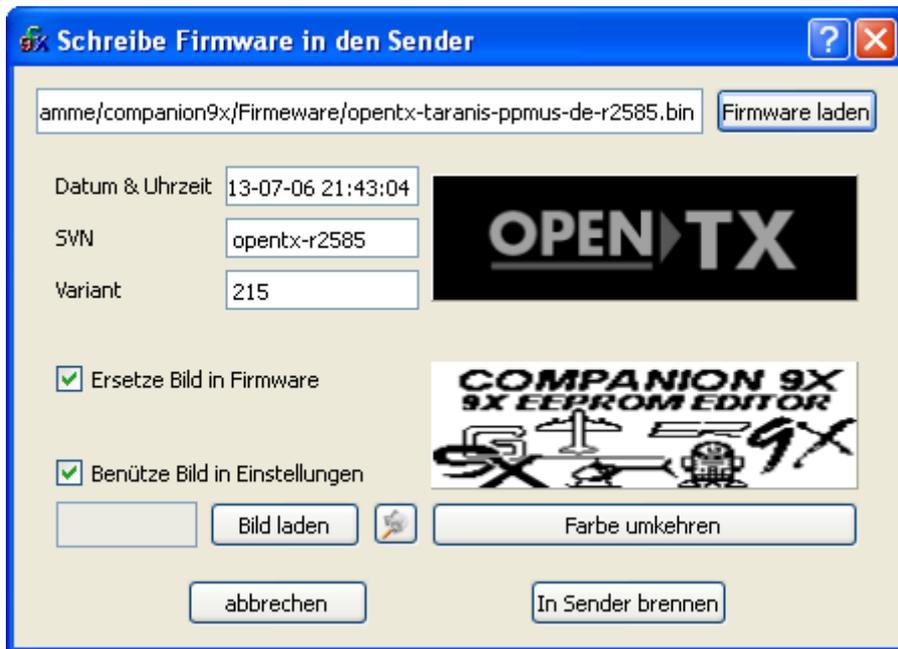


openTx für Taranis Anleitung Deutsch

Für die Sender TH9x und 9XR braucht man zum Flashen das Programm AVRdude.exe
Für **Taranis** wird das Programm **dfu-util.exe** oder ähnlich zum Flashen hier eingetragen.



Es erscheint die eigentliche Auswahl zum Flashen mit ein paar Infos und Menüs



Das eigentliche Programm **OpenTx für Taranis** befindet sich in Dateien mit dem Typ *.bin und den entsprechenden Softwareständen / Infos. r2588 mit EEprom Variante 215
Dann kann man noch einen anderen/eigenen Startbildschirm auswählen.

Beispiel:

```
opentx-taranis-ppmus-de-r2491.bin  
opentx-taranis-ppmus-de-r2523.bin  
opentx-taranis-ppmus-de-r2585.bin  
opentx-taranis-ppmus-de-r2588.bin
```

Achtung:

Sender muss ausgeschaltet sein, dann erst mit USB-Kabel zum PC verbinden!

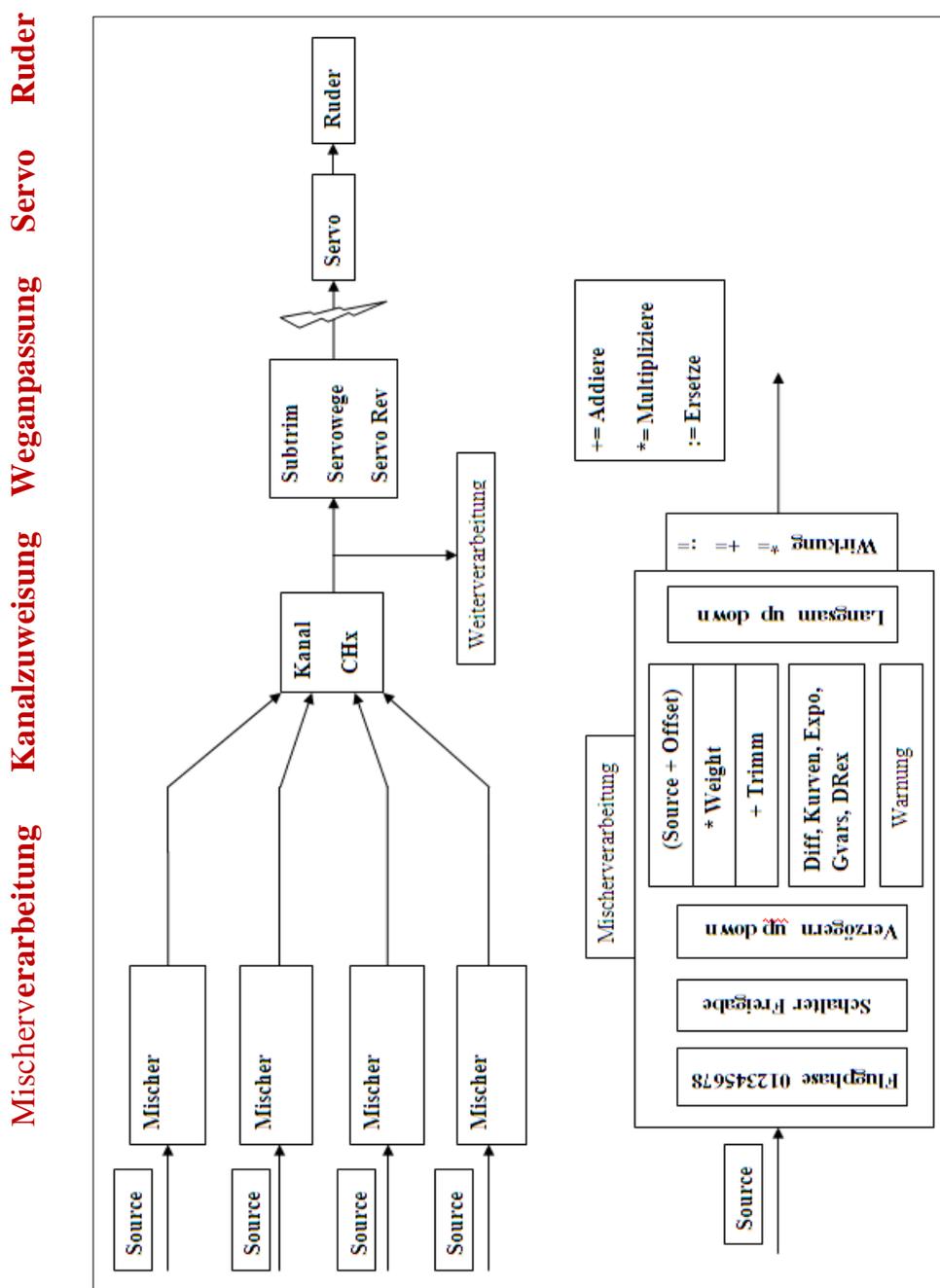
Mit **Schreibe Flash Speicher** wird der eigentliche Flashvorgang gestartet, ein Fortschrittsbalken, dann ein überprüfen und in wenigen Sekunden ist die neue Software auf dem Sender.

USB-Kabel abziehen, Sender einschalten, fertig!

Teil C Modelle mit Companion9x programmieren

Prinzipdarstellung der Mischerverarbeitung

Beliebig viele Mischerzeilen werden auf einen Kanal zusammengeführt und verrechnet. Das Ergebnis kann auch als Zwischenergebnis / Vorverrechnung für die Weiterverarbeitung mehrfach genutzt werden. **Mischerzeilen immer so eingeben und verrechnen, dass positive Mischerwerte zu positiven Ruderbewegungen (nach oben bzw. rechts) führen. Erst dann, wenn alles verrechnet ist und dem Kanal zugewiesen ist, im Servomenü die Laufrichtung der Ruder so anpassen, dass die Ruder „richtig“ laufen! Würde man das schon in den einzelnen Mischerzeilen anpassen handelt man sich zu viele Denkfehler ein.**

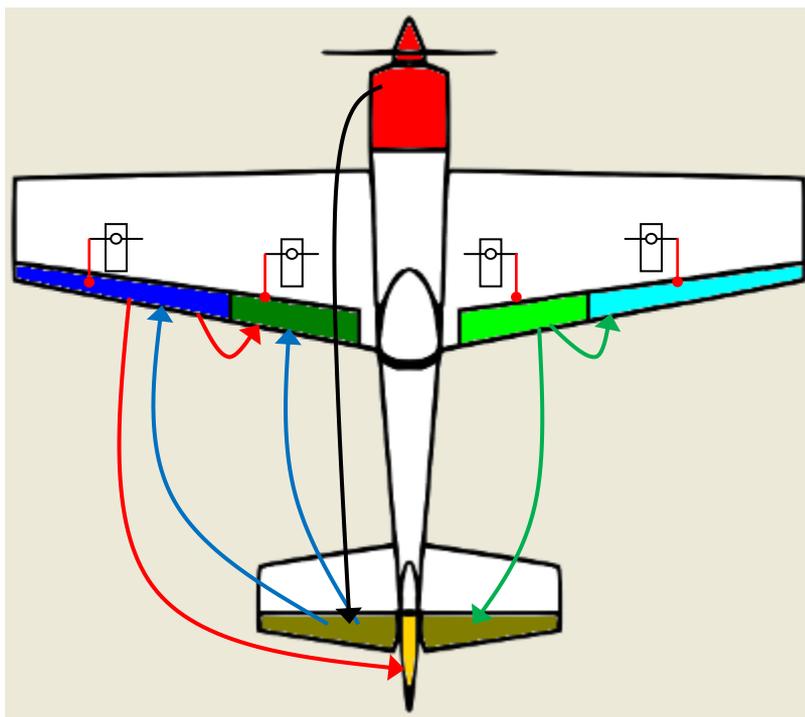


Übersicht der Mischer Quelle und Ziel

Nochmal: Es gibt keine fertigen Mischerprogramme, man braucht sie nicht!
 Jede einzelne Mischerzeile ist ein eigener vollständiger Universalmixer der alles kann.
 Dies hier ist nur mal eine Übersicht was von und an wenn gemischt werden kann.

Mischerprogramme Übersicht Motormodelle

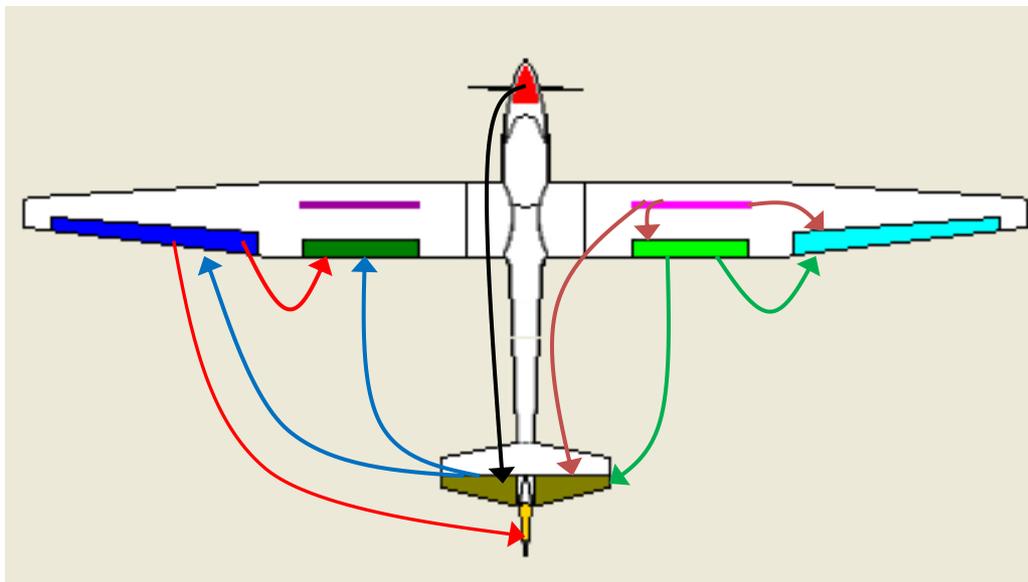
	Anteil		
Querruder-Differenzierung	%		
Wölbklappen-Differenzierung	%		
Differenzierungsreduktion	%		
Quelle	Ziel	Anteile	Schalter
1. Querruder	→ Seitenruder	%	
2. Querruder	→ Wölbklappe	%	
6. Höhenruder	→ Wölbklappe	%	%
7. Höhenruder	→ Querruder	%	%
8. Wölbklappe	→ Höhenruder	%	%
9. Wölbklappe	→ Querruder	%	%
10. Motor	→ Höhenruder	%	%



Dazu kommen noch Kombinationen und unterschiedliche Flugphasen wie, Start, Landung, Gleitflug, Acrobatic,

Mischerprogramme Übersicht Segler

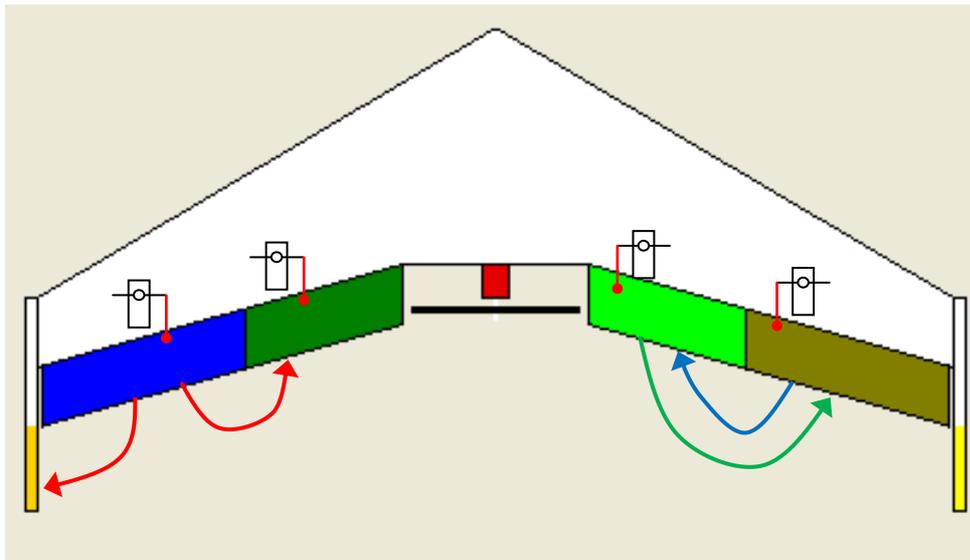
		Anteil			
Querruder-Differenzierung			%		
Wölbklappen-Differenzierung			%		
Differenzierungsreduktion			%		
Quelle	Ziel		Anteile		Schalter
1. Querruder	→ Seitenruder		%		
2. Querruder	→ Wölbklappe		%		
3. Bremse	→ Höhenruder		%		
4. Bremse	→ Wölbklappe		%		
5. Bremse	→ Querruder		%		
6. Höhenruder	→ Wölbklappe		%		%
7. Höhenruder	→ Querruder		%		%
8. Wölbklappe	→ Höhenruder		%		%
9. Wölbklappe	→ Querruder		%		%
10. Motor	→ Höhenruder		%		%



Dazu kommen noch Kombinationen wie Butterfly oder Krähenstellung und unterschiedliche Flugphasen wie, Start, Landung, Gleitflug, Acrobatic,

Mischerprogramme Übersicht Delta

Anteil			
Querruder-Differenzierung	%		
Wölbklappen-Differenzierung	%		
Differenzierungsreduktion	%		
Quelle	Ziel	Anteile	Schalter
1. Querruder	→ Seitenruder	%	
2. Querruder	→ Wölbklappe	%	
6. Höhenruder	→ Wölbklappe	%	%
8. Wölbklappe	→ Höhenruder	%	%
10. Motor	→ Höhenruder	%	%



Dazu kommen noch Kombinationen und unterschiedliche Flugphasen wie, Start, Landung, Gleitflug, Acrobatic,

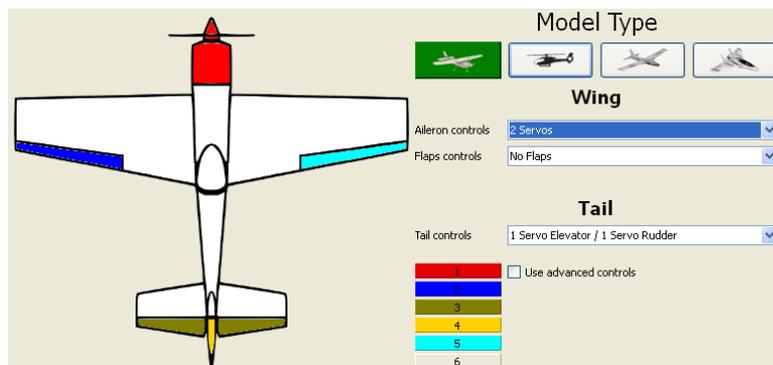
Beispiel: Fertiges Programm für ein Motormodell

Positive Knüppelwerte sollen Ruder nach oben oder nach rechts bewegen
 Negative Knüppelwerte sollen Ruder nach unten oder nach links bewegen

Querruderknüppel nach rechts und halten, Knüppel liefert positive Werte (+100%)
 Quer1 rechts, nach oben (+100%) und gleichzeitig Quer2 links, nach unten (-100%)

Kanalbelegung

- CH1 Motor
- CH3 Seite
- CH4 Höhe
- CH6 Fahrwerk langsam Ein/Aus
- CH2 Quer1 rechts
- CH5 Quer2 links



DR/Expo: Höhe und Quer per Dualrate/Expo umschaltbar Wege 100%/75% , Expo 35%

Rud	Weg (100%) (Seite)
Ele	Weg (100%) Expo (+35%) Schalter (SC↑)
	Weg (75%) Expo (+35%) Schalter (SC↓)
Thr	Weg (100%) (Gas)
Ail	Weg (100%) Expo (+35%) Schalter (SC↑)
	Weg (75%) Expo (+35%) Schalter (SC↓)

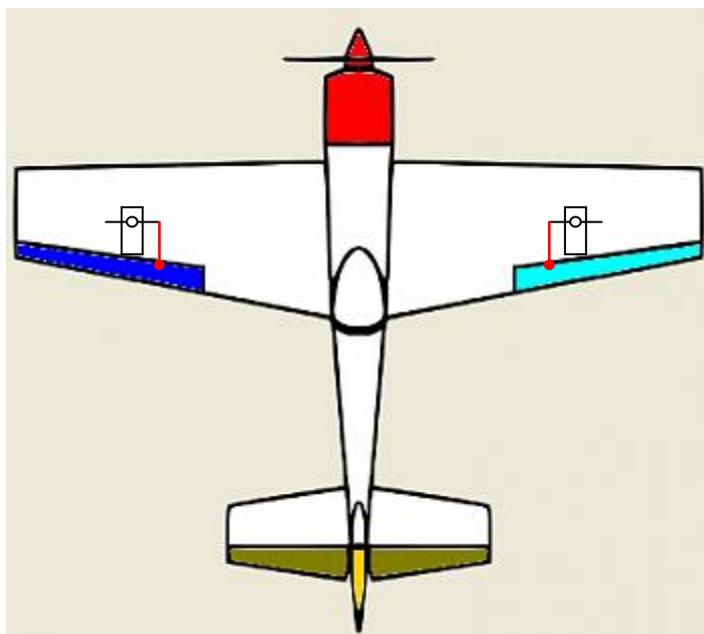
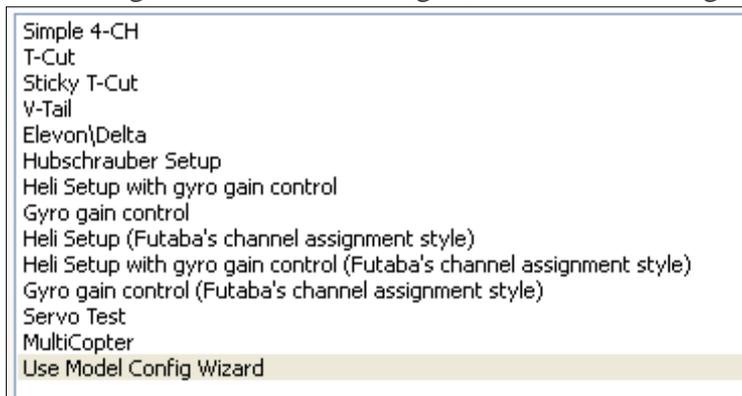
Mischer: 2 Querruder mit 30% Differenzierung kombiniert mit Querruder auch als Landeklappen in 3 Stufen 0% -25% -45% schaltbar

CH01	(+100%) Thr (Gas)
CH02	(+100%) Ail Diff (30%) (Quer1)
	(0%) MAX Schalter (SA↑) (Landel)
	(-25%) MAX Schalter (SA-) (Landel)
	(-45%) MAX Schalter (SA↓) (Landel)
CH03	(+100%) Ele (Höhe)
CH04	(+100%) Rud (Seite)
CH05	(-100%) Ail Diff (30%) (Quer2)
	(0%) MAX Schalter (SA↑) (Lande2)
	(-25%) MAX Schalter (SA-) (Lande2)
	(-45%) MAX Schalter (SA↓) (Lande2)
CH06	(+100%) SBLangsam/u1.2:d1.5) (Fahrwerk)
CH07	
CH08	

Beispiel: Schritt für Schritt Programmierung eines Modell mit den Mixern

Teil 1: Wir beginnen immer bei den Mixern.

Companion9x starten, neues Modell anlegen, Doppelklick darauf und dann ganz links unter Vorlagen, use Model Config Wizard



Motormodell auswählen und zusammenstellen mit 2 Querruder

Das Ergebnis sehen wir hier.

Voreinstellungen GQHS weil ich Knüppel Mode 4 fliege und von meiner Graupnerbelegung ausgehe

CH1 Motor
CH2 Quer 1
CH3 Höhe
CH4 Seite
CH5 Quer 2

Übernehmen und anwenden, dann ist ein neues Modell in den Mixern eingetragen.

Die Grundeinstellung der Mischer ist jetzt so:

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail (AIL)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD)
CH05	(+100%) Ail (AIL2)
CH06	
CH07	
CH08	

Beachte: Die Voreinstellungen bringen **beide** Quer mit **+100% Weight**

So und jetzt mal simulieren, aber kein Servo Reverse machen! Alles so lassen.
Wie man sieht laufen CH2 und CH5 gleich, also eigentlich falsch.

Das könnte man jetzt per Servo Reverse Kanal5 umdrehen, dann wäre es richtig und wir wären fertig.

Das machen wir aber gerade nicht!!

Ich weiß, jetzt kommt das ja aber ...

Teil 2: Querruder und Querruder-Differenzierung

Jetzt werden wir folgendes tun:

Querruder 2, CH5, links, erhält -100% weight

Also so: Ch5 -100% Ail2

Bitte simulieren! Jetzt laufen die Ruder im Simulator "richtig"!

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail (AIL)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD)
CH05	(-100%) Ail (AIL2)
CH06	
CH07	
CH08	

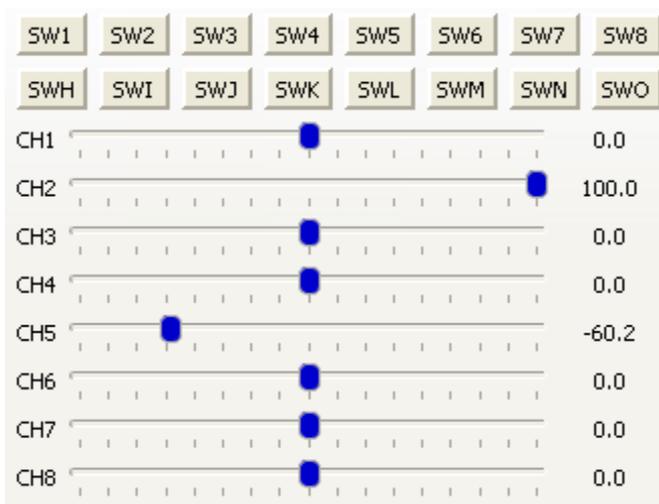
und dann machen wir eine Ruder Differenzierung
 CH2 Diff +40% CH5 Diff +40% ja, beide Werte positiv!
 Das sieht dann so aus, beide Ruder erhalten positive Werte!!

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail Diff (40%) (AIL)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD)
CH05	(-100%) Ail Diff (40%) (AIL2)
CH06	
CH07	

Wieder simulieren, die Ruder laufen richtig
 und die Differenzierung wirkt auch richtig!

Knüppel Quer voll rechts geben
 CH2 geht auch +100% CH5 geht auf -60%

Knüppel Quer voll links geben
 CH2 geht auf -60% CH5 geht auf +100%



Differenzierung ist eigentlich schon der erster "Spezialmischer", das sagt nur keiner.

Das Ruder das nach unten geht, soll weniger nach unten gehen als das Ruder, das nach oben geht.
 Das wird so berechnet: Wenn Ruder $r < 0$ dann $r = r - (r * d)$

Nur wenn der Ruderwert negativ ist, dann wird mit Differenzierungsanteil verrechnet

$$\text{Ruder} = \text{Ruder} - (\text{Ruder} * \text{Diff}) \quad -100\% - (-100\% * +40\%) = -60\%$$

Das sieht dann im Endergebnis so aus:

$$-100\% + 40\% = -60\% \text{ und das Ruder geht nur noch zu } -60\% \text{ runter anstatt zu } -100\%$$

Das andere Ruder geht weiterhin seinen eingestellten Weg nach oben hier +100%

Damit wird das negative Wendemoment ausgeglichen, aus einer Fassrolle wird eine saubere Rolle

Soweit mal ok?

Teil3: Querruder als Landeklappe in 2 Stufen mit einem Schalter 0% -20% -30%

Dazu gehen wir methodisch einen Schritt zurück und schalten die Differenzierung bei beiden Kanälen wieder auf 0 damit man was sieht!

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail (AIL)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD)
CH05	(-100%) Ail (AIL2)
CH06	
CH07	
CH08	

Merke: Negative Werte sollen ein Ruder nach unten bewegen!

Wir müssen also in CH2 und in CH5 je eine Zeile dazumischen also in Zeile für CH2 rein, markieren, dann Rechtsklick, auf Addieren Eine leere Zeile erscheint und das Mischermenü

Als Quelle nehmen wir MAX, das liefert konstant +100% als weight -20% (soll nach unten gehen) und als Schalter SA in -- Stellung

Genau das gleiche machen wir bei CH5 Das sieht dann so aus, und wir simulieren wieder

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail (AIL)
	(-20%)MAX Schalter (SA-)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD)
CH05	(-100%) Ail (AIL2)
	(-20%)MAX Schalter (SA-)
CH06	
CH07	

Wenn ich den Schalter SA auf down habe laufen die beide Querruder normal

Wenn ich den Schalter auf SA-- legen laufen beide erst mal auf -20% als Landeklappe und von dieser Stellung dann das Querruder wieder "normal" hin und her.

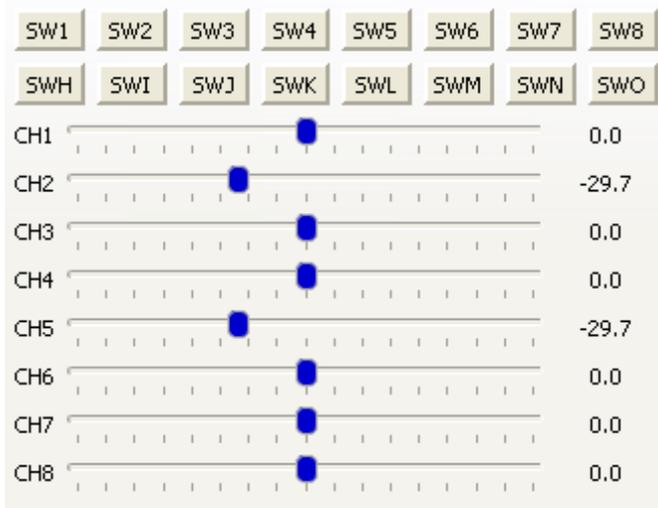
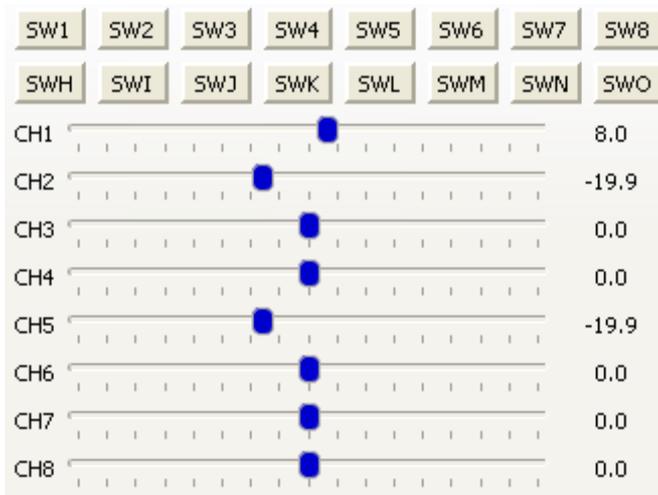
Damit haben wir eine erste Stellung der Landeklappen schon mal fertig.

openTx für Taranis Anleitung Deutsch

Das ganze wiederholen wir jetzt bei CH2 und CH5 aber mit dem Wert MAX weight -30% und SAup und wir erhalten eine zweite Stufen Landeklappe

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail (AIL)
	(-20%) MAX Schalter (SA-)
	(-30%) MAX Schalter (SA↓)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD)
CH05	(-100%) Ail (AIL2)
	(-20%) MAX Schalter (SA-)
	(-30%) MAX Schalter (SA↓)
CH06	
CH07	

Bitte simulieren!



Teil 4 Mischer von Quer auf Seite,

d.h. wenn ich Querruder rechts gebe soll das Seitenruder auch nach rechts

Merke: Positive Werte sollen ein Ruder nach oben bzw. nach rechts bewegen

Quelle ist der Querruderstick Ail, Ziel ist CH4 das Seitenruder, Anteil +25%

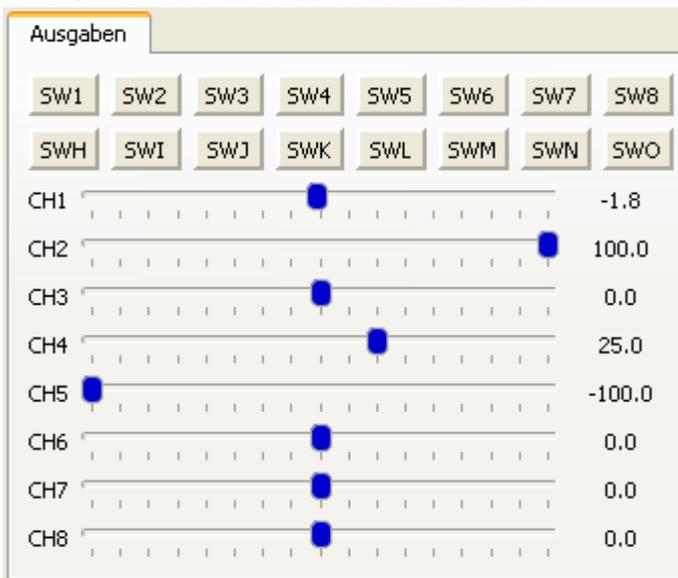
Wir beginnen mit der Grundeinstellung damit man was sieht ohne Differenzierung, ohne Landeklappen also alles auf "normal" mit CH2 Quer rechts +100% CH5 Quer links mit -100%

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail (AIL)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD)
CH05	(-100%) Ail (AIL2)
CH06	
CH07	
CH08	

Dann gehen wir auf CH4 Seitenruder, markieren, und mit Rechtsklick, Addieren wir eine Zeile dazu, Quelle Ail Anteil +25%, und fertig ist der Mischer Quer auf Seite

CH01	(+100%) Thr (THR)
CH02	(+100%) Ail (AIL)
CH03	(+100%) Ele (ELE)
CH04	(+100%) Rud (RUD) (+25%) Ail
CH05	(-100%) Ail (AIL2)
CH06	
CH07	

Bitte Simulieren! Das sieht dann so aus.



Soweit mal diese 4 Teile, hoffe das ist ok

Wie man sieht und das ist entscheidend:

Keine Servos vorab im Simulator invertieren und die Berechnungen/Mathematik bleiben immer richtig!

Wer jetzt Lust hat kann das alles zusammen setzen

Differenzierung, Landeklappen, Quer auf Seite usw.

Ein Problem kommt auf:

Der Elevator, Höhenruderknüppel liefert beim Ziehen negative Werte, verletzt also unser Prinzip!

Das können wir aber auch nicht einfach im DR/Expo Menü invertieren, denn das gibt bei Weight nur 0 bis +100% aber keine -100% bis +100%! .Im Mischer geht das, dort haben wir aber keine Expofunktion die man einfach per Parameter einstellen kann. Was tun?

Dazu gibt es 4 Lösungen:

Elevator Stick mit negativer 3-Punkt Kurve invertieren, das geht ganz schnell, falls wir Expo brauchen, müssen wir diese aber per Vielpunkt-Kurve erzeugen, gut machbar

Alle betroffene Servos invertieren, das ist am Schlechtesten, da passt dann nichts mehr

Poti Höhenruder umlöten, das wäre grundsätzlich die beste Lösung, aber öffnen, löten, na ja

Was immer geht:

Den Elevator in einem freier Kanalmischer vorverarbeiten, als Hilfsvariable weiterverwenden

und umbenennen z.B. in EleInv Ele → -100% → EleInv Dann anstatt Ele immer EleInv benutzen

Vorteil: Man hat weiterhin ganz einfach beides,

Expo aus Menü DR/Expo, Weight -100% und alle weiteren Mischerfunktionen

Nachteil: daran denken, anstatt Ele verwendet man jetzt CH10 bzw EleInv

Beispiel: Kanal CH10 Mischer, Quelle Ele, Mischer Weight -100% Name EleInv

```
CH09
EleInv (-100%) Ele (ELEInv)
CH11
```

Anwendung:

```
CH01 (+100%) Thr (THR)
CH02 (+100%) Ail Diff (40%) (AIL)
CH03 (+100%) CH10 (EleInv)
CH04 (+100%) Rud (RUD)
CH05 (-100%) Ail Diff (40%) (AIL2)
CH06
```

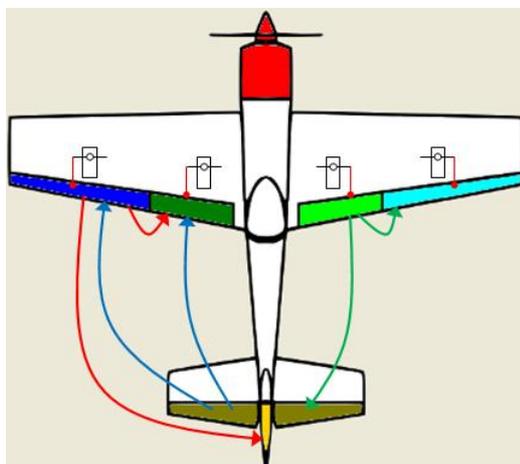
Somit passt auch das Elevator/Höhenruder immer in unser Konzept:

Positive Bewegungen führen zu positive Ruderbewegungen!

Zu dem heiligen Thema: Ein Querruder Servo invertieren anstatt beide positiv

Beim Simulieren gibt es etwas grundsätzliches zu beachten.

Das kommt daher, dass normal die 2 Querruderservos die Ruder gespiegelt ansteuern, also einmal Ruderhorn links, einmal Ruderhorn rechts, d.h. bei gleicher, positiver Ansteuerung der beiden Querruder-Servos laufen die Ruder gegenseitig und somit passend (oder aber beide falsch!).



Servos steuern gespiegelt die Ruder an.,
Das wirkt aber wie eine Servoinvertierung

Das ist bei allen Sender-Herstellern historisch bedingt
und wurde beibehalten.

Alles Weitere wird dann in den „Spezialmischern“ im
Hintergrund versteckt, denn an der Mathematik kommen auch
sie nicht vorbei.

Kein Simulator der Welt kann die reale Welt richtig nachbilden.

Welches Servo ich auch im Simulator vorab invertiere, es ist zu 50% der Falsche!

Je nach Einbaulage, Seite der Ruderanlenkung oder auch Hersteller laufen sie gerade anders rum.

Deshalb gilt hier bei den Einführungen und Beispielen:

Keine Servos invertieren, sonst kommt ihr in einen riesen Schlamassel wenn
vielfache Mischer unterschiedlich eingreifen.

Das kann ich noch ausführlich erläutern.

Egal wie ihr es macht, es ist immer falsch am Simulator Servos zu invertieren!

Am Modell muss ich natürlich schon Servos invertieren damit das Ruder richtig läuft!

Das hat aber nichts mit der Simulation zu tun!

Servos passen nur die Berechnungen auf die reale Welt an. Sie dürfen nicht die Berechnung rückwirkend beeinflussen.

Ich weiß genau jetzt trete ich den Glaubenskrieg los!

Ein bisschen weitergedacht:

ich gebe auf beide Quer +100%

ich invertiere Ch5 damit es richtig läuft

dann mische ich auf beide Querruder was dazu

die beiden Landeklappen sollen auf -25% gehen

Jetzt muss ich aber auf
Quer1 CH2 -25% geben und auf Quer2 CH5 +25%
damit es richtig läuft, soweit mal ok, machbar, gut, passt, fliegt

Jetzt übertrage ich das programmierte Modell in den Sender
und "Mist" die Ruder laufen falsch rum, also im Sender die Servos
in den Laufrichtungen angepasst bis es ok ist.
Damit ist alles gut, einfliegen, trimmen, ok passt, alles gut

Jetzt lade ich das Modell in den Simulator zurück weil ich etwas anpassen will,
und "Mist" da läuft ja alles verkehrt, wie das denn?

Merkt ihr was passiert ist?

Egal was ihr vorher oder nachher mit Servo invers macht, es ist immer falsch!

Deshalb mein Tip: (Es muss sich ja keiner daran halten, macht es wie ihr wollt!)
Servoinvers im Simulator immer auf normal lassen und Modell programmieren, simulieren
Modell in den Sender laden dort die Servoinvers die ich brauche bis es ok ist.

----- fliegt ihr schon oder programmiert ihr noch? -----

Falls ich ein fertiges Modell in den Simulator zurücklade, dann
Servoinvers merken und aufschreiben!
alle Servoinvers auf Norm setzen,
Mischer einfügen, anpassen, testen, simulieren bis alles ok
Die Servo invers wieder setzen und zurück ins Modell und alles passt.

Aufschrei!! Hilfe, was soll das denn!!

Kein Simulator der Welt kann die Welt real vorab nachbilden.

Companion9x simuliert incl. Servo Limits, das ist gut und gleichzeitig schlecht

In sich gehen, nachdenken, wirken lassen!

Ich weiß, wenn die ersten eine Taranis in den „erfahrenen“ Händen halten
kommen genau deshalb die Hilferufe:

Das ist alles ein Murks, das geht ja gar nicht, das läuft verkehrt herum,
das muss aber anders programmiert werden, da ist aber ein Fehler drinnen,
bei Graupner/Futaba macht man das aber anders, usw.

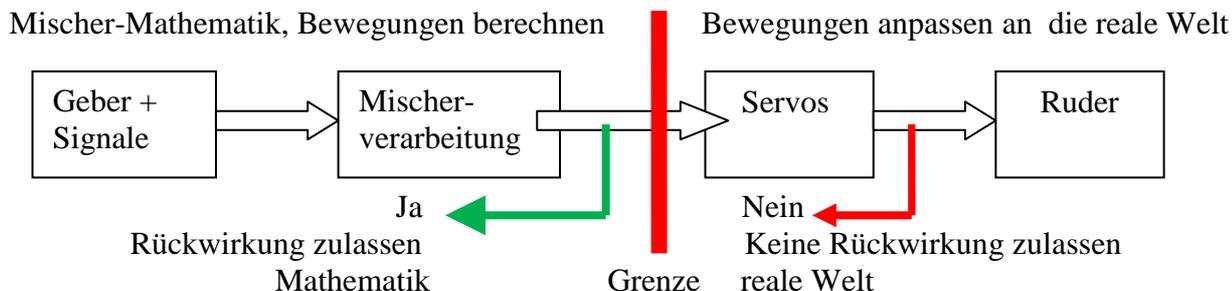
Der Fehler liegt meist zwischen den Ohren!

In der echten Fliegerei ist es genau so, es gibt 3 goldene Regeln:

1.
Ein positives Knüppelsignal muss ein Ruder nach oben bzw. nach rechts bewegen
Ein negatives Knüppelsignal muss ein Ruder nach untern bzw. nach links bewegen
2.
Das erste Querruder ist das rechte Querruder
Das zweite Querruder ist das linke Querruder
3.
Die Umsetzung in die reale Ruderbewegung ist zweitrangig
ob mit Zügen, Hydraulik, Motoren, das darf die Berechnungen nicht rückwirkend beeinflussen!

Wenn Ihr diese 3 Dinge beachtet werdet ihr nie mehr ein Problem mit Mischerverrechnungen haben!

Positive Mischerausgangswerte gehen immer nach oben bzw. nach rechts!
Negative Mischerausgangswerte gehen immer nach unten bzw. nach links!



Hier mal 2 Beispiel nebeneinander, mit 4 Mixern auf Querruder.
Beide Beispiele funktionieren einwandfrei

Bitte mal vergleichen, was sind das für Mischer, wo gehen sie hin, wie weit,
was ist einfacher zu verstehen

Kein Servo Revers, Regeln gelten!
CH2 +100% CH5 -100%

CH01	(+100%)Thr (THR)
CH02	(+100%)Ail Diff(40%)(Quer1 R) (-20%)MAX Schalter(SA-)(Flap R) (-40%)MAX Schalter(SA!)(Flap R) (+25%)Rud(SeiQuer) (+30%)Ele Schalter(SB!)(HöheQuer)
CH03	(+100%)Ele(ELE)
CH04	(+100%)Rud(RUD)
CH05	(-100%)Ail Diff(40%)(Quer2 L) (-20%)MAX Schalter(SA-)(Flap L) (-40%)MAX Schalter(SA!)(Flap L) (-25%)Rud(SeiQuer) (+30%)Ele Schalter(SB!)(HöheQuer)
CH06	

Kanal 5 Servo Revers, Regeln gelten nicht!
CH2 +100% CH5 +100%

CH01	(+100%)Thr (THR)
CH02	(+100%)Ail Diff(40%)(Quer1 R) (-20%)MAX Schalter(SA-)(Flap R) (-40%)MAX Schalter(SA!)(Flap R) (+25%)Rud(SeiQuer) (+30%)Ele Schalter(SB!)(HöheQuer)
CH03	(+100%)Ele(ELE)
CH04	(+100%)Rud(RUD)
CH05	(+100%)Ail Diff(40%)(Quer2 L) (+20%)MAX Schalter(SA-)(Flap L) (+40%)MAX Schalter(SA!)(Flap L) (+25%)Rud(SeiQuer) (-30%)Ele Schalter(SB!)(HöheQuer)
CH06	

Achtung: Das sind alles nur mal Spielbeispiele mit großen Zahlen damit man was sieht.

Beispiel: Dualrate und Expo als Vorverarbeitung und Signalanpassung der Knüppel

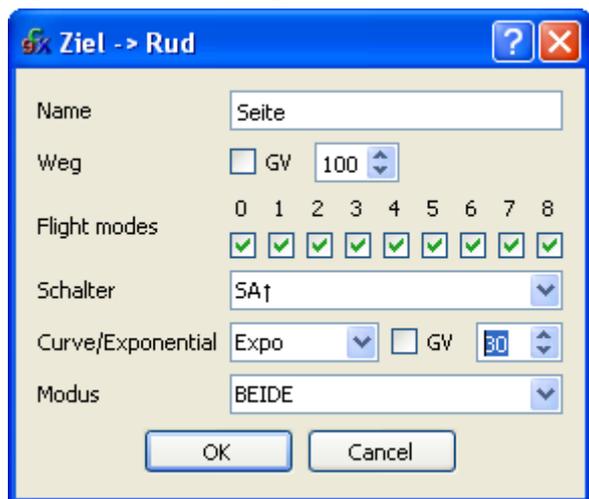
Dualrate = Wegeumschaltung für große oder kleine Ruderausschläge, meist in 2 bis 3 Stufen

Expo = sorgt um die Mittelstellung für sanfte Bewegungen, oft um die 30%-40% Expoanteil

Das sind Geraden und Kurven für die 4 Knüppelsignale.

Bevor die 4 Knüppelwerte in den Mischern als Quellen auftauchen können sie angepasst und vorverarbeitet werden.

Inputs Menü DR/Expo Menü

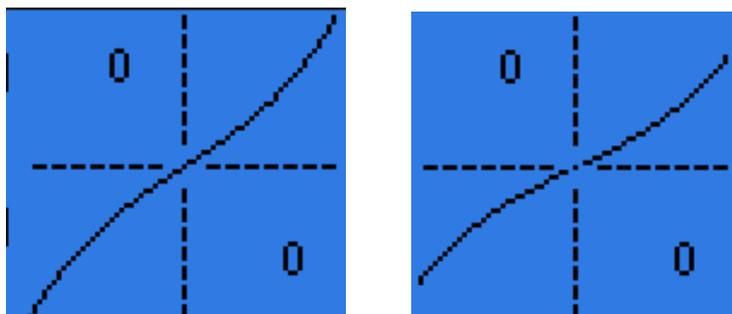


Beispiel für DR/Expo Umschaltung und Anpassung und

Beachte: **SA** als 2-Stufen Schalter verwendet mit **SA↑** und **!SA↑**

Rud	Weg (100%)	Expo (+35%)	Schalter (SA↑) (Heck)
	Weg (75%)	Expo (+35%)	Schalter (!SA↑)
Ele	Weg (100%)	Expo (+35%)	Schalter (SA↑) (Nick)
	Weg (75%)	Expo (+35%)	Schalter (!SA↑)
Thr	Weg (100%)	Expo (+50%)	(Gas)
Ail	Weg (100%)	Expo (+35%)	Schalter (SA↑) (Roll)
	Weg (75%)	Expo (+35%)	Schalter (!SA↑)

Kurvendarstellung am Sender Wegeumschaltung 100% auf 75% bei 35% Expo



Beispiel: Einfachster Deltamischer für Nuri mit 2 Servos für QR+HR gemischt

Beim Nuri werden Höhenruder und Querruder gemischt auf die 2 Servos gegeben
 Bei Höhe ziehen laufen beide Ruder nach oben,
 bei Quer rechts: QR1R nach oben, QR2L nach unten

Bezeichnungen: HRW1 HöhenRuderWeg1 QRW1 QuerRuderWeg1

DR/Expo zum Umschalten der Wege in 2 Stufen

Rud			
Ele	Weg(100%)	Expo (+35%)	Schalter(SA↑)(HRW1)
	Weg (75%)	Expo (+35%)	Schalter(SA↓)(HRW2)
Thr			
Ail	Weg(100%)	Expo (+35%)	Schalter(SB↑)(QRW1)
	Weg (75%)	Expo (+35%)	Schalter(SB↓)(QRW2)

Variante mit 3 Stufen im DR/Expo

Rud			
Ele	Weg(100%)	Expo (+35%)	Schalter(SA↑)(HRW1)
	Weg (80%)	Expo (+35%)	Schalter(SA-)(HRW2)
	Weg (65%)	Expo (+45%)	Schalter(SA↓)(HRW3)
Thr			
Ail	Weg(100%)	Expo (+35%)	Schalter(SB↑)(QRW1)
	Weg (80%)	Expo (+35%)	Schalter(SB-)(QRW2)
	Weg (65%)	Expo (+45%)	Schalter(SB↓)(QRW3)

Das ist der eigentliche Kanal-Mischer

Normaler Mischer QR + HR (im Simu zeigt HR aber negativ!) Anteile QR und HR anpassen

CH01	(+100%) Thr
CH02	(+70%) Ail (QR1R)
	(+65%) Ele (HR)
CH03	(-70%) Ail (QR2L)
	(+65%) Ele (HR)
CH04	

Kanal-Mischer, Variante mit invertiertem HR-Mischer, damit es im Simu auch „richtig“ angezeigt wird

CH01	(+100%) Thr
CH02	(+70%) Ail (QR1R) (+65%) CH10 (HRinv)
CH03	(-70%) Ail (QR2L) (+65%) CH10 (HRinv)
CH04	
CH05	
CH06	
CH07	
CH08	
CH09	
HRinv	(-100%) Ele (HRinv)
CH11	

Trick: CH10 HRinv Quelle: HR Anteil: -100%, damit Knüppel und Trimm invers Name: HRinv

Warum bei den Mixern nicht beide auf 100%?

Klar könnte man überall 100% eingeben.

Aber +100% Höhe + 100% Quer ist halt 200%, wenn du voll Höhe ziehst und dann noch Querruder brauchst (beim Landen), dann ist keine Weg mehr da, denn dein Servo ist schon voll auf Anschlag

Darum die beiden Mischer für HR und QR nur mal auf ca. 70%
die unterschiedlichen Zahlen 70% und 65% sind nur damit man sieht was von woher kommt

In der Realität sind die Werte noch kleiner, vor allem bei QR,
wenn du 3 Rollen pro Sekunde machst, dann merkst du schnell dass es zu viel ist.

Zum Erfliegen deshalb auch die DR/Expo Umschaltung, denn fast immer hat man zu viel Ruderausschlag, dann kann man umschalten.

Das kann man auch auf 2 getrennte Schalter legen für QR und HR

Beim Landen, kurz vor dem Aufsetzen braucht man aber wieder viel Höhe, bis zum ganz Durchziehen und etwas Querruder um die Richtung zu halten

So, jetzt mal ein paar Werte berechnen:
vom Geber via DR/Expo kommen 100% oder 75%
vom Mischer kommen 70% bzw. 65%

Gesamtausschlag = QR + HR
(75%*70%) + (75%*65%)
52,5% + 48,75% = das sind zusammen 101.25% das passt doch!

bzw. (100%*70%) + (100%*65%) = das sind zusammen 135%

Und was macht das in Servoweg?

Das kann man so nicht sagen, kommt drauf an was man für Servos einsetzt.
Die meisten machen max +60° und - 60°
dann die Ruderanlenkung und Übersetzung und Winkel usw.

Aber flieg doch mal mit 60° Ruderausschlag, viel Spaß!

Servos übersetzen nur die Mathematik in die reale Welt
Servowege setzt man nie ein um die Mathematik zu begrenzen,
sondern nur um die Ruder und die Getriebe vor Beschädigung zu schützen.

Servo Min/Max: um die maximal mögliche mechanische Ausschläge zu begrenzen
(aber intern wird so gerechnet: %Mischer * %Servolimit = %Servoweg)

Servo Subtrim: Um die mechanisch Nulllage, Hebel, Ruderhorn anzupassen
Das sollte immer weniger als 10-15% sein sonst bist zu weit aus der
Nulllage, die max Wege werden zu ungleich dann Ruder, Hebel,
Gestänge anpassen, kürzen, verlängern, bis Ruderhorn wieder
auf Mitte und Servo Subtrimm wieder auf 0%

Servo Reverse: Nur um die Drehrichtung anzupassen. Positiver Mischerausgang muss zu
positivem Ruderbewegung führen

Nochmal: Wege-Anpassungen macht man mit Weight/Anteil in den Mischern
oder schon vorher bei DR/Expo aber nicht mit Servo Min/Max

Am Modell die Servos so anpassen:

Querruder voll rechts geben und halten

QR1R Ruder rechts muss nach oben sonst mit Servo Reverse anpassen

und gleichzeitig geht

QR2L Ruder links muss nach unten sonst mit Servo Reverse anpassen

Segler mit 4 oder 6 Klappen 2QR 2WK 2BR programmieren

Wir arbeiten mit positiver Logik in der Mathematik der Mischerberechnungen
d.h. positive Werte gehen nach oben bzw. rechts, negative Werte nach unten bzw. links

Wir brauchen keine Servoumkehr vorab!

- 1. Querruder ist rechts 2. Querruder ist links
- 1. Wölbklappe ist rechts 2. Wölbklappe ist links

Das hat nichts mit Kanälen zu tun, die sind frei verfügbar!

Wir verwenden aber einheitliche Bezeichnungen, die Kanalbelegung hat sich so ergeben

QR	Querruder	Ail	CH2 Rechts	QR1R	CH5 Links	QR2L
WK	Wölbklappen		CH6 Rechts	WK1R	CH7 Links	WK2L
SR	Seitenruder	Rud	CH4			
GS	Motor, Gas	Thr	CH1			
BR	Bremsklappe		CH8 Rechts	BR1R	CH9 Links	BR2L
HR	Höhenruder	Ele	CH3			
HRinv	Höhenruder invers	ein Trick wg positiver Mathematik, HR ziehen soll positive Werte in die Mischer geben!				

Wir beginnen mit den Mischern, denn das ist das wichtigste!

Companion9x starten, neues Modell laden,
in die Mischer gehen, ein neues Grundmodell ist schon angelegt. Das sieht so aus:

CH01	(+100%) Thr
CH02	(+100%) Ail
CH03	(+100%) Ele
CH04	(+100%) Rud
CH05	
CH06	
CH07	

Wir ergänzen das QR2L auf Kanal5 mit -100% Anteil (wg positiver Logik mit -100%)
und vergeben die Namen GS QR1R HR SR QR2L schon mal passend

CH01	(+100%) Thr (GS)
CH02	(+100%) Ail (QR1R)
CH03	(+100%) Ele (HR)
CH04	(+100%) Rud (SR)
CH05	(-100%) Ail (QR2L)

Positive Logik und warum wir das so machen ist ein eigenes Kapitel.

Dazu gibt es 3 Regeln die wir einhalten, dann wird das ganze „Vermischen“ von zig Funktionen und Quellen immer eindeutig und immer gleich, egal wie die nachfolgenden Servos ihre tatsächlichen Bewegungen umsetzen. Das ist dann völlig wurscht!

Jetzt ergänzen wir die Wölbklappen auf CH6 und CH7 und vergeben die Namen
 Quelle ist der Querruder-Stick Ail Anteil Rechts +80% und Links -80%
 wie beim QR mit etwas weniger Anteil, (wg positiver Logik links -80%)
 per Schalter SA ist die Kopplung mit dem Querruder wegschaltbar

CH01	(+100%) Thr (GS)
CH02	(+100%) Ail (QR1R)
CH03	(+100%) Ele (HR)
CH04	(+100%) Rud (SR)
CH05	(-100%) Ail (QR2L)
CH06	(+80%) Ail Schalter (SA↑) (WK1R)
CH07	(-80%) Ail Schalter (SA↑) (WK2L)

Jetzt noch die 2 Bremsklappen BR auf CH8 und CH9

Vorab mal auf einen Schalter zum ausfahren Quelle: MAX Schalter SB
 und in 1s langsam aus- und einfahren (up down u1:1d)

CH01	(+100%) Thr (GS)
CH02	(+100%) Ail (QR1R)
CH03	(+100%) Ele (HR)
CH04	(+100%) Rud (SR)
CH05	(-100%) Ail (QR2L)
CH06	(+80%) Ail Schalter (SA↑) (WK1R)
CH07	(-80%) Ail Schalter (SA↑) (WK2L)
CH08	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH09	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH10	

So jetzt wird es erst mal Zeit zum Simulieren/Testen!

Querruder Stick voll rechts geben, Kanäle verfolgen, dann voll links geben:

Beide Querruder laufen richtig auf +100% und -100%

Beide Wölbklappen laufen richtig auf +80% und -80%

und sind wg SA aktiv mit Querruderstick gekoppelt.

SA umschalten und Wölbklappen sind weg, auf neutral, nur die QR laufen noch

Dann mal Schalter SB betätigen und die Bremsen fahren langsam aus und ein

Falls nicht kontrollieren, korrigieren, überlegen, nachdenken was da gerade passiert ist.

Soweit mal dieses.

Jetzt mal die Differenzierungen auf die Querruder CH2 CH5 beide je +40%

CH01	(+100%) Thr (GS)
CH02	(+100%) Ail Diff (40%) (QR1R)
CH03	(+100%) Ele (HR)
CH04	(+100%) Rud (SR)
CH05	(-100%) Ail Diff (40%) (QR2L)
CH06	(+80%) Ail Schalter (SA↑) (WK1R)
CH07	(-80%) Ail Schalter (SA↑) (WK2L)
CH08	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH09	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH10	

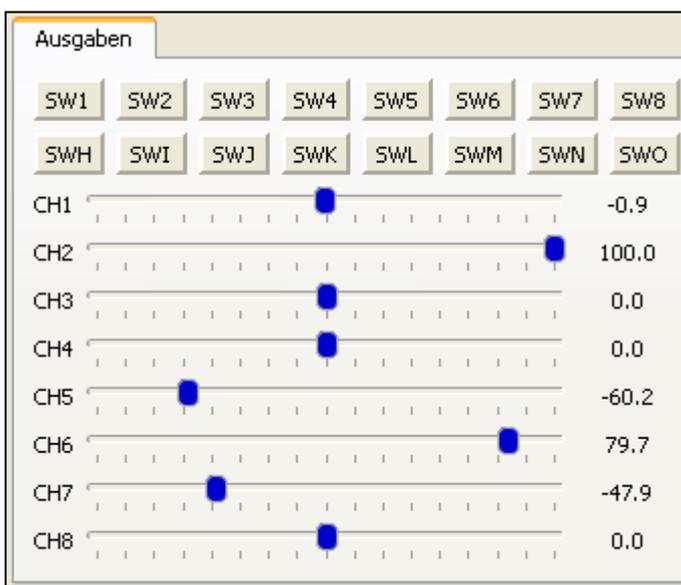
Und auch auf die beiden Wölbklappen differenzieren CH6 CH7 beide je +40%

CH01	(+100%) Thr (GS)
CH02	(+100%) Ail Diff (40%) (QR1R)
CH03	(+100%) Ele (HR)
CH04	(+100%) Rud (SR)
CH05	(-100%) Ail Diff (40%) (QR2L)
CH06	(+80%) Ail Schalter (SA↑) Diff (40%) (WK1R)
CH07	(-80%) Ail Schalter (SA↑) Diff (40%) (WK2L)
CH08	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH09	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH10	

Wieder simulieren, Querruder rechts geben und halten (HalteX)

Querruder: CH2 nach oben +100%, CH5 nach unten -60% $x = -100\% - (-100\% * 40\%)$

Wölbklappen: CH6 nach oben +80%, CH7 nach unten -48% $x = -80\% - (-80\% * 40\%)$



Das muss dann für die Querruder und die Wölbklappen so aussehen.

Jetzt kommt ein kleiner, aber sehr praktischer Trick:

Wenn ich das Höhenruder ziehe gibt der Stick (leider) negative Werte ab. Das passt gar nicht zu unserer positiver Logik, deshalb müssen wir das Signal des **Höhenruder-Stick und der zugehörige Trimmung „irgendwie“ invertieren.** Dazu gibt es 4 Möglichkeiten. Wir verwenden die Bequemste. (Warum das so ist, das ist weiter oben genau beschrieben!)

Hinweis:

Wenn ich in den Inputs den Höhenruder-Knüppel mit -100% invertiere, bringt der Knüppel jetzt ein positive Signal, aber nicht die Trimmung. Die bringt weiterhin ein negatives Signal. Deshalb der Trick mit der Verrechnung in einem freien Mischer-Kanal. Hier wird das Gesamtsignal invertiert, dann passt Knüppel und Trimmung zusammen!

Trick:

Dazu verwenden wir einen freien Mischerkanal als Hilfskanal für Vorberechnungen und benennen ihn entsprechend um, damit wir uns das besser merken können. Hier CH12 Quelle: Höhenruder Ele mit -100% **somit ist der Stick und die Trimmung invertiert** Namen vergeben HRinv und wir haben alle Möglichkeiten die DR/Expo- und Mischer-Menüs uns bieten!

Text eintragen bei Grenzen:
HRinv in CH12 eintragen

Anwenden:
im Mischer erscheint HRinv anstatt CH12

CH 11	0,0	<input type="checkbox"/>
HRinv	0,0	<input type="checkbox"/>
CH 13	0,0	<input type="checkbox"/>

CH10	
CH11	
HRinv	(-100%) Ele (HRinv)
CH13	
CH14	

Anwenden:

Anstatt Höhenruder Ele in Kanal CH3 verwenden wir jetzt **CH12 mit Name HRinv**

CH01	(+100%) Thr (GS)
CH02	(+100%) Ail Diff (40%) (QR1R)
CH03	(+100%) CH12 (HRinv)
CH04	(+100%) Rud (SR)
CH05	(-100%) Ail Diff (40%) (QR2L)
CH06	(+80%) Ail Schalter (SA↑) Diff (40%) (WK1R)
CH07	(-80%) Ail Schalter (SA↑) Diff (40%) (WK2L)
CH08	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH09	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH10	
CH11	
HRinv	(-100%) Ele (HRinv)

Rumpfprogramm für 9 Servos 2QR 2WK, 2BR HR, SR, GS

Und jetzt noch etwas Expo auf Querruder und Höhen damit die Reaktionen um die Mittelstellung sanfter werden. 35% ist ein guter Wert.

Im Menü DR/Expo

Rud				
Ele	Weg (100%)	Expo	(+35%)	(HR)
Thr				
Ail	Weg (100%)	Expo	(+35%)	(QR)

Zusammenfassung:

Wir haben jetzt mal ein Rumpfprogramm für Segler mit 4 und 6 Klappen

Das läuft schon mal in positiver Logik richtig und kann schon ein paar Kleinigkeiten:

Querruder laufen mit Differenzierung und haben Expo 35%

Wölbklappen laufen mit den Querrudern mit, haben Expo, sind differenziert und wegschaltbar

Bremsklappen auf Schalter laufen langsam rein und raus

Höhenruder hat positive Logik per Hilfskanal

Seitenruder hat positive Logik

Motor/Gas(ESC) kommt von Gasknüppel (noch)

Wir haben aber noch nichts „vermischert“, das kommt jetzt:

Dazu sind immer diese 3 Mischer-Fragen zu klären:

- 1. Was sind die Quellen, wo kommt das Signal her**
- 2. Wie soll das verrechnet werden, Anteile, Wirkrichtung, Kurve, Schalter, Flugphasen,...**
- 3. Wo soll es wie das wirken**

Mögliche Mischer , Kombinationen und Funktionen

GS → HR

QR → SR

QR → WK

BR → HR

BR → WK

BR → QR

HR → WK

HR → QR

WK → HR

WK → QR

QR und WK als Butterfly/Krähenstellung

BR, QR und WK als Butterfly/Krähenstellung

Differenzierungsreduktion bei QR WK in Butterfly

QR Umschaltung als Landeklappe/Bremsklappe

WK Umschaltung als Landeklappe/Bremsklappe

GS Motoransteuerung

Das „vermischen“ von Funktionen

Bevor wir weitermachen, speichern wird das Grundmodell erst mal mehrfach ab und vergeben verschiedene Namen, damit wir damit rumspielen und auch mal was „verschlimmbessern“ können.

Dann brauchen wir das Blatt Companion9x Step by Step oder Handbuch ab S.152 dort sieht man wie die div Mischerfunktionen greifen.

Ein Graupner Handbuch MX16 oder MC22 hilft auch sehr, dort wird vieles sehr schön erklärt und ich erspare mir viel zu schreiben.

A: Speed- und Thermik -Stellung dazu programmieren

Das Rumpfprogramm wird jetzt ergänzt um Speed- und Thermik- Stellung

Speed: auf beide Querruder +12% beide Wölbklappen +12%

Thermik: auf beide Wölbklappen -15% beide Wölbklappen -15%

Mit Schalter SC auf UP für Speed (alle Ruder gehen etwas nach oben)

Mit Schalter SC auf Down für Thermik (alle Ruder gehen etwas nach unten)

Schalter SC auf SC-- ist Neutralstellung

Die Werte werden in den Zeilen dazu addiert

Ch2 Ch5 CH6 CH7 dazu addiert += (in companion9x steht nix extra dabei, dann ist addiert!)

Funktionen zum Kopieren und Einfügen verwenden damit das schnell und effektiv geht!

Bitte mal simulieren!

CH01	(+100%) Thr (GS)	CH01	(+100%) Thr (GS)
CH02	(+100%) Ail Diff (40%) (QR1R)	CH02	(+100%) Ail Diff (40%) (QR1R)
	(+12%) MAX Schalter (SC↓) (Speed)		(+12%) MAX Schalter (SC↓) (Speed)
	(-15%) MAX Schalter (SC↑) (Thermik)		(-15%) MAX Schalter (SC↑) (Thermik)
		CH03	(+100%) CH12 (HRinv)
		CH04	(+100%) Rud (SR)
		CH05	(-100%) Ail Diff (40%) (QR2L)
			(+12%) MAX Schalter (SC↓) (Speed)
			(-15%) MAX Schalter (SC↑) (Thermik)
		CH06	(+80%) Ail Schalter (SA↑) Diff (40%) (WK1R)
			(+12%) MAX Schalter (SC↓) (Speed)
			(-15%) MAX Schalter (SC↑) (Thermik)
		CH07	(-80%) Ail Schalter (SA↑) Diff (40%) (WK2L)
			(+12%) MAX Schalter (SC↓) (Speed)
			(-15%) MAX Schalter (SC↑) (Thermik)
CH08	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)	CH08	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
		CH09	(+100%) MAX Schalter (SB↓) Langsam/u1:d1) (BR)
		CH10	
		CH11	
		HRinv	(-100%) Ele (HRinv)

B: Butterfly zum Landen

(noch statisch, wird aber noch dynamisch!)

Dazu müssen: Beide Wölbklappen voll nach unten. Beide Querruder nach oben

Die Differenzierung der Querruder reduziert werden, damit ich im Negativen weiter nach unten komme, ich hab ja eh beide QR oben.

Das machen wir mit je einer weiteren Mischerzeile in jedem Kanal, aber als Replace all above R-Zeichen

R (:=) alle Zeilen darüber werden ungültig (Darunter bleiben sie weiterhin gültig!)

Bei den Wölbklappen: Quelle ist MAX Weight ist -80% sollen ja nur auf festen Wert laufen

Bei den Querrudern: Quelle ist Ail, Weight ist +/-100%, aber mit Offset +/-50 % verschieben

~~beachte CH5 negativ wg der Berechnung einer Mischerzeile!~~ (CH5 positiv wg neuer Misherberechnung)

Diff von 40 auf 30% (oder noch weiter) zurückgenommen

Als Schalter der SF mit Butterfly Ein /Aus

Der Knackpunkt ist das Replace R Befehl. Mit einer einzigen Zeile werden alle Zeilen darüber ungültig und wir beginnen praktisch neu.

Mit Kopieren und Einfügen arbeiten, Namen vergeben, dann geht das alles innerhalb von Sekunden und das sieht jetzt mal so aus:

CH01	(+100%)Thr (GS)
CH02	(+100%)Ail Diff (40%) (QR1R)
	(+12%)MAX Schalter (SC↓) (Speed)
	(-15%)MAX Schalter (SC↑) (Thermik)
	R (+100%)Ail Schalter (SF↓) Offset (50%) Diff (30%) (Butterfl)
CH03	(+100%)CH12 (HRinv)
CH04	(+100%)Rud (SR)
CH05	(-100%)Ail Diff (40%) (QR2L)
	(+12%)MAX Schalter (SC↓) (Speed)
	(-15%)MAX Schalter (SC↑) (Thermik)
	R (-100%)Ail Schalter (SF↓) Offset (-50%) Diff (30%) (Butterfl)
CH06	(+80%)Ail Schalter (SA↑) Diff (40%) (WK1R)
	(+12%)MAX Schalter (SC↓) (Speed)
	(-15%)MAX Schalter (SC↑) (Thermik)
	R (-80%)MAX Schalter (SF↓) (Butterfl)
CH07	(-80%)Ail Schalter (SA↑) Diff (40%) (WK2L)
	(+12%)MAX Schalter (SC↓) (Speed)
	(-15%)MAX Schalter (SC↑) (Thermik)
	R (-80%)MAX Schalter (SF↓) (Butterfl)
CH08	(+100%)MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH09	(+100%)MAX Schalter (SB↓) Langsam/u1:d1) (BR)
CH10	
CH11	
HRinv	(-100%)Ele (HRinv)
CH13	

Bitte ausgiebig simulieren!

Offset +50% wg neuer Mischerberechnung!

Da nun fast alles funktioniert und der R-Befehl verstanden wurde, was machen wir mit den Bremsklappen?

Da gibt es 2 Möglichkeiten, einzeln dazu schalten wie bisher auch, das geht ja immer noch oder aber auch mit auf den Schalter SF legen als Butterfly

openTx für Taranis Anleitung Deutsch

Mehr abbremsten ist dann aber nicht mehr möglich!

Wir legen sie auch auf den SF-Schalter und benennen sie um in ButterBR

CH01	(+100%) Thr (GS)
CH02	(+100%) Ail Diff (40%) (QR1R)
	(+12%) MAX Schalter (SC↓) (Speed)
	(-15%) MAX Schalter (SC↑) (Thermik)
R	(+100%) Ail Schalter (SF↓) Offset (50%) Diff (30%) (Butterfl)
CH03	(+100%) CH12 (HRinv)
CH04	(+100%) Rud (SR)
CH05	(-100%) Ail Diff (40%) (QR2L)
	(+12%) MAX Schalter (SC↓) (Speed)
	(-15%) MAX Schalter (SC↑) (Thermik)
R	(-100%) Ail Schalter (SF↓) Offset (-50%) Diff (30%) (Butterfl)
CH06	(+80%) Ail Schalter (SA↑) Diff (40%) (WK1R)
	(+12%) MAX Schalter (SC↓) (Speed)
	(-15%) MAX Schalter (SC↑) (Thermik)
R	(-80%) MAX Schalter (SF↓) (Butterfl)
CH07	(-80%) Ail Schalter (SA↑) Diff (40%) (WK2L)
	(+12%) MAX Schalter (SC↓) (Speed)
	(-15%) MAX Schalter (SC↑) (Thermik)
R	(-80%) MAX Schalter (SF↓) (Butterfl)
CH08	(+100%) MAX Schalter (SF↓) Langsam/u1:d1) (ButterBR)
CH09	(+100%) MAX Schalter (SF↓) Langsam/u1:d1) (ButterBR)
CH10	
CH11	
HRinv	(-100%) Ele (HRinv)
CH13	

Soweit mal ok, aber ein paar Schönheitsfehler sind noch zu beheben

Das Butterfly schaltet schlagartig zu wenn ich den SF betätige,
nur bei den Bremsen läuft es mit 1s raus und 1 sec rein

Dieses langsame Ein- und Ausfahren wollen wir auch auf den anderen 4 Rudern haben.

Aber:

Zeiten mit Verzögerung und Langsam kann direkt nur in einer einzelnen Mischerzeile stehen (So wie bei den Bremsen nur eine einzelne Zeile je Kanal)

Was tun: Flugphasen verwenden da kann man sanft umschalten

Wie geht das: Verblüffend einfach!

Flugphasen haben Prioritäten

FP0 ist immer da

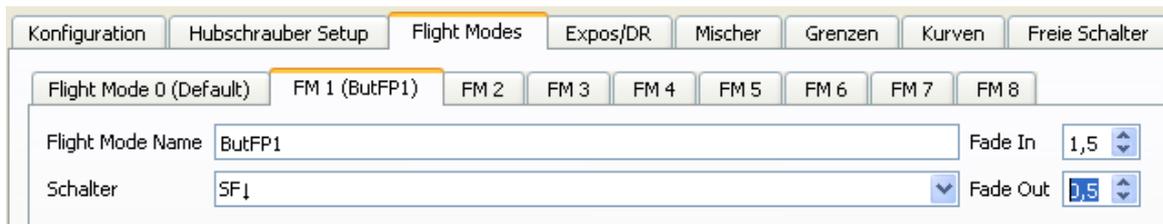
FP1 ist die höchste

FP8 die niedrigste

Die höherer FP überschreibt die niedrigere FP

openTx für Taranis Anleitung Deutsch

Das ist aber hier mal egal, also rein in die Flugphasen
und eingestellt FP1 Up-Zeit 1,5s Down-Zeit 0,5s Schalter SFup
Das heißt hier Fade In und Fade Out für Flugphase aktivieren und deaktivieren

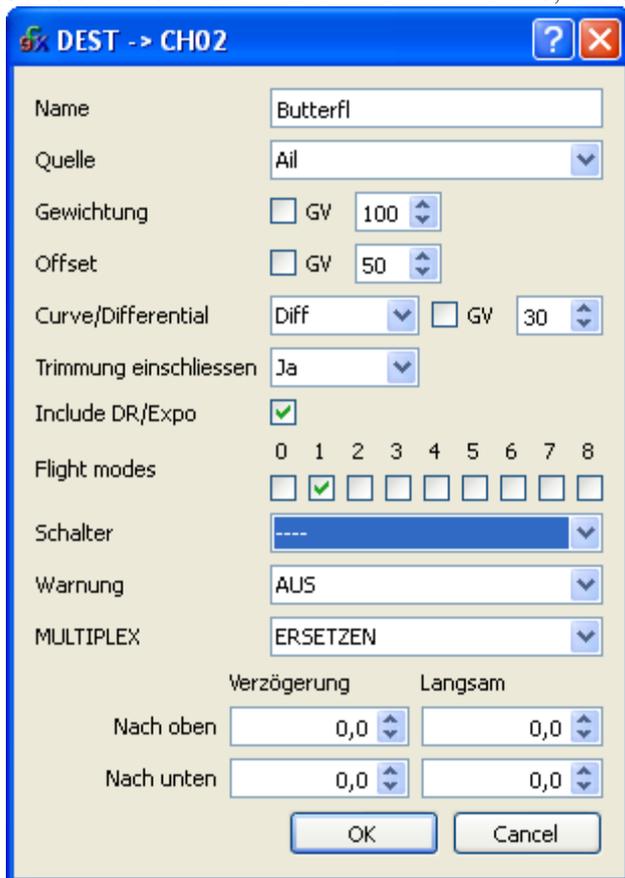


Und nun in den Mischern anpassen, die Zeile wo das R steht, dort nur die FP1 aktivieren
das wars, und schon haben wir einen sanften Übergang.
Jetzt ganz schnell alle Zeilen wo da R steht die Mischer auf FP1 abändern, fertig.

Schalter in den Mischern brauche ich auch keine mehr, das ist ja in der FP1 definiert.

Bei den Bremsen, die ja auch via SF laufen, aber die Zeiten noch unabhängig von der Flugphase,
aktivieren wir im Mischermenü auch nur die FP1, SF rausnehmen, Zeiten rausnehmen. Denn das kommt
ja jetzt aus den Flugphasen.

So sieht das Mischermenü für die FP1 aus, die Schalter und Zeiten braucht man nicht mehr



openTx für Taranis Anleitung Deutsch

Jetzt laufen auch die 6 Klappen (2QR 2WK 2BR) absolut synchron rein und raus
Die Zeiten sind zentral in der FP1 eingestellt der Schalter SF auch

Und so sieht der fertige Mischer vorläufig aus.

Da haben sich durch die Verwendung der Flugphase FP1 ein paar Dinge vereinfacht
Bitte ausgiebig simulieren!

CH01	(+100%)Thr (GS)
CH02	(+100%)Ail Diff (40%) (QR1R) (+12%)MAX Schalter (SC↓) (Speed) (-15%)MAX Schalter (SC↑) (Thermik)
	R (+100%)Ail Flight mode (ButFP1) Offset (50%) Diff (30%) (Butterfl)
CH03	(+100%)CH12 (HRinv)
CH04	(+100%)Rud (SR)
CH05	(-100%)Ail Diff (40%) (QR2L) (+12%)MAX Schalter (SC↓) (Speed) (-15%)MAX Schalter (SC↑) (Thermik)
	R (-100%)Ail Flight mode (ButFP1) Offset (-50%) Diff (30%) (Butterfl)
CH06	(+80%)Ail Schalter (SA↑) Diff (40%) (WK1R) (+12%)MAX Schalter (SC↓) (Speed) (-15%)MAX Schalter (SC↑) (Thermik)
	R (-80%)MAX Flight mode (ButFP1) (Butterfl)
CH07	(-80%)Ail Schalter (SA↑) Diff (40%) (WK2L) (+12%)MAX Schalter (SC↓) (Speed) (-15%)MAX Schalter (SC↑) (Thermik)
	R (-80%)MAX Flight mode (ButFP1) (Butterfl)
CH08	(+100%)MAX Flight mode (ButFP1) (ButterBR)
CH09	(+100%)MAX Flight mode (ButFP1) (ButterBR)
CH10	
CH11	
HRinv	(-100%)Ele (HRinv)
CH13	

Namen sind absolut wichtig! Positive Logik ist auch wichtig!

Soweit alles klar?

Jetzt kommen nur noch ein paar Schönheitsmischer

Butterfly auf Höhe kompensieren auch auf langsam

Quer auf Seite wg scharfen Kurvenflug oder auch nicht bei Thermik

Motor auf Höhe wg Motor-Zug ausgleichen

Motoransteuerung per 2/3-Stufen Schalter oder seitlichen Schieberegler SL

Und das Wichtigste zum Schluss:

Butterfly nicht statisch fix, sondern dynamisch auf den (Gas)-Knüppel 6 Ruder rein und raus

openTx für Taranis Anleitung Deutsch

So, wenn man das jetzt mal wirken lässt.

Die paar Zeilen pro Kanal genau betrachtet (nur 4 Zeilen pro Kanal, beliebig viele möglich), gute Namen vergibt (leider nur 6-8 Buchstaben wg den kleineren Steuerungen die auch auf openTx laufen) und positive Logik verwendet.

Wie Ihr seht, da ist absolut nichts dahinter und wir haben vielleicht 5% der Möglichkeiten von openTx genutzt, nicht mehr!

Dann vergleicht das mal mit den großen "Profi"-Steuerungen, was da für ein Tam - Tam gemacht wird und zig Seiten in den RCnetwork verblasen wird für nichts, nur heiße Luft.

Jetzt bitte mal ein Handbuch einer MC MX oder Futaba oder Spektrum anschauen und vergleichen was da im Hintergrund für ein "unsichtbares" Zeug abläuft. Dies nur mit diesem Schalter, jenes nur damit, das ist aber nur mit der großen Steuerung möglich und den Schalter so zuordnen usw.

Sorry, das ist historischer Krampf, leider so gewachsen!

Das kümmert uns überhaupt nicht, wir verwenden was wir wollen, wie und wo wir es wollen.

Jeder Mischerzeile ist eine komplett eigenständige universal einsetzbare Zeile!

Wir müssen immer nur an 3 Dinge denken:

Wo kommt das Signal her
Wie soll es verarbeitet werden
Wo soll es wirken

Dann kann man das alles auch sauber dokumentieren und als File ausdrucken
→ bitte ausprobieren, dort kann man viel Text ergänzen!!

C: Falls es jemand aufgefallen ist:

Linkes QR CH5 Offset - 50% Rechts QR CH2 Offset +50%

Warum das denn, wenn ich doch beide QR nach oben verschieben will, also ins positive, wo ist da die positive Logik?

Das liegt an der Art der Mischer Berechnung!

Stark vereinfacht wird so gerechnet:

$$Y = [(source + offset) * weight] + Trim$$

Mathe auf gelöst

$$Y = [(source * weight) + (offset*weight)] + Trim$$

CH5: weight ist aber negativ, denn QR soll ja nach unten gehen!

Minus * Minus = Plus

$$- offset * - weight = + offset$$

Somit klar? Gib bei Ch5 -offset ein damit er nach oben geht

Ja, die ganze „Verschieberei“ per Offset könnte man auch ganz anders machen

Eine eigene unabhängige Mischerzeile rein, z.B. nach der R- Zeile

oder Kurven verwenden,

oder min noch 3 oder 4 Varianten sind möglich

Achtung: ab openTx2.0 für die Taranis werden die Mischer anders berechnet!

$[(Source * Weight) + Offset] = \text{Mischerwert} + Trim \rightarrow (DR/Expo/Kurve) \rightarrow \text{Kanal}$

Dadurch vereinfacht sich vieles, siehe Seite:74, 75

Also bei Offset und Gewichtung nachrechnen!

D: Dynamische Bremsklappen, Wölbklappen, Querruder

Jetzt machen wir mal weiter und zwar mit dynamischen Werten vom Gasknüppel, der alle 6 Ruder des Butterfly steuert

Wenn das Butterfly ausfährt bäumt sich der Segler auf, das wird auch dynamisch mit Tiefe weggemischt.

Kein Segler braucht einen Gasknüppel für den Motor, der wird via Schalter oder Analogschieber gesteuert. Aber er braucht dynamisches Butterfly beim Landen, darum wird der Gasknüppel dazu verwendet.

Vom Gas Knüppel gesteuert fahren die Bremsklappen 0-100% aus die Wölbklappen nach unten 0-80% und die Querruder nach oben 0-50%, Diff auf 5% reduziert und das Tiefenruder nach unten 0-20% zum Ausgleich

Die Werte für Weight und Offset werden berechnet.

Ausgangslage unseres bisherigen Mischers

```

CH01      (+100%) Thr (GS)
CH02      (+100%) Ail Diff (40%) (QR1R)
          (+12%) MAX Schalter (SC↓) (Speed)
          (-15%) MAX Schalter (SC↑) (Thermik)
R (+100%) Ail Flight mode (ButFP1) Offset (50%) Diff (30%) (Butterfl)
CH03      (+100%) CH12 (HRinv)
CH04      (+100%) Rud (SR)
CH05      (-100%) Ail Diff (40%) (QR2L)
          (+12%) MAX Schalter (SC↓) (Speed)
          (-15%) MAX Schalter (SC↑) (Thermik)
R (-100%) Ail Flight mode (ButFP1) Offset (-50%) Diff (30%) (Butterfl)
CH06      (+80%) Ail Schalter (SA↑) Diff (40%) (WK1R)
          (+12%) MAX Schalter (SC↓) (Speed)
          (-15%) MAX Schalter (SC↑) (Thermik)
R (-80%) MAX Flight mode (ButFP1) (Butterfl)
CH07      (-80%) Ail Schalter (SA↑) Diff (40%) (WK2L)
          (+12%) MAX Schalter (SC↓) (Speed)
          (-15%) MAX Schalter (SC↑) (Thermik)
R (-80%) MAX Flight mode (ButFP1) (Butterfl)
CH08      (+100%) MAX Flight mode (ButFP1) (ButterBR)
CH09      (+100%) MAX Flight mode (ButFP1) (ButterBR)
CH10
CH11
HRinv     (-100%) Ele (HRinv)
CH13
    
```

E: Berechnete Einstellwerte für Weight und Offset:

Wölbklappen fahren mit dem Gasknüppel Thr nach unten Aus und Ein

von 0 bis -80%, davon die Mitte ist -40%

$-80/200 = -0,4 = -40\%$ Weight

$-40/-40 = 1,0 = 100\%$ Offset

CH06	(+80%)	Ail	Schalter (SA↑)	Diff (40%)	(WK1R)
	(+12%)	MAX	Schalter (SC↓)	(Speed)	
	(-15%)	MAX	Schalter (SC↑)	(Thermik)	
	R (-40%)	Thr	Flight mode (ButFP1)	Offset (100%)	(Butterfl)
CH07	(-80%)	Ail	Schalter (SA↑)	Diff (40%)	(WK2L)
	(+12%)	MAX	Schalter (SC↓)	(Speed)	
	(-15%)	MAX	Schalter (SC↑)	(Thermik)	
	R (-40%)	Thr	Flight mode (ButFP1)	Offset (100%)	(Butterfl)

Bremsklappen fahren mit Gasstellung Thr nach oben Aus und Ein

von 0 bis +100% davon die Mitte ist +50%

$100/200 = 0,5 = 50\%$ Weight

$50/50 = 1,0 = 100\%$ Offset

CH08	(+50%)	Thr	Flight mode (ButFP1)	Offset (100%)	(ButterBR)
CH09	(+50%)	Thr	Flight mode (ButFP1)	Offset (100%)	(ButterBR)

Höhenruder Tiefenkorrektur mit Gasknüppel Thr 0% bis -20%, davon die Mitte ist -10%

$-20%/200\% = 0,1 = -10\%$ Weight

$-10\%/-10\% = 1,0 = 100\%$ Offset

Was jetzt noch fehlt ist Kleinkram

CH03	(+100%)	CH12	(HRinv)		
	(-10%)	Thr	Flight mode (ButFP1)	Offset (100%)	(HRkorrek)

Achtung ab openTx2.0 für die Taranis werden die Mischer anders berechnet!

$[(\text{Source} * \text{Weight}) + \text{Offset}] = \text{Mischerwert} + \text{Trim} \rightarrow (\text{DR/Expo/Kurve}) \rightarrow \text{Kanal}$

Dadurch vereinfacht sich vieles, siehe Seite:74, 75

Also Offset und Gewichtung nachrechnen!

Das Querruder soll mit Gasknüppel Thr nach oben fahren,
dazu verwenden wir **nach der R-Zeile** einen weiteren Mischer
der durch Thr das Querruder nach oben fährt, **addiert zur R-Zeile!**

Querruder fahren nach oben 0 bis +50% davon die Mitte ist +25%

$50/200=0,25=25\%$ Weight

$25\%/25\%=1,0=100\%$ Offset

```
CH02      (+100%)Ail Diff(40%) (QR1R)
           (+12%)MAX Schalter(SC↓) (Speed)
           (-15%)MAX Schalter(SC↑) (Thermik)
R (+100%)Ail Flight mode(ButFP1) Diff(5%) (Butterfl)
           (+25%)Thr Flight mode(ButFP1) Offset(100%) (QR1 hoch)
```

```
CH05      (-100%)Ail Diff(40%) (QR2L)
           (+12%)MAX Schalter(SC↓) (Speed)
           (-15%)MAX Schalter(SC↑) (Thermik)
R (-100%)Ail Flight mode(ButFP1) Diff(5%) (Butterfl)
           (+25%)Thr Flight mode(ButFP1) Offset(100%) (QR2 hoch)
```

Diese sind jetzt exakt die gleichen Bewegungs-Werte wie vorher als wir noch rein statisch die Mischer per Schalter umgeschaltet haben.

Bitte mal simulieren:

Gas ganz nach unten
Schalter SC in die Mitte

Dann Schalter SF umlegen
und jetzt langsam Gas geben

Genau schauen was jetzt 8 oder 9 Servos alles gleichzeitig machen

Und mit positiver Logik ist das sofort zu erkennen!

Wehe dem, der vorher schon Servo Revers gemacht hat, kann mir keiner erzählen das der sieht was da in der Simulation abläuft.

Was jetzt noch fehlt ist Kleinkram

Mischer Querruder auf Seite für scharfe Kurven, (heist wohl Kombiswitsh) oder eben auch gerade nicht bei Thermik, damit er sich nicht schräg reinlegt.

Je nach Geschmack kann man das dann auch wegschalten

CH04	(+100%) Rud (SR)
	(+25%) Ail (QR->SR)

und

Motoransteuerung via Schalter oder Geber LS oder RS

CH01	(+100%) LS (GSsanft)
------	----------------------

Die Motoransteuerung könnte man jetzt noch verfeinern.

Mit sanft Anlauf und sanft Stop also slow up und slow down Zeiten

F: Fertiger Segler mit 4 und 6 Klappen

Das ist jetzt unser universeller 4 und 6 Klappen Mischer

Mit dynamischem Butterfly für 6 Klappen und dynamischer Höhenkorrektur

Mit Speedflug, Thermikflug, Normalflug,

Motor von LS Geber, Quer auf Seite gemischt

CH01	(+100%) LS (GSSanft)
CH02	(+100%) Ail Diff (40%) (QR1R) (+12%) MAX Schalter (SC↓) (Speed) (-15%) MAX Schalter (SC↑) (Thermik) R (+100%) Ail Flight mode (ButFP1) Diff (5%) (Butterfl) (+25%) Thr Flight mode (ButFP1) Offset (100%) (QR1 hoch)
CH03	(+100%) CH12 (HRinv) (-10%) Thr Flight mode (ButFP1) Offset (100%) (HRkorrek)
CH04	(+100%) Rud (SR) (+25%) Ail (QR->SR)
CH05	(-100%) Ail Diff (40%) (QR2L) (+12%) MAX Schalter (SC↓) (Speed) (-15%) MAX Schalter (SC↑) (Thermik) R (-100%) Ail Flight mode (ButFP1) Diff (5%) (Butterfl) (+25%) Thr Flight mode (ButFP1) Offset (100%) (QR2 hoch)
CH06	(+80%) Ail Schalter (SA↑) Diff (40%) (WK1R) (+12%) MAX Schalter (SC↓) (Speed) (-15%) MAX Schalter (SC↑) (Thermik) R (-40%) Thr Flight mode (ButFP1) Offset (100%) (Butterfl)
CH07	(-80%) Ail Schalter (SA↑) Diff (40%) (WK2L) (+12%) MAX Schalter (SC↓) (Speed) (-15%) MAX Schalter (SC↑) (Thermik) R (-40%) Thr Flight mode (ButFP1) Offset (100%) (Butterfl)
CH08	(+50%) Thr Flight mode (ButFP1) Offset (100%) (ButterBR)
CH09	(+50%) Thr Flight mode (ButFP1) Offset (100%) (ButterBR)
CH10	
CH11	
HRinv	(-100%) Ele (HRinv)
CH13	

Mehr kann man dann aber nicht für einen 4 und 6 Klappen Segler „vermischen“
(doch kann man schon → Klapptriebwerk Ein- und Ausfahren, Fahrwerk, usw.)

Und Leute, nicht vergessen, die Zahlen sind Spielbeispiele damit man was sieht.
Das muss alles auch erfolgen werden!

Anstatt dieser zusätzlichen 2 Zeilen bei CH2 CH5 nach der R-Zeile hätte man im Offset dieser R-Zeilen eine GVAR als Festwert im Flugphase 1 eintragen können. usw. usw.

Ja, es gibt noch viele weitere Möglichkeiten

Soweit mal dies Übung

**Tip: Taranis Seglereinstellungen gibt es hier : <http://rc-soar.com/opentx/setups/f3f/index.htm>
Dort sind besten F3F und F3J Segler-Spezialisten Englands!**

Beispiel: 4 Klappen-Segler Butterfly, Wölbklappen, Speed Thermik variabel einstellbar

Ohne viel Aufwand und Schalter, alles mit Kurven programmieren

Normal hat man dazu 3-4 Flugphasen die man per Schalter umschaltet

Neutral: Alles im Strak +/-0mm

Speed: Quer und Wölb etwas nach oben, verringert den Widerstand, ca 2-3mm
(hier Kurve 3 linker Teil mit LS und SA zur Freigabe)

Thermik: Quer und Wölb etwas nach unten, gibt mehr Auftrieb, ca 3-4mm
(hier Kurve 3 rechter Teil mit LS und SA zur Freigabe)

Landung: Butterfly, Quer nach oben und Wölb stark nach unten (hier per Schalter SA freigeben)
Quer nach oben ca +20° bis +35° wirkt wie eine Schränkung,
verhindert einen Strömungsabriss
Wölb stark nach unten ca 45° bis 80° das ist für das Bremsen zuständig.

Speed und Thermik: Mit LS variabel einstellen können, LS in der Mitte, alles im Strak

Gas-Knüppel: wird hier auch zentriert, steht somit in der Mitte.

Motor: Gasknüppel ab Mitte nach vorne variabel einstellen, (Kurve 1 Motor-Regler 0-100%)

Butterfly: Gasknüppel ab Mitte nach hinten variabel einstellen (Kurve 2 und SA zur Freigabe)

Anstatt mit Flugphasen und Schaltern, kann man das aber auch alles mit 3-4 Kurven machen

Vorverarbeitung, incl Kurven und Freigaben erfolgt in den Inputs I1, I7, I8

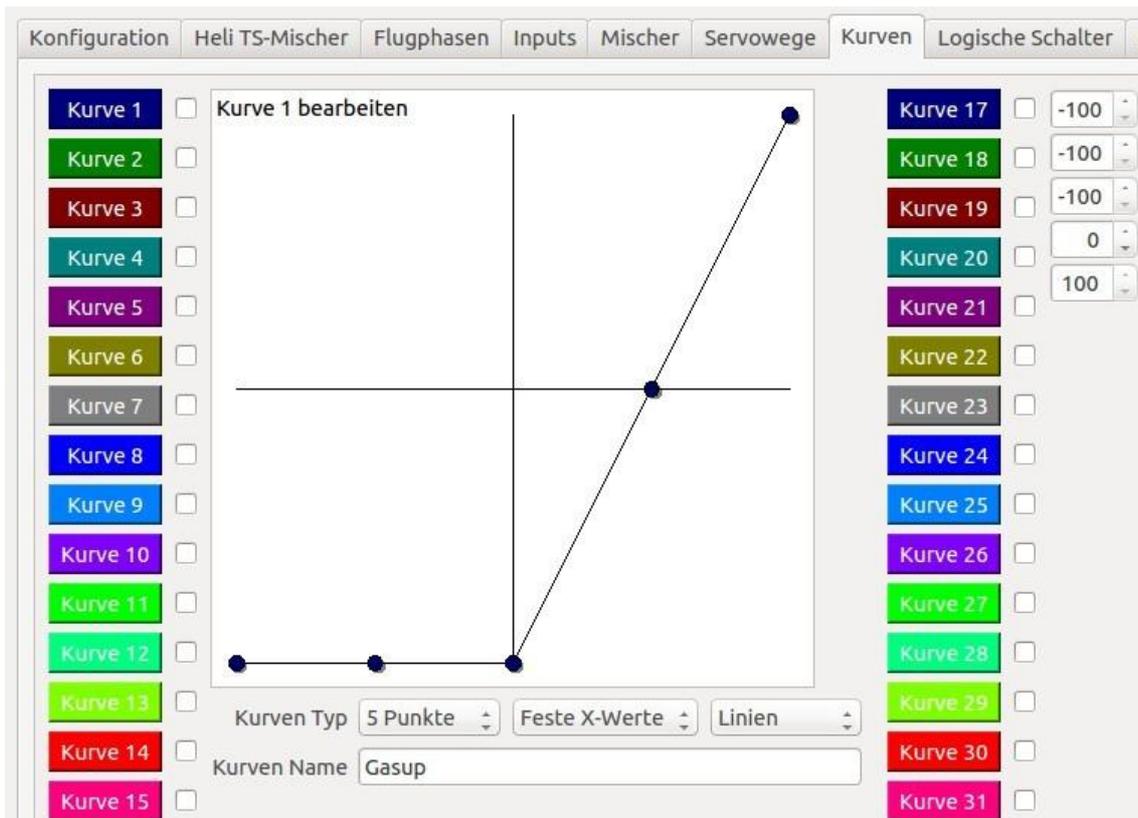
Werte berechnen und Richtungen erfolgt in den Mischern

Kanalbelegung am Beispiel:

CH1 Gas
CH2 QR1
CH3 Hoh
CH4 Sei
CH5 QR2
CH6 frei
CH7 Wölb1
CH8 Wölb2

Bitte Beispiel eingeben und nach Bedarf anpassen

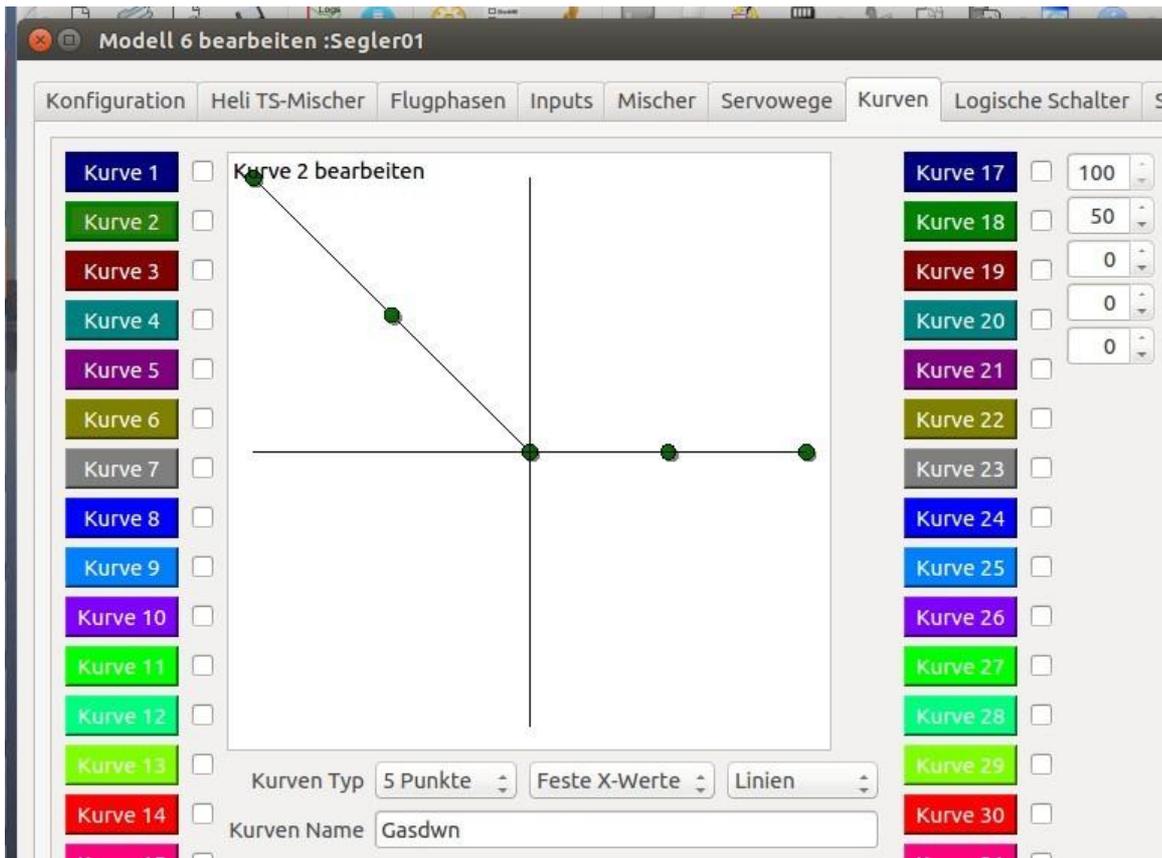
Kurve 1 für Gasknüppel ab Mitte nach vorne -->Motor voll steuerbar 0-100% des Reglers



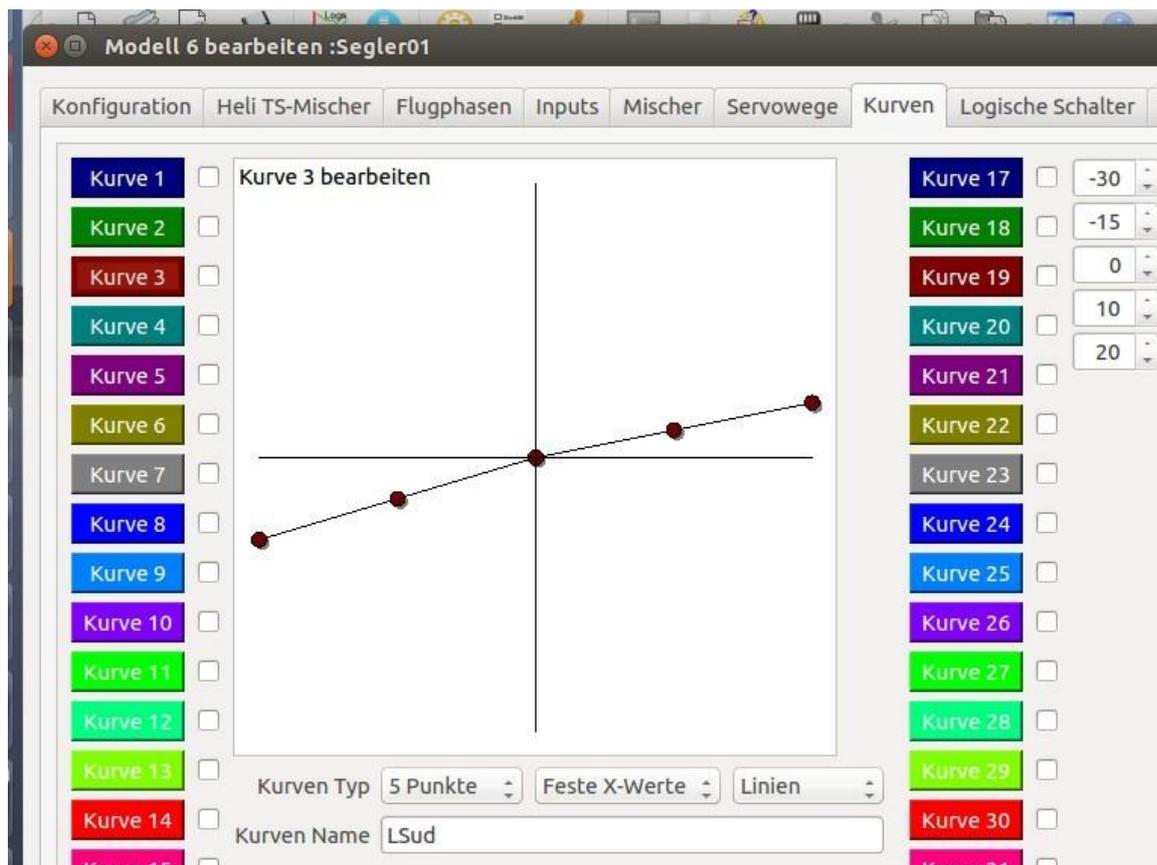
Da kann man auch noch 3-5% Totgang nach vorne eingeben, damit der Motor nicht sofort ab Mitte losläuft.

3.Punkt ändern statt 0/-100 nach 3/-100 (mit variablen X/ Y Punkten)

Kurve 2 für Gasknüppel ab Mitte nach hinten --> Butterfly, QR, Wölb invers in den Mischern



openTx für Taranis Anleitung Deutsch Kurve 3 für Speed und Thermik durch 2 unterschiedliche Steigungen

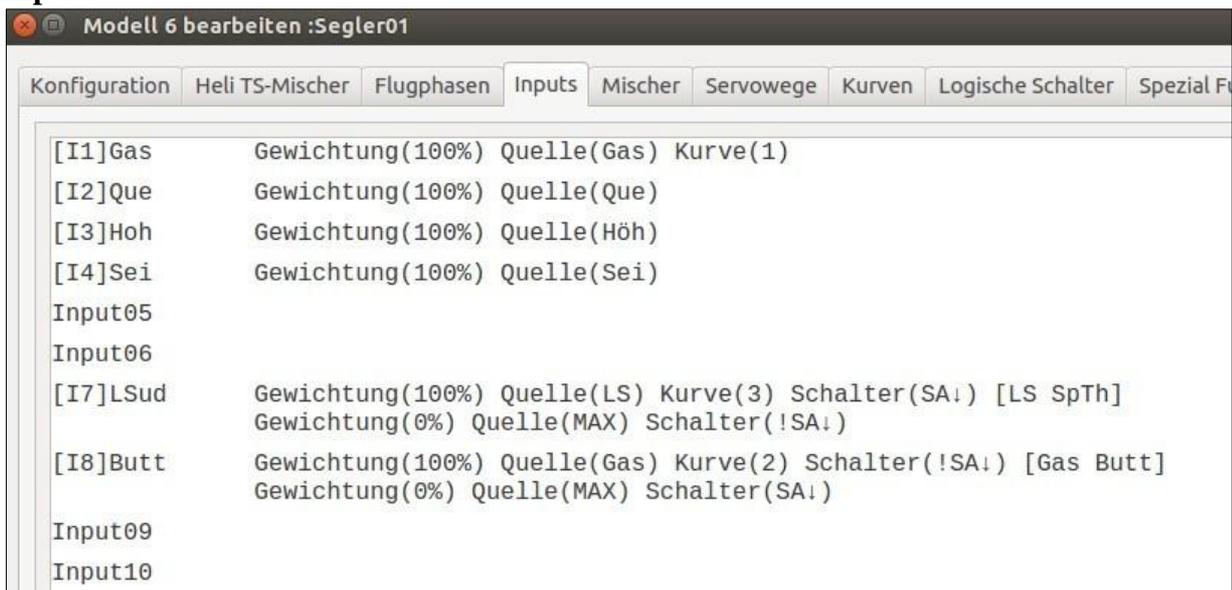


LS Thermik oder Speed variabel einstellbar, LS in der Mitte, dann Neutral alles im Strak

SA Schalter gibt Butterfly frei, sperrt Speed und Thermik und umgekehrt.

Trick: Wird gegenseitig gemacht, damit in den Inputs die Zeilen auf Null umgeschaltet werden.

Inputs:



Mischer und Kanäle



Das ist nur mal ein Spielbeispiel damit man was am Simulator sieht!

Man kann auch mal den Schallter SA rausnehmen, dann hat man alles gleichzeitig zur Verfügung, macht man aber normal nicht.

Automatische Querruder-Differenzierung-Reduzierungsfunktion

Wenn soweit klar, dann folgt eine automatische Querruder-Differenzierung-Reduzierung d.h. je mehr man Butterfly gibt, desto weniger soll die Querruderdifferenzierung wirken.

Ich habe hier im Beispiel einen Festwert von 30% für die Querruderdifferenzierung eingegeben. Den werden wir durch eine globale Variabel GVAR einstellbar machen!

Und zwar so:

Kein Butterfly, also Gasknüppel =>0 dann wirkt GVAR mit 30% als Diff auf die beiden QR
Volles Butterfly, also Gasknüppel bei -100% dann wirkt GVAR mit 0% als Diff auf die beiden QR

Das kann man mit der Kurve 2 machen, oder mit einer eigenen Kurve (ist eleganter, Kurve 4)

Lösung:

Input I10 mit Kurve 2 mit Gewichtung -30 und Offset +30

Lösung genau anschauen und in den GVAR die GV1 beobachten.

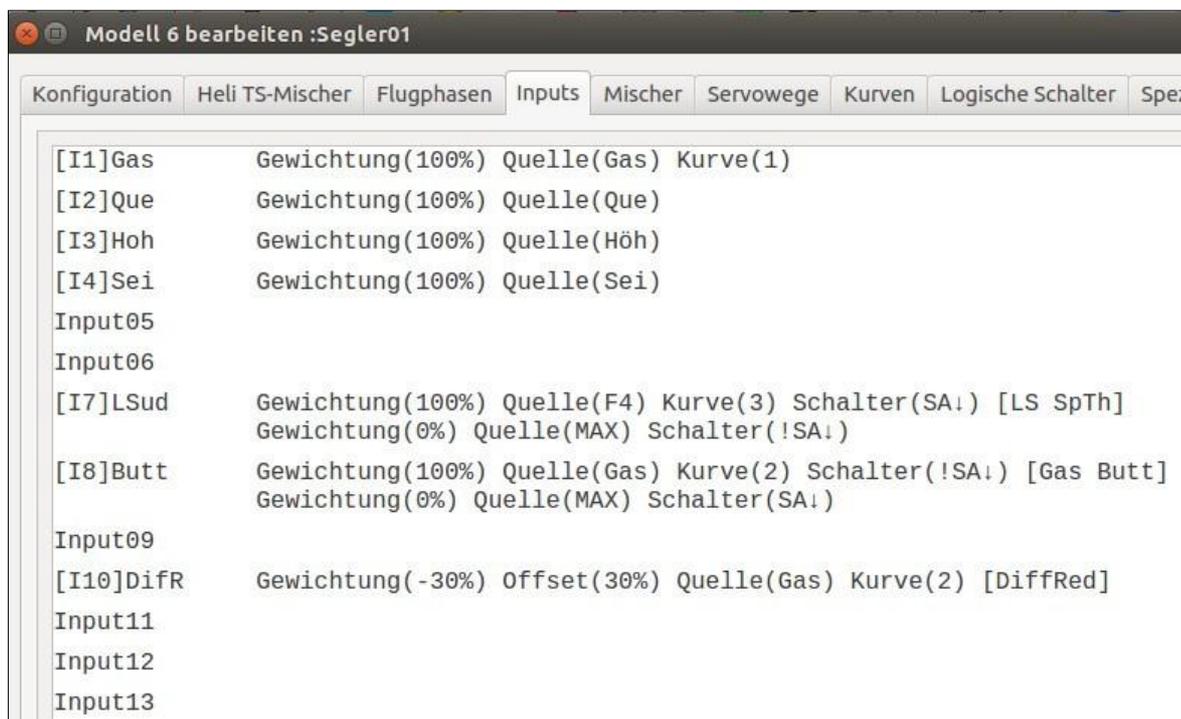
Die geht von 30 auf 0 zurück

Da man beim Butterfly die QR hochstellt, geht dann das QR mehr als sonst nach unten d.h. Die Querruderdifferenzierung ist variabel reduziert auf Null,

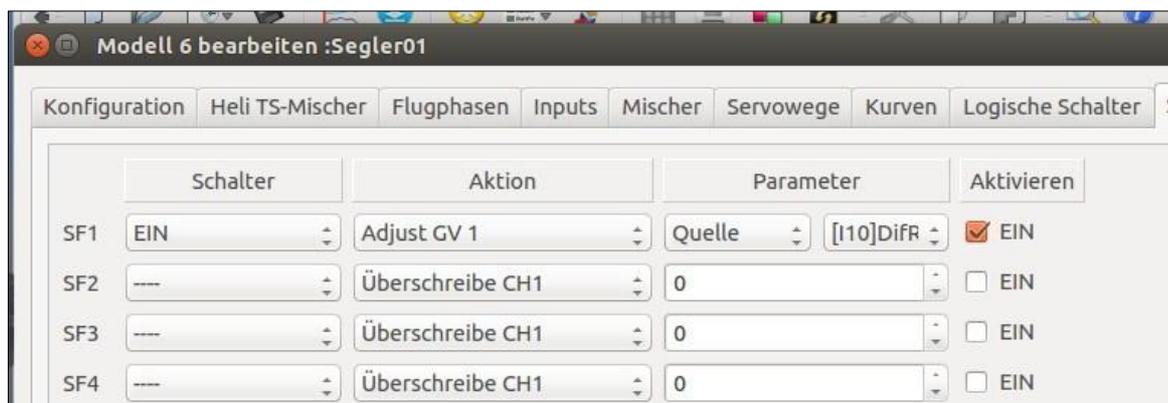
Das könnte man sogar noch in die andere Richtung übertreiben



Mischer 10 zur Querruder-Differenzierung-Reduzierung von 30% auf 0%



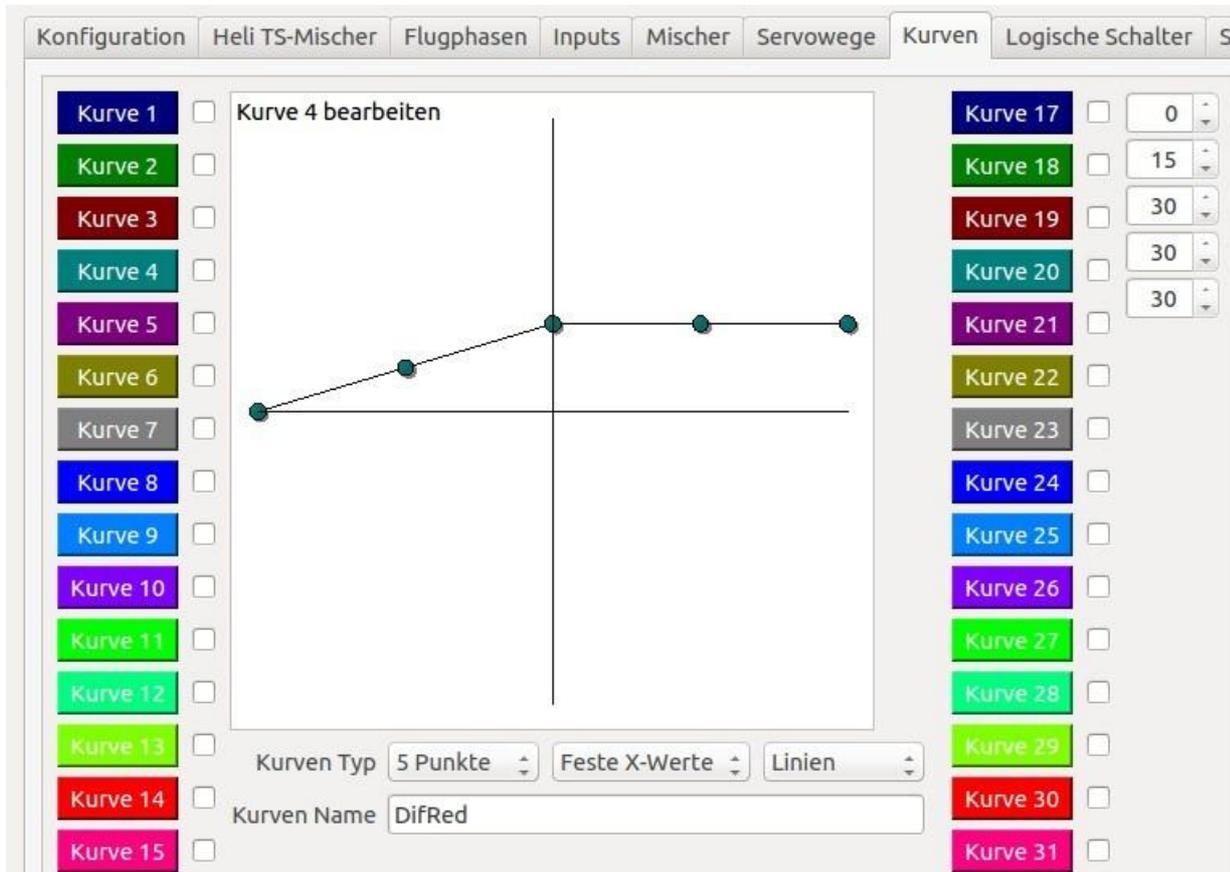
Versorgung der globalen Variablen GV1 mit Werten vom Mischer 10



Eine Alternative wäre mit Kurve 4 möglich

zur Querruder-Differenzierung-Reduzierungs-Funktion

dann aber Gewichtung = 100 und Offset = 0, denn die Kurve 4 macht schon alles selber.



Was fehlt noch?

- Ein Gas Sicherheitsschalter im Gasmischer oder als Spezialfunktion Override -100%
- Etwas Tiefe wenn das Butterfly ausfährt, aber das ist ein normaler Mischer

Ich hoffe man konnte mir folgen, sonst mache ich das Ganze in kleineren Schritten
Rückmeldungen erwünscht!

Ja: „Segler-Profis“ und Wettbewerbsflieger machen das noch viel differenzierter.
Für viele Segelflieger reicht das, wenige Schalter, alles variabel und sanft einstellbar.

Teil D Viele Beispiele, Tips und Tricks

Beispiel: Die grundsätzliche Dinge der Programmierung

Es sind im wesentliche 6 wichtige Dinge:

1. Vergiss alles von bisherigen Sendern und Herstellern mit festen Vorgaben, Kanälen, Gebern, Belegungen, Schalter, Mischer und fertigen Funktionen. Es gibt bei opentx keine Beschränkungen oder feste Vorgaben, alles ist mit allem überall gleichberechtigt möglich.

2. Das zentrale Element ist der Mischer, alles läuft über Mischer, jeder Mischer ist ein Universalmischer und kann alles, es gibt keine Spezialmischer
Berechnung: $[(\text{Quelle} * \text{Gewichtung}) + \text{Offset}]$

3. Überleg dir für die Programmierung immer 3 Dinge: (EVA-Prinzip)

E- Eingang: Wo kommt mein Signal her, was ist meine Signalquelle

V- Verarbeiten: Was will ich mit dem Signal wie machen, verrechnen, mischen,

A- Ausgang: Wo soll das Signal wie wirken, Kanal, Servo, Schalter

4. Schalter als **Mischerquelle**, liefern von sich aus schon automatisch -100% 0% +100% (3-Stufen)

bzw -100% +100% (2-Stufen), Logische Schalter liefern -100% oder +100%

Schalter als **Mischerschalter** aktivieren/deaktivieren Mischerzeilen

R= Replace ersetzen alle Mischerzeile die darüber stehen.

:= Replace += Addier *= Multiplizer

5. Logische Schalter, Programierbare Schalter, Custom Switch, sind logische Verknüpfungen, Abfragen, Schaltungen und Kontrollstrukturen die wieder überall wirken können.

6. Servos übersetzen nur die Ergebnisse der Mischermathematik an die reale Welt, also die Wege und Richtungen für die Ruder.

Bitte nie die Servos **vorab** einfach invertieren damit das Ruder „schon mal richtig läuft“.

Erst die Mischermathematik, Mischerverechnung fertig eingeben, prüfen und simulieren, damit hier alles richtig läuft.

Grundsatz: Positive Signale führen zu Ruderbewegungen nach oben bzw nach rechts

Erst dann, wenn die „Vermischung“ eines Kanals komplett fertig ist, werden am Modell die Bewegung mit Servo Reverse an die reale Bewegungs-Welt angepasst.

Alles andere ist nur Beiwerk:

Telemetrie, Ansagen, Hubi, Flugphasen, globale Variablen, Bilder, Wav-Dateien, Kurven

Übe, teste und simuliere mit companion9x, dann ist die eigentliche Bedienung des Senders ganz einfach.

Alles läuft immer mit dem EVA-Prinzip gleich ab.

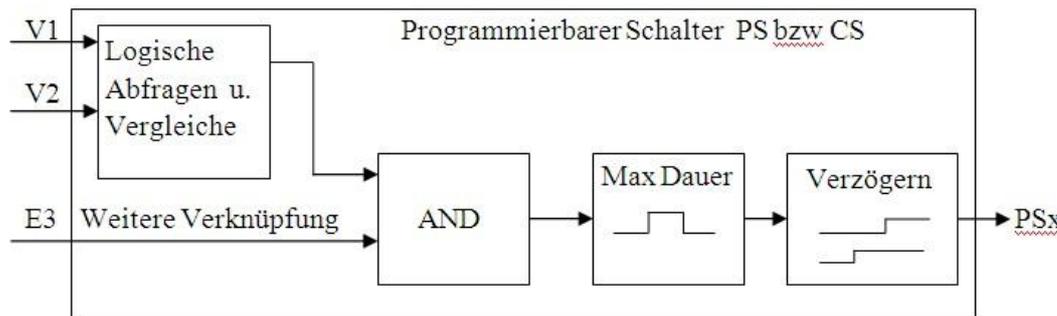
Beispiel: Logische. Schalter mit logischen Verknüpfungen und Abfragen

Merke: Das ist alles das selbe, je nach Softwarestand LSx = PSx = CSx

Es gibt 32 logische Schalter LSx, programmierbare SchalterPSx, Custom Switch CSx die mit Abfragen und Verknüpfungen arbeiten.

Jeder logische. Schalter hat 2 Vergleichs-Eingänge V1 und V2 sowie einen weiteren 3. Eingang als UND-Verknüpfung, denn man belegen kann, aber nicht muss.

Danach kann noch eine max Zeitdauer und eine Verzögerung eingeben werden (bis 15s) Stehen dort die Werte 0,0 sind sie nicht wirksam und der Ausgang PSx bringt solange eine „1“, solange die logische Verknüpfung wahr ist.

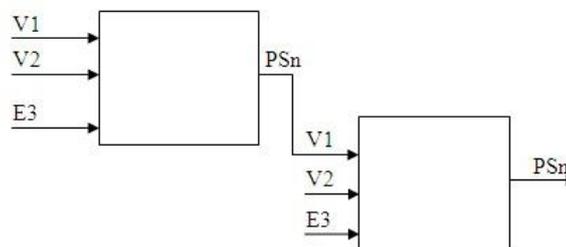


Funktion	Funktion	Funktion	V1	V2	AND	Duration	Verzögerung
----	XOR	CS1	a~x	Thr	----	0,0	0,0
a~x	a=b	CS2	a>x	Cnsp	----	0,0	0,0
a>x	a!=b	CS3	OR	CS1	SH↑	3,0	1,0
a<x	a>b	CS4	TIM	0,5	SB↓	0,0	0,0
a >x	a<b	CS5	----	----	----	0,0	0,0
a <x	a>=b						
AND	a<=b						
OR	d>=x						
XOR	d >=x						
a=b	TIM						

Log. Schalter können beliebig kaskadiert und damit weitere kompliziertere log. Verknüpfungen, Abfragen, Freigaben aufgebaut werden. Auch eine RS-Flip- Flop Funktion mit Set und Reset ist damit möglich. Somit hat man hier eine frei programmierbare Logikbaugruppe deren Ergebnis wieder überall verwendet werden kann.

Beispiele von Kaskadierung:

Log. Schalter können sich auch selber aufrufen. Damit wird ein RS-Flip-Flop realisiert



Das RS-Flip- Flop sieht dann so aus:

L2 = (L1 OR L2) AND SH↑

L2 ist der Ausgang des Flip-Flop

Mit **L1** wird es gesetzt (wenn die Bedingung für CS1 erfüllt ist, ein Impuls reicht aus)

Mit **SH↓** wird es rückgesetzt (wenn der Taster SH kurz betätigt wird, er steht normal in SH↑)

Ab opentx2.0 gibt es auch ein SR-Flip-Flop SRFF als Softwarefunktion

Beispiel: Fahrwerks Door Sequenzer Variante 1

Verblüffend einfach!

Kanal 6 sind die Fahrwerksklappen, Kanal 7 sind die Fahrwerke

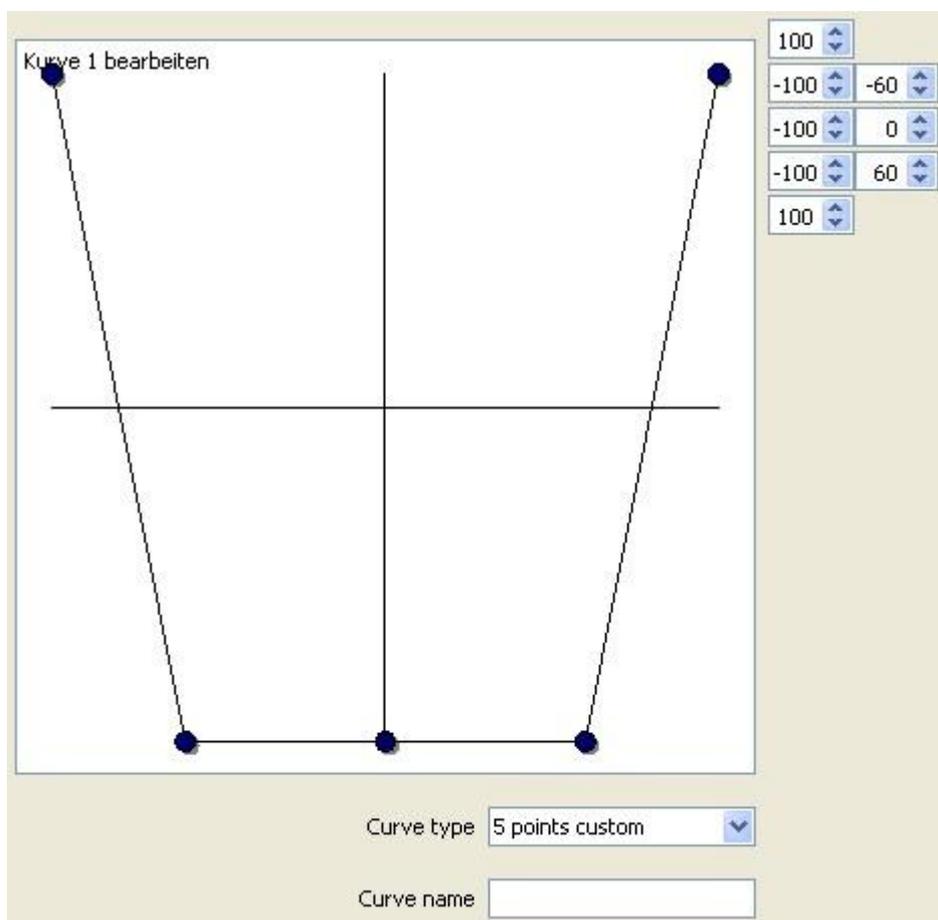
Schalter SF aktiviert den Ablauf

Mit 5 Punktkurve, den 2 Mischerzeilen und etwas Verzögerung und Langsam

SF betätigen, Klappen gehen auf, Fahrwerk fährt aus, Klappen gehen zu

SF betätigen, Klappen gehen auf, Fahrwerk fährt ein, Klappen gehen zu.

```
CH05  
CH06      (-100%) SF Kurve (Kurve 1) Langsam/u8:d8)  
CH07      (+100%) SF Verzögerung (u2:d2) Langsam/u2:d2)  
CH08
```



Beispiel: Fahrwerks Door Sequenzer Variante 2

Mit nur 3 Zeilen Logik in den Logischen Schaltern ein kompletter Door-Sequenzer

SF ist der Fahrwerksschalter

CH16 ist ein Hilfskanal der bei bestimmten Positionen die Doors und Gears auslöst

CH15 ist die Fahrwerksklappe

CH14 ist das Fahrwerk

Grundstellung

SF = off = unten

CH14,CH15 CH16 bei -100%

SF = ON = Oben Klappe auf, Fahrwerk Aus, Klappe zu

CH16 läuft langsam von -100% nach +100%

bei -90% von Ch16 läuft die Klappe auf

bei 0% von CH16 fährt das Fahrwerk aus

bei +90% von CH16 läuft die Klappe wieder zu

SF = OFF= Unten Klappe auf, Fahrwerk Ein, Klappe zu

CH16 läuft langsam von +100 nach -100%

bei +90% von CH16 öffnet die Klappe

bei 0% von CH16 fährt das Fahrwerk ein

bei -90% von CH16 schließt die Klappe

Die Logik steckt in den 3 Zeilen der

Programmierbaren Schaltern:

CS1 a>x CH16 -90 AND CS3

CS2 a>x CH16 0

CS3 a<x CH16 90

	Funktion	V1	V2	AND
CSw1	a>x	CH16	-90	CS3
CSw2	a>x	CH16	0	----
CSw3	a<x	CH16	90	----

Mischer

CH14 +100% CS2 Slow(u3:d3)

CH15 +100% CS1 Slow(u3:d3)

CH16 +100% SF Slow(u10:d10)

CH13	
CH14	(+100%) CS2Langsam/ u3: d3)
CH15	(+100%) CS1Langsam/ u3: d3)
CH16	(+100%) SFLangsam/ u10: d10)
CH17	

Die tatsächlich benötigten Servo-Wege für Fahrwerk und Door kann völlig unabhängig eingestellt werden da sie in den Kanälen 14 und 15 sind und die eigentliche zeitliche Steuerung über CH16 läuft.

Ist doch verblüffend einfach.

Beispiel: V-Leitwerk Ruderwirkung und Mischer

Ruderrichtung - Sinnrichtiger Ausschlag

Blick von hinten auf das Leitwerk - Rumpf in Flugrichtung

Höhenruder:

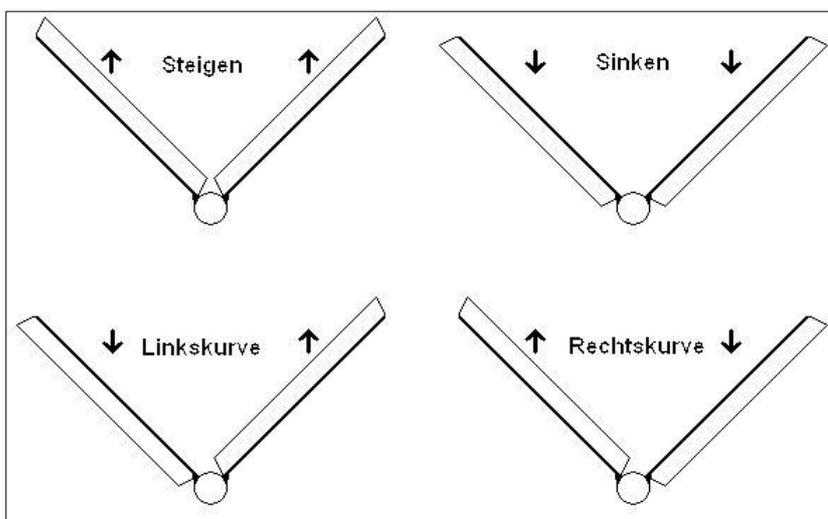
Höhenruder ziehen - beide Klappen gehen gleichsinnig nach oben.

Höhenruder drücken - beide Klappen gehen gleichsinnig nach unten.

Seitenruder:

Seitenruder rechts - beide Klappen gehen nach rechts.

Seitenruder links - beide Klappen gehen nach links.



Konfiguration	Hubschrauber Setup	Flight Modes	Sticks	Mischer
<input type="checkbox"/> Show channels names in mixes				
CH01	(+100%) Thr			
CH02	(+100%) Ail (Quer Re)			
CH03	(+100%) CH12 (VLeit Re)			
	(-100%) Rud			
CH04	(+100%) CH12 (VLeit Li)			
	(+100%) Rud			
CH05	(-100%) Ail (Quer Li)			
CH06				
CH07				
CH08				
CH09				
CH10				
CH11				
CH12	(-100%) Ele (Ele Inv)			

Beispiel: Mischer Querruder auf V-Leitwerk mischen.

Ich habe eine Graupner-Kanalbelegung, CH1-CH5= GQHS Q

CH1 Gas

CH2 Quer1 rechts, mit 30% Differenzierung

CH5 Quer2 links, mit 30% Differenzierung

CH3, CH4 V-Leitwerk, Höhe und Seite gemischt,
Höhen laufen gleichsinnig, Seiten laufen gegensinnig

CH12 nur wg der Schönheit der Programmierung und der positiven Logik
Höhe invertiert, damit beim Ziehen pos Signale kommen, vereinfacht das Mitdenken

Normales V-Leitwerk CH3 CH4 als Ausgangsbasis

Konfiguration	Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Sp
CH1				Gas Gewichtung(+100%)				
CH2				Que Gewichtung(+100%) Diff(30%) (QR recht)				
CH3				CH12 Gewichtung(+100%) (VL recht) Sei Gewichtung(-100%)				
CH4				CH12 Gewichtung(+100%) (VL links) Sei Gewichtung(+100%)				
CH5				Que Gewichtung(-100%) Diff(30%) (QR links)				
CH6								
CH7								
CH8								
CH9								
CH10								
CH11								
CH12				Höh Gewichtung(-100%) (Hoh_Inv)				
CH13								

Vergleiche Bild V-Leitwerk im Handbuch und Ruderbewegungen

Normales V-Leitwerk CH3 CH4, aber jetzt mit Mischer 25% Quer --> auf V-Leitwerk

Konfiguration	Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter
CH1				Gas Gewichtung(+100%)			
CH2				Que Gewichtung(+100%) Diff(30%) (QR recht)			
CH3				CH12 Gewichtung(+100%) (VL recht) Sei Gewichtung(-100%) Que Gewichtung(-25%) (QR VL re)			
CH4				CH12 Gewichtung(+100%) (VL links) Sei Gewichtung(+100%) Que Gewichtung(+25%) (QR VL li)			
CH5				Que Gewichtung(-100%) Diff(30%) (QR links)			
CH6							
CH7							
CH8							
CH9							
CH10							
CH11							
CH12				Höh Gewichtung(-100%) (Hoh_Inv)			
CH13							

Achtung:

Die Werte der Gewichtungen muss man noch anpassen!

Die Servolaufrichtungen am realen Modell nur im Servomenü anpassen, da die Vermischung schon richtig rechnet.

Merke:

Positive Knüppelwerte erzeugen Ruderbewegungen nach oben oder rechts

Wenn ich Quer rechts gebe, geht das rechte QR nach oben und leitet ein Rechtskippen ein

Wenn ich Seite rechts gebe, geht das rechte V-Ruder nach rechts/unten und leitet eine Rechtskurve ein.

Vergleiche Bild V-Leitwerk im Handbuch und Ruderbewegungen

Bitte mal simulieren!

Beispiel: Mischer Quer auf Seite Mischer Seite auf Quer

Quer --> Seite mischen: Wenn man Quer gibt, soll Seite zusätzlich mitgehen
 (Graupner)-Kanalbelegung, CH1-CH5 = GQHS Q

Im Kanal für Seite (hier CH4) eine zusätzliche Mischerzeile einfügen.
 Quelle ist der Querruder Stick oder Querruder Input, Gewichtung ca 20-30%
 Gewichtung mit +/- seitenrichtig dazumischen, so verblüffend einfach ist das.

Gibt man am Knüppel Quer rechts, folgt Seite rechts mit 25% Anteil (addierend)

Konfiguration	Heli	Flugphasen	Inputs	Mischer	Servos
CH1			[I1]Gas	Gewichtung(+100%)	
CH2			[I2]Que	Gewichtung(+100%)	
CH3			[I3]Höh	Gewichtung(+100%)	
CH4			[I4]Sei	Gewichtung(+100%)	
			[I2]Que	Gewichtung(+25%)	
CH5			[I2]Que	Gewichtung(-100%)	
CH6					

Seite --> Quer mischen: Wenn man Seite gibt soll Quer zusätzlich mitgehen

Auf beide Querruder Kanäle jeweils 25% Seite dazumischen (hier bei CH2 und CH5)
 Gewichtung mit +/- seitenrichtig dazumischen

Konfiguration	Heli	Flugphasen	Inputs	Mischer	Servos
CH1			[I1]Gas	Gewichtung(+100%)	
CH2			[I2]Que	Gewichtung(+100%)	
			[I4]Sei	Gewichtung(+25%)	
CH3			[I3]Höh	Gewichtung(+100%)	
CH4			[I4]Sei	Gewichtung(+100%)	
CH5			[I2]Que	Gewichtung(-100%)	
			[I4]Sei	Gewichtung(-25%)	
CH6					

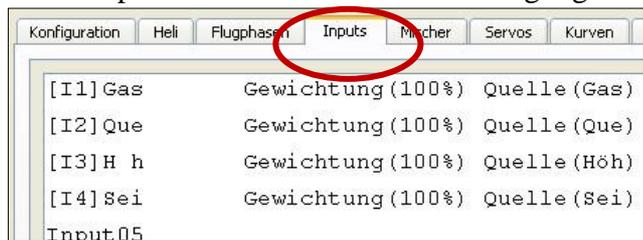
Achtung:

Die Werte der Gewichtungen muss man noch anpassen!

Die Servolaufrichtungen am realen Modell nur im Servomenü anpassen,
 da die Vermischung schon richtig rechnet.

Beispiel: Mischer Seite auf Höhenruder soll immer etwas Tiefe bringen

Am Beispiel meiner Standard Kanalbelegung für Kanal 1-4 Gas Quer Höhe Seite, wie bei Graupner



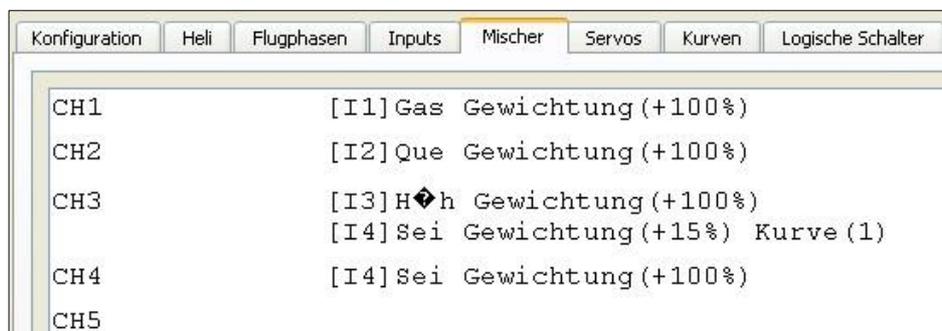
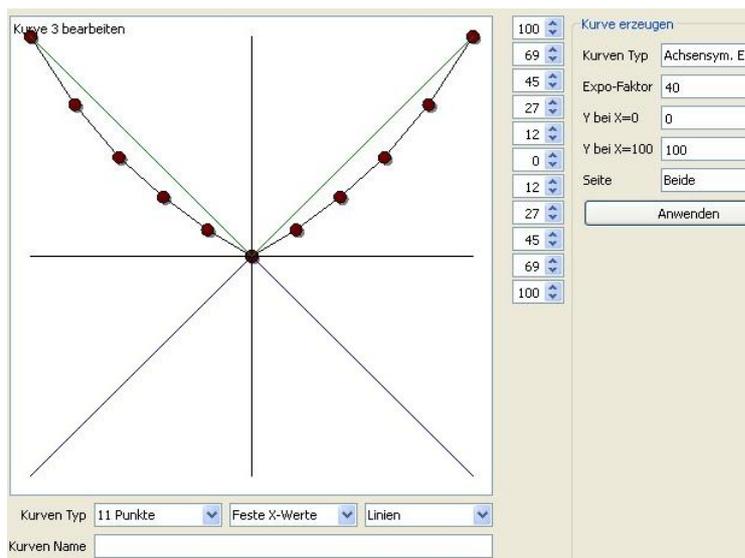
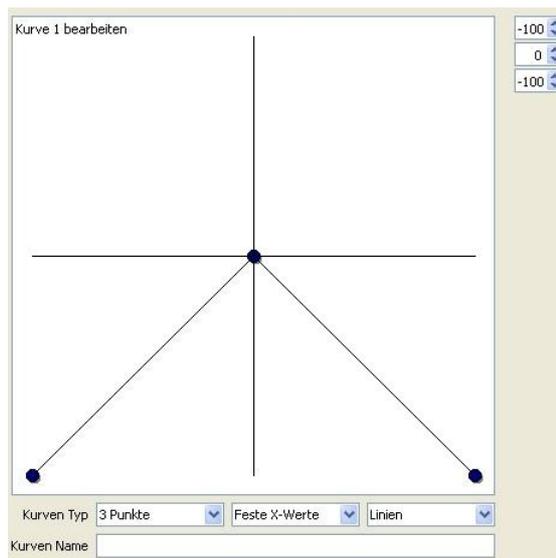
Egal ob man Seite rechts/links (positiv/ negativ) gibt, es soll immer ca 15% Tiefe gemischt werden.

Dazu gibt es min 3 - 4 Möglichkeiten

1. Mit einer negative V-Kurve / Expo-Kurve oder auch positiver V-Kurve / Expo-Kurve
2. Mit einer Inputs Signal-Vorverarbeitung des Seitenruderknüppels und der Betragsfunktion $|x|$
3. Direkt in den Mischern mit 2 Funktionen $x < 0, x > 0$
- 3a Wie bei 2. aber direkt im Mischer, mit einer Betragsfunktion $|x|$ und einer negativen Gewichtung

Variante 1: negative V-Kurve und Mischer oder positive V-Kurven / Expokurven

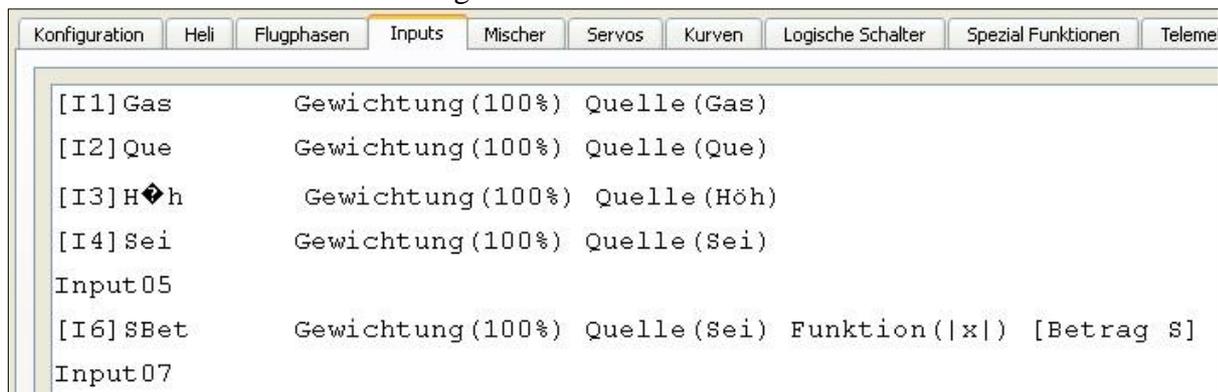
Man erzeugt sich einfach eine passende Kurve, die man dann im Mischer aufruft.



Da die Kurve 1 schon negativ ist, muss im Mischer CH3 die Gewichtung +15% eingestellt werden. Ansonsten könnte man auch eine positive V-Kurve nehmen, dann im Mischer -15% nehmen. Man kann auch V-förmige Expokurven (Kurve 3) erzeugen.

Variante 2: Mit einer Input Signalvorverarbeitung

Wir verarbeiten das Signal des Seitenruderknüppels zusätzlich mit einer Betragsfunktion $F|x|$. Dadurch erhalten wir immer positive Signale. Diese Signal (hier I6) mischen wird dann dem Höhenruderkanal dazu. Gewichtung mit -15%



Variante 3: direkt im Mischer mit 2 zusätzlichen Mischerzeilen $X>0$ und $X<0$



Hier wird das Seitenrudersignal in den Inputs nur durchgereicht, keine extra Vorverarbeitung. Die Anpassung erfolgt direkt im Mischer CH3 Höhenruder mit 2 zusätzlichen Zeilen. Wenn der Wert positiv ist ($x>0$) wird die Mischerzeile mit -15% aktiv. (plus * minus = minus) Wenn der Wert negativ ist ($x<0$) wird die Mischerzeile mit +15% aktiv (minus * plus = minus)

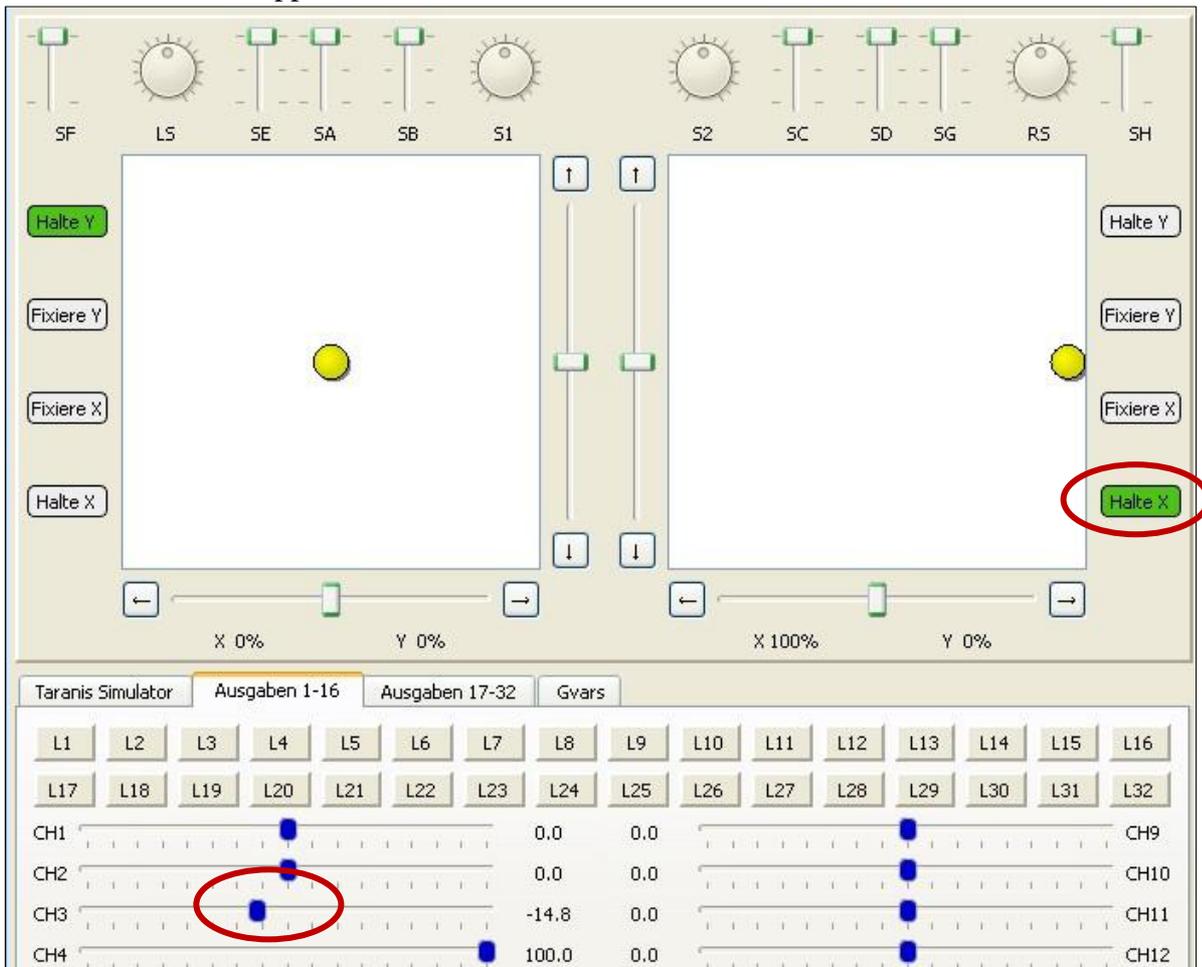
Variante 3a:

Das gleiche wie in Variante 2 kann man auch gleich im Mischer machen, Mit der Betragsfunktion $|x|$ den Seitenruderknüppel immer ins positive bringen, dann mit -15% dazumischen.



Tipp für die Simulation:

Mit Halte X den Knüppel fixieren, dann sieht man besser was läuft



Beispiel: Dynamische Servo Geschwindigkeit mit einem Integral-Mischer

z.B. für Kamera-Schwenksteuerung angepasst an Taranis

Wenn der Knüppel aus der Mitte bewegt wird soll das Servo folgen,
 Wenn der Knüppel wieder in der Mitte steht soll das Servo stehen bleiben wo es gerade ist.
 Wird der Knüppel schnell bewegt soll das Servo auch schnell folgen

Prinzip:

Der Kanal CH01 ruft sich selber mit CS1 auf wenn der Unterschied $|a| > x$ Ele > 2 ist und addiert zu seinem aktuellen Wert 3% dazu, ansonsten bleibt er dort stehen wo er ist. CS2 und CS3 überwachen nur die Grenzen und setzen +100% bzw -100% fix, (dann kann man auch auf andere Werte begrenzen)

Die 1 Kanal Ausführung nur mit Elevator Knüppel

```
CH01      (+100%)CH01 No Trim
          (+3%)Ele Schalter(CS1) No Trim Kurve(Kurve 9)
          R (+100%)MAX Schalter(CS2) No Trim
          R (-100%)MAX Schalter(CS3) No Trim
CH02
```

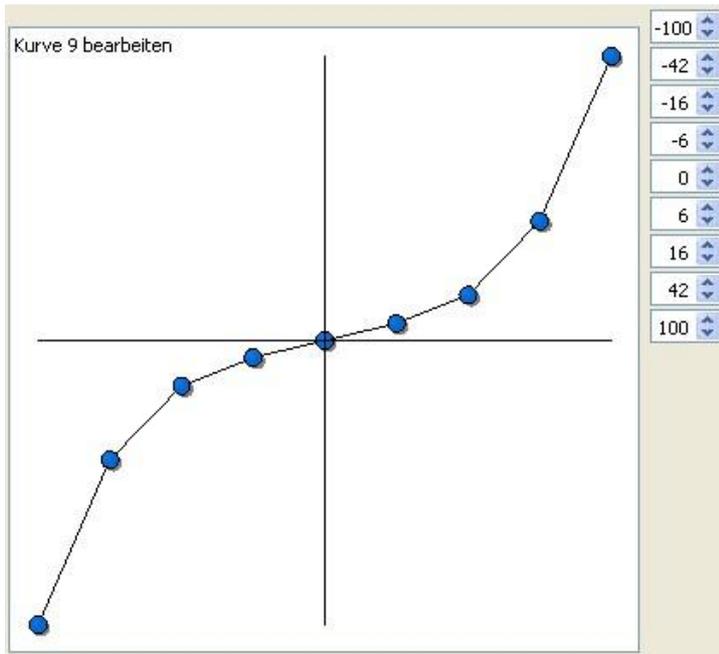
	Funktion	V1	V2
CSw1	$ a > x$	Ele	2
CSw2	$a > x$	CH01	100
CSw3	$a < x$	CH01	-100

Die 2 Kanal X/YAusführung Elevator und Ruder Knüppel

```
CH01      (+100%)CH01 No Trim
          (+3%)Ele Schalter(CS1) No Trim Kurve(Kurve 9)
          R (+100%)MAX Schalter(CS2) No Trim
          R (-100%)MAX Schalter(CS3) No Trim
CH02      (+100%)CH02 No Trim
          (+3%)Rud Schalter(CS4) No Trim Kurve(Kurve 9)
          R (+100%)MAX Schalter(CS5) No Trim
          R (-100%)MAX Schalter(CS6) No Trim
CH03
```

	Funktion	V1	V2
CSw1	$ a > x$	Ele	2
CSw2	$a > x$	CH01	100
CSw3	$a < x$	CH01	-100
CSw4	$ a > x$	Rud	2
CSw5	$a > x$	CH02	100
CSw6	$a < x$	CH02	-100

Geschwindigkeits-Anpassungen über Kurve 9 und über Weight 3% auf 2% oder 1 %



Das ist auch so ein verblüffend einfaches Beispiel mit ein paar Zeilen

Tip:

Wem die 2% oder 3% noch zu schnell sind kann das nochmal mit einem freien Kanal unterteilen.

CH 12 Seite Gewichtung (20%)

und dann anstatt (2%) Seite nimmt man (2%) CH12 in der 2. Zeile

Dann hat man 2% von 20% = 0,4% Auflösung

Logische Schalter zur Überwachung von Mitte und Grenzen mit Betragsfunktion

L1: $|a| > x$ Sei 2 erst wenn Stick aus der Mitte bewegt wird, dann aktiv

L2: $|a| < x$ CH10 100 (-100 und +100) min / max Grenzen überwachen

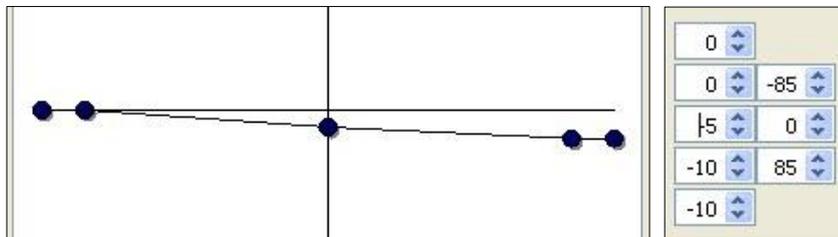
L1	$ a > x$	Sei	2
L2	$ a < x$	CH10	100

CH12 mit 20% von Seite, CH10 mit 2 % von CH12 damit = 0,4%

CH10	CH10 Gewichtung (+100%)
	CH12 Gewichtung (+2%) Schalter (L1)
CH11	
CH12	Sei Gewichtung (+20%)

Beispiel: Mischer Gas auf Höhe mit Kurve, ohne Kurve

Auf das Höhenruder soll gasabhängig etwas Tiefe (5-10%) dazugemischt werden. Das Zumischen soll aber nur im Bereich von -85% bis +85% Gasstellung erfolgen. (Im Gegensatz dazu würde eine normale Zumischung im Bereich -100% bis +100% erfolgen) Dazu verwenden wir einfach eine frei einstellbare 5-Punkt-Kurve, X / Y-Wert frei einstellbar



- Wenn das zu viel oder zu wenig ist, kann man:
- die Kurve ändern, steiler, flacher oder
 - die Gewichtung (Weight) in der Mischerzeile anpassen

CH01	(+100%) Thr
CH02	(+100%) Ail
CH03	(-100%) Ele
CH04	(+100%) Thr Kurve (Kurve 1)
CH05	(+100%) Rud

Hinweis Höhe CH3:
 -100% Ele weil der Höhen-Knüppel beim ziehen negative Werte liefert.
 +100% Thr weil die Kurve definiert dass es -10% ins Negative geht

Alternative: Die „normale“ Tiefen-Zumischung ohne Kurve

Dieser Mischer erzeugt genau das Gleiche, Tiefenzumischung von bis zu -10% aber im Gas-Bereich -100% bis +100%

CH01	(+100%) Thr
CH02	(+100%) Ail
CH03	(-100%) Ele
	(-5%) Thr Offset (100%)
CH04	(+100%) Rud
CH05	

Hinweis Höhe CH3:
 -100% Ele weil der Höhen-Knüppel beim ziehen negative Werte liefert.
 -5% Thr und Offset 100%, weil die Mischerberechnung so läuft:
 $[(\text{Quelle} * \text{Gewichtung}) + \text{Offset}] * \text{Kurve}$

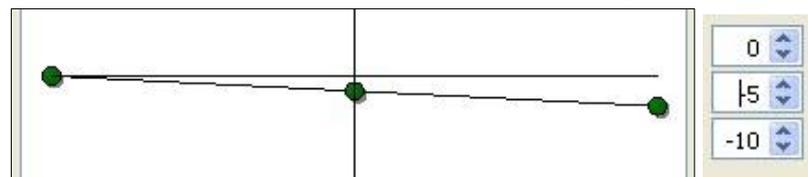
Berechnung:

Gesamter Thr-Bereich = -100% bis +100% = 200

10 von 200 = 0,05 = +5%, soll aber ins negative = -5%

Thr min: $(-100\% + 100\%) = 0 * -5\% = 0\%$ **Thr max:** $(+100\% + 100\%) = 200 * -5\% = -10\%$

Und so würde das wieder als Mischer mit einer 3-Punkt-Kurve aussehen.



CH03	(-100%) Ele
	(+100%) Thr Kurve (Kurve 2)
CH04	(+100%) Rud

Beispiel: Mischer Gas auf Höhe mit Kurve aber variabel zumischen per S1 und GVAR

Das ist eine Erweiterung zum vorigen Beispiel. Die Tiefenzumischung per Kurve soll nicht fest, sondern mit S1 variabel einstellbar sein.

Mischkurve definieren:

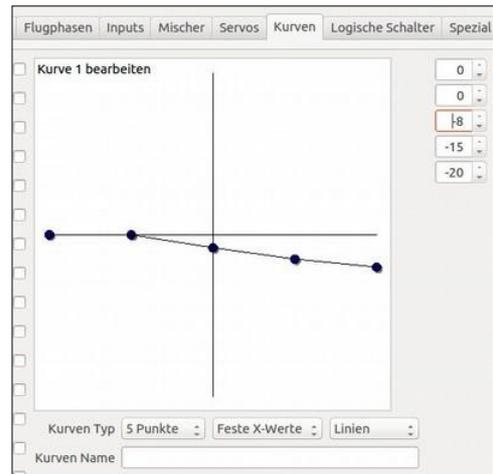
Es soll ab einem bestimmten Gaswert (hier ab -50%)

Etwas Tiefe auf das Höhenruder gemischt werden, wenn man Gas gibst.

Die Kurve 1 hab ich einfach mal so eingegeben, damit man deutlich sieht wie das geht.

Meist reicht schon -5 bis -8% Zumischung.

Also Kurve selber anpassen und aufpassen!



Poti S1 anpassen:

S1 läuft von -100% bis +100%

das musst man erst mal so vorverarbeiten, dass nur noch

0 bis +100% rauskommt. Wird in Kanal10 gemacht.

Poti S1 als Quelle bringt: -100 bis +100 = 200

Gewichtung CH10 soll: 0 bis 100 = 100 → $100/200 = 50\%$

Offset: Die Mitte den neuen Bereich 0 bis 100 = **50%**

Globale Variable GV1 mit CH10-Wert versorgen:

diesen Kanal 10 gebe als Quelle der GVAR1 in den Spezialfunktionen ein.

Damit geht GVAR1 von 0 bis +100% wenn Poti S1 gedreht wird.

Schalter	Aktion	Parameter	Aktivieren
SF1 EIN	Adjust GV 1	Quelle CH10	<input checked="" type="checkbox"/> EIN
SF2 —	Override CH1	0	<input type="checkbox"/> EIN

Gas auf Höhe mischen:

CH3 Höhenruder hat 2 Zeilen

1. ganz normal vom Höhenknüppel via Input Höhe [I3]Höh

2. vom Gasknüppel via Input die Tiefenzumischung

Quelle: Input Gas [I1]Gas

mit Kurve 1 verarbeiten

Gewichtung GV1 von 0 bis 100%

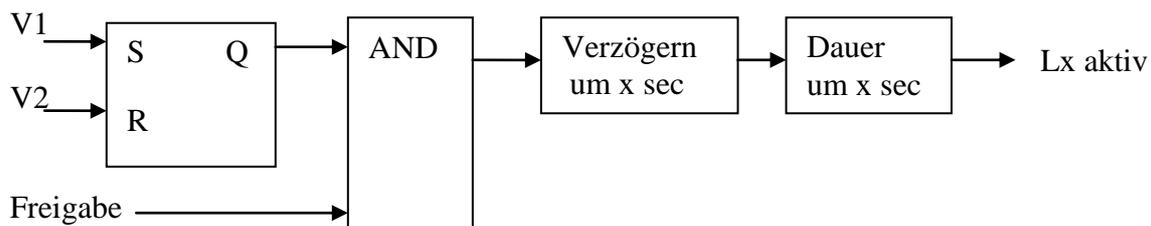
einstellbar mit S1

CH	Definition
CH1	[I1]Gas Gewichtung(+100%)
CH2	[I2]Que Gewichtung(+100%)
CH3	[I3]Höh Gewichtung(+100%)
CH4	[I1]Gas Gewichtung(GV1) Kurve(1)
CH5	
CH6	
CH7	
CH8	
CH9	
CH10	S1 Gewichtung(+50%) Offset(50%)

Jetzt kann man die Kurve noch beliebig anpassen und davon dann per GVAR 0 bis 100% zur Höhe dazumischen. Damit kann man den tatsächlich benötigten Wert erfliegen und dann in der Gewichtung als Festwert eingeben.

Achtung: obige Kurve1 mischt max -20% dazu, das ist ein Spielbeispiel damit man etwas sieht meist reichen schon ca -5% schon aus.

Beispiel: Logische Schalter Funktion SRFF = SR-FlipFlop mit Bedingungen



SRFF ist eine neue universelle Flip Flop Funktion, die mehr kann als die bisherige Toggle-Funktion mit dem vorangestellten „t“ und wird ihn ersetzen.

Das Flip-Flop wird durch einen kurzen Impuls gesetzt und durch einen anderen kurzen Impuls wieder rückgesetzt. V1= Setz, V2= Rücksetz (Reset)

Das Flip-Flop kann noch durch ein Freigabesignal gesperrt/freigegeben werden
Der Ausgang Lx ist so lange aktiv bis das FlipFlop einen Resetimpuls erhält, oder die Freigabe weggenommen wird (führt auch zu einem Reset).

Falls eine Verzögerung und/oder Dauer eingegeben werden folgt:

Der Ausgang Lx kann um bis zu 15 Sekunden verzögert werden bis er aktiv wird

Die Dauer kann auf bis 15 s Sekunden eingestellt werden,

Ist die Dauer abgelaufen wird das Flip- Flop automatisch zurückgesetzt (Vorrang)

Setzen und Rücksetzen kann auch mit dem gleichen Impulsgeber erfolgen
damit haben wir ein T-Flip-Flop (Toggle Flip-Flop)

Als Impulseingabe und Freigabe können alle Arten von Schaltern und Schalterstellungen verwendet werden. (Physikalischer Schalter, Logische Schalter auch mit 3 Stellungen).

Liegen Setz- und Rücksetz gleichzeitig an hat Rücksetz den Vorrang.

L1 SRFF SA↓ SA↓

L1 wird durch SA gesetzt und rückgesetzt

L2 SRFF SB↓ SC↓

L2 wird durch SB gesetzt und SC rückgesetzt

L3 SRFF SH↓ Dauer 5s

L3 wird durch SH gesetzt und nach 5s autom. rückgesetzt

L4 SRFF SH↓ Dauer 3s Verzög 2s

L4 wird durch SH gesetzt, muss aber min 2s anstehen, ist dann 3s an und wird dann autom. rückgesetzt

Beispiel: Logische Schalter Funktion Puls (Edge) einen einmaligen Impuls erzeugen

Ein einmaliger Impuls kann erzeugt werden mit der Funktion Puls (Edge), ein Monoflop.

Das ersetzt z.B. die Short-und Long-Funktion des SH-Tasters

Am Beispiel log Schalter LS1:

Schalter SA wird für max 0,7s betätigt, dann wird ein einmaliger Impuls für 5 s Dauer erzeugt.

L1 Puls [0,0 : 0,7] SA↓ Duration 5,0

L2 Puls [1,0 : 1,0] SH↓ Taster SH muss min 1 sec betätigt sein

L3 Puls [1,0 : 2,5] SH↓ Taster SH muss zwischen 1 und 2,5sec betätigt sein

L4 Puls [0,0 : 0,6] SH↓ Taster SH darf nur max 0,6 sec betätigt sein

Wird keine Dauer (Duration) angegeben erfolgt nur ein sehr kurzer Impuls (Rechenzyklus ca 10ms)

SH↓ SH↓s long und short ersetzen:

L5 Puls [0,0 : 0,4] SH↓ das ersetzt den SH↓s short

L6 Puls [0,8 : 0,8] SH↓ das ersetzt den SH↓l long

Beispiel: Logische Schalter Funktion Takt = TIM ist ein Taktgenerator

Ein Taktgenerator mit On- und Off-Zeiten Takt (bzw TIM)

L3 SB↓ Takt 0,5 0,2 einstellbares Taktverhältnis 0,5 s Ein + 0,2s Aus = 0,7s Periode

Da zu gibt es weitere Beispiele mit Anwendungen

Beispiel: Logische Schalter Funktion Range = einen Analogwert als Bereich abfragen

(Noch nicht implementiert)

Ein Analogwert kann in einem Bereich abgefragt werden

L2 S2 Range -35 +45

Beispiel Bereichsabfrage (Alternative zu Range)

L2 a>x S1 15

L3 a<x S1 37 UND L2 (AND Switch)

L3 wird aktiv im Bereich von 15 bis 37

Beispiel: Flip Flop mit SH-Taster für Set und Reset

Hier mal ein programmierbarer Schalter als Flip Flop das mit SH gesetzt und rückgesetzt wird

Was im ersten Augenblick aufwändig erscheint, damit kann man auch zählen von, bis, ab
(Beim Flip Flop zähle ich halt nur bis 2 und resete dann den CS)

Wenn man anstatt des SH den Taktgenerator **Takt** (bisher **TIM**) verwendet, haben wir
Zeitrelais, Einschaltverzögert, Abfallverzögert, ImpulsRelais mit einstellbarer Impulsbreite von bis usw.
bei entsprechender Abfrage der CS

CF4	SH1	Adjust GV2	Increment	+1	<input checked="" type="checkbox"/>	ON
CF5	CSB	Adjust GV2	Wert	0	<input checked="" type="checkbox"/>	ON
CF6	----	Safety CH01	0		<input type="checkbox"/>	ON
CS9	----	----	0	----	0,0	0,0
CSA	a~x	GV2	1	----	0,0	0,0
CSB	a~x	GV2	2	----	0,0	0,0

Dazu gibt es sehr umfangreiche Ergänzungen und Beispiel für Vorwärts, Rückwärts, Reset bei Wert

Beispiel: Stufenschalter mit SH und globalen Variablen nachbilden

Das Prinzip läuft immer gleich: In den Spezialfunktionen gibt es die Funktion Increment +/- 1 Also den Wert einer GVAR immer um 1 ändern. Das können wir z.B. mit dem Tastschalter SH oder aber auch mit dem Taktgenerator **Takt** (bzw TIM) erzeugen

Schalter	Funktion	Parameter	aktivieren
CF1	SH↓	Adjust GV1	Increment +1 <input checked="" type="checkbox"/> ON
CF2	CS7	Adjust GV1	Wert 1 <input checked="" type="checkbox"/> ON
CF3	----	Safety CH01	0 <input type="checkbox"/> ON

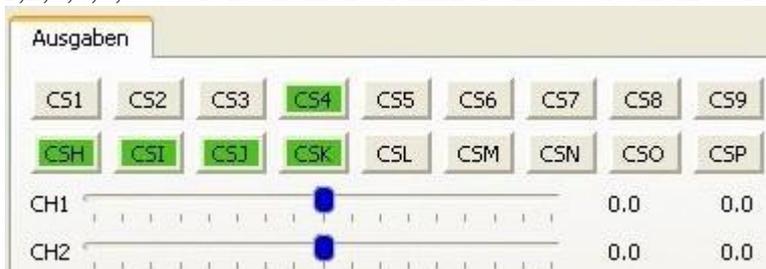
Den Wert der GV1 wird in den Log. Schaltern abgefragt/verglichen mit **a~x** oder **a>x**. Dadurch wird entweder genau ein Prog. Schalter aktiv oder mehrere Prog. Schalter aktiv

Funktion	V1	V2	AND	Duration	Verzögerung	
CS1	a~x	GV1	1	----	0,0	0,0
CS2	a~x	GV1	2	----	0,0	0,0
CS3	a~x	GV1	3	----	0,0	0,0
CS4	a~x	GV1	4	----	0,0	0,0
CS5	a~x	GV1	5	----	0,0	0,0
CS6	a~x	GV1	6	----	0,0	0,0
CS7	a~x	GV1	7	----	0,0	0,0
CS8	----	----	0	----	0,0	0,0
CS9	----	----	0	----	0,0	0,0

Funktion	V1	V2	AND	Duration	Verzög	
CSH	a>x	GV1	0	----	0,0	0,0
CSI	a>x	GV1	1	----	0,0	0,0
CSJ	a>x	GV1	2	----	0,0	0,0
CSK	a>x	GV1	3	----	0,0	0,0
CSL	a>x	GV1	4	----	0,0	0,0
CSM	a>x	GV1	5	----	0,0	0,0
CSN	a>x	GV1	6	----	0,0	0,0
CSO	----	----	0	----	0,0	0,0
CSP	----	----	0	----	0,0	0,0

Dann müssen wir ab einem bestimmten Vergleichswert (hier bei GV=7) die Globale Variable wieder auf den Startwert setzen. Dazu fragen wir den Log. Schalter ab CS7 a~x GV1 7 und setzen damit in den Spezialfunktionen CF2 CS7 die GV1 wieder auf 1.

Damit haben wir jetzt eine Art Stufenschalter der mit SH von 1-6 zählt. Entweder mit einzelnen Stufen 1,2,3,4,5,6 oder nacheinander immer eine Stufe mehr dazuschaltet.



Bitte das Beispiel eingeben und simulieren, damit man den Ablauf versteht!

Dieses Stufenbeispiel kann man beliebig ausbauen z.B. 10 Stufen für LED Light-Ccontroller

Was kann man jetzt damit machen? Alles was mit Log. Schalter möglich ist!

z.B. In Mischern nacheinander Werte setzen, die per Log. Schalter aktiviert werden
per Replace für die einzelnen oder per Addiere für mehrere

LED-Light Controller umschalten oder APM-Mode setzen

Beispiel: Mischer im Kanal 6 per Replace setzt feste Ausgangswerte per Taster SH

```
CH6          MAX Gewichtung (-100%)  Schalter (L1)
              R MAX Gewichtung (-65%)  Schalter (L2)
              R MAX Gewichtung (-35%)  Schalter (L3)
              R MAX Gewichtung (+10%)  Schalter (L4)
              R MAX Gewichtung (+70%)  Schalter (L5)
              R MAX Gewichtung (+100%) Schalter (L6)
```

Achtung/Tip:

Die GV1 **kann/ muss** man mit einem Startwert vorbelegen hier z.B. GV1=1,
damit ist L1 schon aktiv beim Aufruf des Modells und im Kanal steht ein passender Wert schon an.

GVAR1	<input type="text"/>	1	<input checked="" type="checkbox"/> Anzeige im Fenster
GVAR2	<input type="text"/>	0	<input type="checkbox"/> Anzeige im Fenster
GVAR3	<input type="text"/>	0	<input type="checkbox"/> Anzeige im Fenster

Beispiel: Automatisch verschiedene Telemetriewerte ansagen lassen.

	Schalter	Aktion	Parameter	Aktivieren
SF1	SH ↓	Adjust GV 1	Increment +1	<input checked="" type="checkbox"/> EIN
SF2	L7	Adjust GV 1	Wert 0	<input checked="" type="checkbox"/> EIN
SF3	---	Safety CH1	0	<input type="checkbox"/> EIN
SF4	L1	Sag Wert	RX Batt	Keine Wiederholung
SF5	L2	Sag Wert	RSSI RX	Keine Wiederholung
SF6	L3	Sag Wert	Strom	Keine Wiederholung
SF7	L4	Sag Wert	Leistung	Keine Wiederholung
SF8	L5	Sag Wert	Verbrauch	Keine Wiederholung
SF9	L6	Sag Wert	Zellen	Keine Wiederholung
SF10	---	Safety CH1	0	<input type="checkbox"/> EIN

	Funktion	V1	V2	UND Schalter
L1	a~x	GV1	1	---
L2	a~x	GV1	2	---
L3	a~x	GV1	3	---
L4	a~x	GV1	4	---
L5	a~x	GV1	5	---
L6	a~x	GV1	6	---
L7	a~x	GV1	7	---

Wenn anstatt dem SH-Taster der Taktgenerator verwendet wird kommen alle 3,5s die Ansagen

L9	Takt	2,5	1,0	SA ↓
----	------	-----	-----	------

	Schalter	Aktion	Parameter	Aktivieren
SF1	L9	Adjust GV 1	Increment +1	<input checked="" type="checkbox"/> EIN
SF2	L7	Adjust GV 1	Wert 0	<input checked="" type="checkbox"/> EIN

Beispiel: Mit SRFF Flip-Flop und SH Log Daten Aufzeichnung Start/Stop

SH ist ein Taster. Damit können wir mit einem kurzen Impuls SH↓ das Set/Reset FlipFlop SRFF in den logischen Schalter steuern.

Ein kurzer Impuls von SH↓ setzt das SRFF, der nächste Impuls setzt es zurück, usw. Das nennt man eine Toggle-Funktion

Damit wird ein logischer Schalter L1 solange aktiv wie das SRFF gesetzt ist.

	Funktion	V1	V2	UND Schalter
L1	SRFF	SH↓	SH↓	----
L2	---	----	0	----

Mit diesem logischen Schalter L1 können wir jetzt die Log-Datenaufzeichnung In den Spezialfunktionen starten und stoppen

	Schalter	Aktion	Parameter	Aktivieren
SF1	L1	Start Log	0,1	
SF2	----	Override CH1	0	<input type="checkbox"/> EIN

Beispiel: Flip Flop mit Toggle Funktion Ein/Aus auf Timer anwenden

Das „t“ hinter allen Schaltern ist eine Toggle-Funktion, also ein T-Flip- Flip, das EIN und AUS geschaltet werden kann. („t“ gibt es ab opentx2.0 nicht mehr aber SRFF) Damit kann man auch jeden log. Schalter setzen und reseten

Ein Timer hat die fertigen Funktionen: ABS, GSs, GS% (bzw THs, TH%)
ABS startet einfach den Timer, THs startet und stoppt den Timer sobald Gas > min,
TH% ist eine gasstellungsabhängige Zeit.

Man kann aber einen Timer auch mit einem ganz normalen Schalter starten und stoppen.
Also mit z.B. SA↓= Ein SA↑=Aus

Oder aber mit der Toggle-Funktion und z.B. dem Taster SH
Start mit SHt Stop mit SHt t=Toggle Flip Flop EIN/AUS Funktion

Modell Name	Timerstart		
Stoppuhr1	00:00	ABS	
Stoppuhr2	00:00	SH↓t	

Schalter	Funktion	Parameter
CF1	SH↓t	Reset
		Stoppuhr2

Und mit **SH↓t** also SH „long“ (>1s) betätigen, kann man den Timer wieder zurück auf bzw. Startwert z.B. 03:00 stellen.

Achtung:

Short und Long gibt es ab opentx V2.00 nicht mehr, deshalb mit der Funktion Puls arbeiten!

Beispiel: Flip Flop mit Log. Schalter Lx (CSx) auf Timer anwenden

Einen Timer ab einer best. Gasstellung starten und mit einem anderen Schalter wieder stoppen, das können die fertigen Funktionen ABS, GSs, GS% (bzw THs, TH%) nicht.

Das kann man aber leicht mit den log.Schalter realisieren. **L1 a>x THR -95** fragt die Gasstellung ab und aktiviert L1. Mit **L2 OR L2 L1** wird aus L2 ein Flip-Flop, das gesetzt wird wenn L1 aktiv wird. L2 startet den Timer und stoppt ihn wenn L1 wieder rückgesetzt wird.

Mit dem Taster **SH** wird L1 wieder resetet, da **L2** mit **AND SH↑** inaktiv wird wenn **SH↓**

Ablauf: Gas auf min stellen, dann Gas>-95, L1 wird aktiv,

L2 wird mit sich selbst und L1 verodert und damit gesetzt und bleibt ON

mit AND SH hat der L2 Freigabe und damit wird später das Reset von L2 bewirkt.

L2 startet jetzt den Timer 2 und läuft durch , Gas kann jetzt beliebig sein.

mit SH wird L2 resetet und damit Timer 2 gestoppt

Merke: L2 ist damit ein Flip Flop das mit L1 gesetzt und mit SH resetet wird

The screenshot shows the 'ModelSetup' window with the 'Failsafe' tab selected. The 'Modell Name' is 'Timerstart'. There are two stopwatches: 'Stoppuhr1' set to 00:00 with function 'ABS', and 'Stoppuhr2' set to 00:00 with function 'CS2'. Below this is a table for logic gates:

	Funktion	V1	V2	AND
CS1	a>x	Thr	-95	---
CS2	OR	CS1	CS2	SH↑
CS3	---	---	0	---

Below the logic gates is a table for switches:

Schalter	Funktion	Parameter
CF1	SH↓	Reset

Und mit **SH↓** also SH „long“ (>1s) betätigen, kann man den Timer wieder zurück auf 00:00 stellen bzw. auf den Anfangswert z.B. 3min 03:00

Achtung:

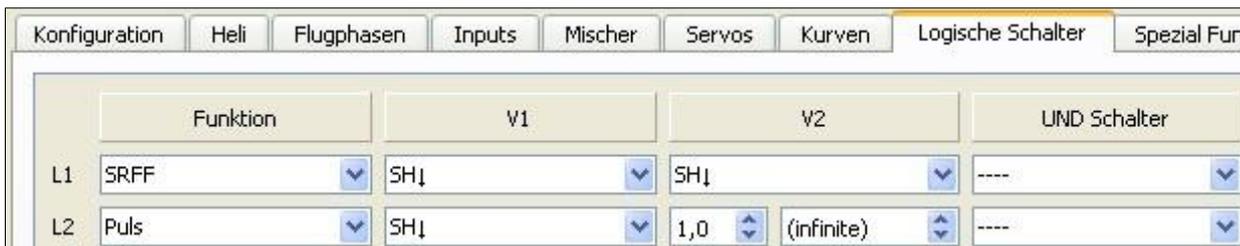
Short und Long gibt es ab openTx V2.00 nicht mehr, deshalb mit der Funktion Puls arbeiten!

Beispiel: Timer Start, Stop, Reset mit Taster SH

Die 3 Timer kann man nicht nur mit ABS, GSt, GSs, GS% laufen lassen, sondern mit sehr vielen anderen Bedingungen (siehe Companion) starten und stoppen. Mit dem Taster SH wollen wir einen Timer Starten, Stoppen und wieder auf den Startwert setzen. Wir verwenden ein SRFF (Set/Reset FlipFlop), die Puls-Funktion und 2 Log. Schalter L1, L2

L1: Der Taster SH↓ setzt und reset ein SR-FlipFlop, V1 setzt, V2 reset das SR-FlipFlop (Toggle-FF)
Nur wenn das SR-FlipFlop gesetzt ist, dann ist L1 auch aktiv.

L2: Wird der Taster SH↓ mindestens 1 sec gedrückt, dann wird ein kurzer Impuls erzeugt (Puls-Funktion) und L2 wird kurz aktiv.

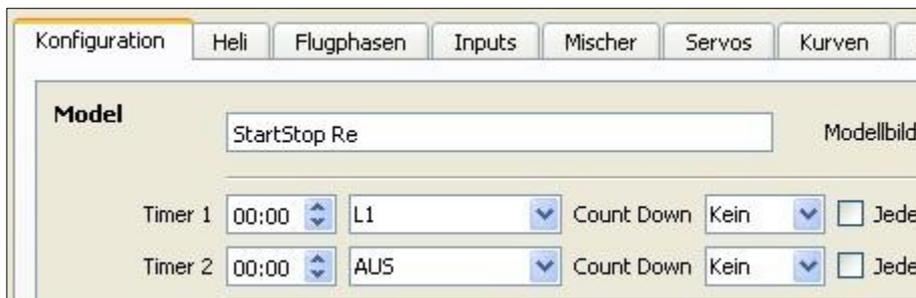


Mit L1 können wir nun den Timer 1 freigeben. Dazu bereiten wir den Timer 1 vor.

Da wir ihn auf 00:00 setzen läuft er vorwärts.

Steht ein anderer Werte als 00:00 drinnen läuft er automatisch rückwärts!

Wenn der Log Schalter L1 aktiv ist, ist der Timer 1 freigegeben und läuft.

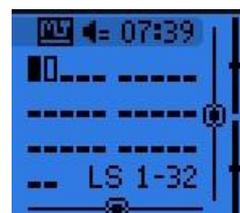


Mit L2 setzen wir den Timer 1 (Stoppuhr 1) zurück auf den Startwert, (hier eben auf 00:00)

(EIN nicht vergessen)

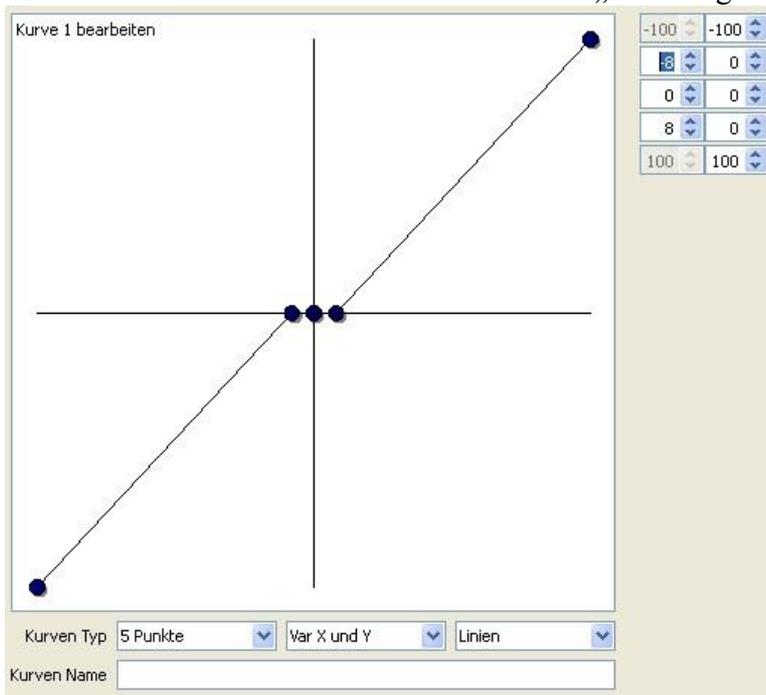


Timer 1 läuft wenn L1 aktiv ist.

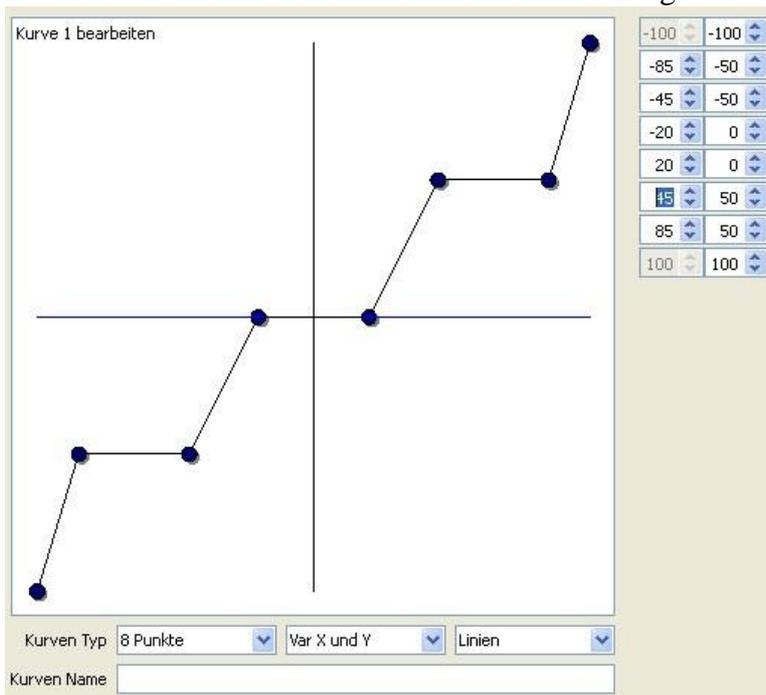


Beispiel: Potipositionen exakt einstellen.

Oft hat man das Problem Potipositionen exakt einstellen zu müssen. Also exakt auf Mitte „0“ nicht nur fast auf Mitte „0“ und $a \sim 0$ ist zu ungenau. Die Potiwerte kann man gut abfragen mit $a > 99$ oder $a < -99$ oder mit $a \sim 100$ usw. Für die Mittelstellungserkennung kann man zusätzlich den Mittelstellungspieps (Sender Grundeinstellungen) aktivieren, aber auch der ist zu ungenau. Gut sind auch Potis mit Rasterungen (Poti with detent) Oft haben sie nur eine Rasterung in der Mitte oder viele feine Rasterstufen. Am Besten geht es mit einer Kurve die um die Mitte „0“ eine Hysterese hat, also einen Bereich der einen konstanten Wert „0“ erzeugt. Hier im Bereich von ± 8



Mit dem gleichen Prinzip kann man auch ein „Stufen-Poti“ realisieren. 5 Stufen kann man per Hand noch gut einstellen. Die 2 Endlagen, die Mitte und 2 Werte dazwischen. Das kann man mit einer Ansage der Werte oder Pieps noch stark verbessern.



Beispiel: Einmalige Ansage eines Potiwertes nach der Veränderung

Nachdem ein (Poti) Wert verändert wurde, soll der neue Wert einmal angesagt werden.
Mit der Funktion Betrag Delta $\geq x$ kann eine Veränderung erkannt werden.

Aktion: Poti auf Veränderungen überwachen:

Log. Schalter L2 : Die Veränderung der Potistellung erkennen und merken

L2 mit Betrag **Delta $\geq x$** S1 5 Dauer 3s somit Veränderung von S1 >5 ist für 3s aktiv

Das kann man sich so klar machen:

In jedem Rechenzyklus des Senders werden alle Analogwerte neu eingelesen und gespeichert
Jetzt kann man intern die letzten Werte mit den aktuellen Werte vergleichen
und eine Differenz bilden, also den Unterschied, die Veränderung erkennen und merken

Die Delta-Funktion überwacht ständig einen Analogwert auf eine Änderungen.

Das Poti S1 muss sich um mind 5 % ändern damit L2 vom Delta x aktiv wird.

Wenn L2 vom Delta x aktiviert wird, dann ist der log. Schalter L2 für 3 sec aktiv

Reaktion auslösen in den Spezial Funktionen SFx:

Nach der Änderung den neuen Wert ansagen

Mit **SF1** (Not) **!L2** Play Value **S1**, damit erfolgt erst dann die Ansage erst,

wenn die Verstellung von S1 fertig ist und L2 nach 3s wieder inaktiv ist, also bei !L2

	Funktion	V1	V2	AND	Duration	Verzögerung
CS1	---	---	0	---	0,0	0,0
CS2	d >=x	S1	5	---	3,0	0,0
CS3	---	---	0	---	0,0	0,0

	Schalter	Funktion	Parameter	aktivieren
CF1	!CS2	Play Value	S1	No repeat
CF2	---	Safety CH01	0	<input type="checkbox"/> ON

Die Zeiten von 3s noch auf praktische Werte von 1-2s anpassen!

Beispiel: Schalter, Schaltkanal, einfache Zeitfunktion, Blinken

A: Die physikalischen Schalter können 2 Funktionen erfüllen.

1. Als **Mischer-Quelle** liefert ein Schalter immer automatisch (gilt auch für logische Schalter!)
 -100% +100% (2-Stufen) das entspricht: 1000us, 2000us
 bzw.
 -100% 0% +100% (3-Stufen) das entspricht: 1000us, 1500us, 2000us

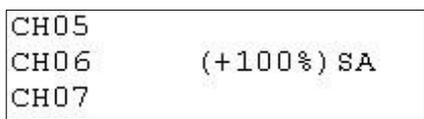
2. Als **Mischer-Schalter** aktiviert oder deaktiviert ein Schalter die komplette Mischerzeile

Wenn ein Schalter also nur einfach eine LED (via Servo-Schaltkanal) ein- oder ausschalten soll, dann reicht es ihn als Mischerquelle direkt zu verwenden.
 Gewichtung/Weight dann auf 100% lassen, das wars.

Man sollte natürlich schon wissen mit welchem (Servo)-Wert der Schaltkanal einschaltet und mit welchem Wert wieder ausschaltet.
 Meist liegt das ON bei >>1500us und Off bei <<1500us

Im Mischer den Schalter **SA** eingetragen

In Kanal 6 steht dann nur: **CH6 100% SA**



mehr ist nicht nötig für einen Schaltkanal

Wer es nicht glaubt kann es ja unter companion9x "simulieren"

Hier kann man noch eine langsame Bewegung eingeben z. B. für ein Fahrwerk oder um ein Ventil ganz langsam umsteuern oder zu verzögern.

Das kann man auch noch mit einer Kurve verfeinern

Als Mischerquelle / als Mischerschalter kann man alle, 2-Stufen, 3-Stufen, und Log Schalter verwenden.

3-Stufen Schalter kann man auch als 2-Stufen Schalter umprogrammieren.

Als **SA↑** und als nicht **!SA↑** und dann in den Mischerschaltern zu verwenden .

Oder man kann z.B. den 3 Stufen Schalter **SA↑** in den log. Schaltern verknüpfen und dann den log Schalter **L1** als Mischerquelle verwenden. Dann hat man mit **L1** 2 Stufen statt 3 Stufen



B: Die logischen Schalter Lx = PSx = CSx sind auch nur Schalter

und liefern als Mischerquelle -100% und +100%.

Also anstatt eines Physikalischen 2-Stufen Schalter kann man auch einen Lx einsetzen.

Aber ein Logischer Schalter muss erst irgendwie aktiv werden, damit er etwas bewirkt!
Er braucht Bedingungen um aktiv zu werden.

Da verwenden wir mal hier die Taktgenerator-Funktion **Takt (bzw TIM)**

Das ist ein Timer/Takt-Funktion mit einstellbarem ON/Off Taktverhältnis

V1= ON-Zeit und V2= Off-Zeit hier als zusammen $0,3s+0,7s=1,0s$

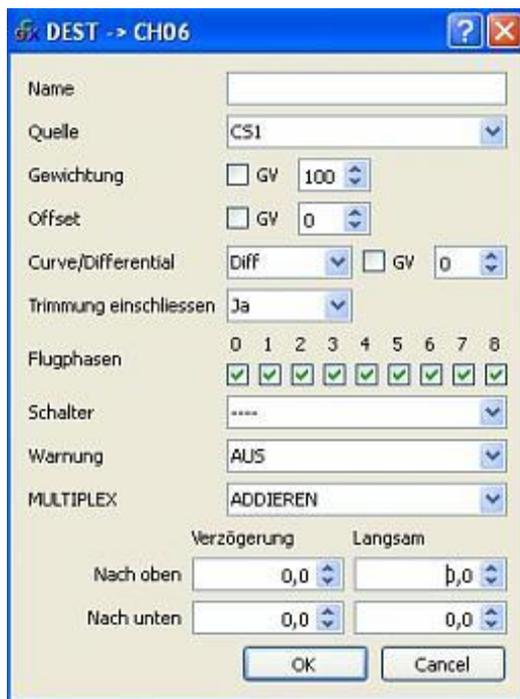
Damit wird der CS1 für 0,3s aktiv und für 0,7s inaktiv, d.h. der Kanal6 CH6 blinkt

Der Programmierbare Schalter CS1 wird so belegt

	Funktion	V1	V2	AND	Duration	Verzögerung
CS1	TIM	0,3	0,7	----	0,0	0,0
CS2	----	----	0	----	0,0	0,0
CS3	----	----	0	----	0,0	0,0

und im Mischer den **CS1** eingetragen

Im Kanal 6 steht dann auch nur: **CH6 100% CS1**



```
CH05
CH06      (+100%) CS1
CH07
```

und schon haben wir eine einstellbare Blinkschaltung oder einen Schaltkanal

C: Eine Kombination aus beiden Möglichkeiten Merke: **Takt = TIM** je nach Softwarestand

Das können wir jetzt noch steigern, indem wir die **Takt-** (bzw **TIM**)- Funktion einfach 2 mal mit unterschiedlichen Werten versorgen und aufrufen und damit einen echten Positionflasher programmieren.

Dazu verwenden wir 2 Logische Schalter CS1 und CS2 und tragen ein:

CS1 mit **TIM** 0,2 0,2 und ist mit **AND CS2** verknüpft
CS2 mit **TIM** 1,2 1,2

Das liest sich so:

Während CS2 EIN ist (für 1,2s) kann CS1 3 mal Ein-und Aus-Schalten $3 \cdot (0,2 + 0,2)$ und bleibt dann für weitere 1,2 sec aus.

	Funktion	V1	V2	AND	Duration	Verzögerung
CS1	TIM	0,2	0,2	CS2	0,0	0,0
CS2	TIM	1,2	1,2	---	0,0	0,0

Im Mischer Kanal CH6 gibt Schalter SA das Ganze als Schalter frei.

Und so haben wir hier mal beides zusammengestellt:

Im Kanal 6 einen einstellbaren Flasher der mit SA gesperrt und freigegeben wird.
 und im Kanal 7 einen einfachen Schaltkanal



```

CH05
CH06      (+100%) CS1 Schalter (SA↓)
CH07      (+100%) SA
CH08
    
```

Das kann man jetzt natürlich variabel einstellen.

Beispiel: Gaslimiter mit opentx Taranis wie bei einer Graupner MX16

Hier mal ein Gaslimiter so wie in einer Graupner MX16, einstellbar von -100% bis +100%
Das geht mit 2 Zeilen im Mischer, hier Kanal3, ganz einfach.

Poti **S1** begrenzt das max Gas des Gasknüppel, d.h. den Gaswert

Der Kniff ist die Vergleichs-Abfrage in **CS1 a>b Thr S1**

CS1 wird aktiv wenn Thr größer als S1 ist,
damit wird per Replace Zeile 2 im Mischer aktiv und der Wert kommt von S1,
Zeile 1 wird inaktiv.

S1 ist damit der max mögliche Throttelwert.

Ganz genau so funktioniert ein Gaslimiter

	Funktion	V1	V2	AND
CS1	a>b	Thr	S1	----
CS2	----	----	0	----

Variante 1:

Nicht vergessen: Gas Trimmung ganz nach unten!

CH01	(+100%)Ail
CH02	(+100%)Ele
CH03	(+100%)Thr
	R (+100%)S1 Schalter (CS1)
CH04	(+100%)Rud
CH05	

Variante mit 3 Zeilen Mischercode:

Eine kleine Erweiterung damit die Trimmwerte selbständig errechnet werden.

3. Zeile: Eigene Trimwerte Zeile mit 25% für Thr und S1 Gaslimiter

Damit treten keine Sprünge auf, auch wenn die Timmung nicht auf Null ist!

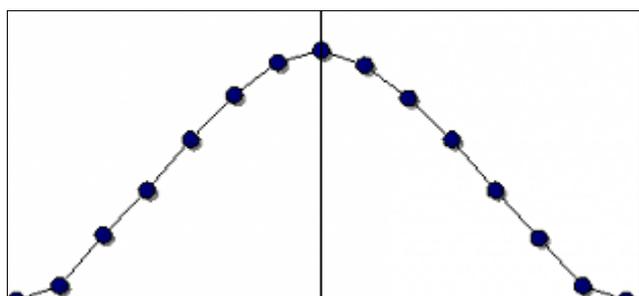
CH01	(+100%)Ail
CH02	(+100%)Ele
CH03	(+100%)Thr No Trim
	R (+100%)S1 Schalter (CS1) No Trim
	(+25%)TrmT No Trim
CH04	(+100%)Rud
CH05	

Beispiel: Pan -Tilt Kamera mit Limiter-Grenzkurve die nicht unter/überschritten wird

Eine Pan / Tilt Mechanik ist dicht am Rumpf über der Kabinenhaube angebracht.
Wenn man geradeaus schaut, darf die Kamera die Kabinenhaube nach unten nicht berühren.
Links und rechts am Rumpf vorbei kann sie frei nach unten schauen.

Das heisst, wenn man das Pan-Servo1 von -100% bis +100% schwenkt, darf das Tilt-Servo2 in Abhängigkeit des Pan-Servos1 einen bestimmten Bereiche nach unten nicht unterschreiten.
Das Tilt-Servo2 muss also **automatisch** eine **Limiter-Grenze** beachten.
Das ist eine Horizontallinien-Ausblendung, nur der Bereich über der Kurve soll möglich sein.

Pan / Tilt Kruven-Abhängigkeiten



Kamerahalterung
Mit 2 Servos
Pan= hin/ her
Tilt = auf / ab

Pan = Horizontalachse, x-Achse, Servo1 **Tilt** = Vertikalachse, y-Achse Servo2

Pan als X-Achse, das Servo1 läuft von -100 bis +100
Tilt als Y-Achse, das Servo2 läuft in Abhängigkeit der x-Achse-Kurve
Die Horizontal Grenzkurve darf nicht unter/überschritten werden.

Ablauf:

Ausgang Kanal 10 Pan → Eingang HilfsKanal 11 → Wert der Horizontalgrenz-Kurve erfassen → Wert in GVARs 1 schreiben → Vergleich als Limiter, der nicht unter/überschritten wird.

Hier zum Test ist folgendes eingestellt:

QuerruderKnüppel: als Horizontalbewegung, HöheruderKnüppel: als Vertialbewegung

Kanal 10: Pan-Servo1, ist die normale horizontale Bewegung von-100% bis +100%

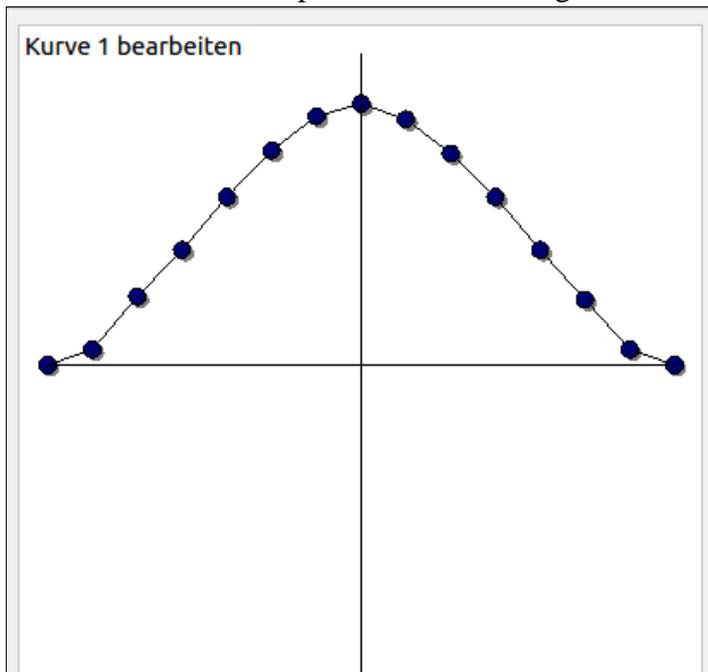
Kanal 11: Ein Hilfskanal, der in Abhängigkeit von Kanal10 eine Kurve abfährt.

Das ist die Grenzbewegung für den Limiter. Diese Grenzwerte des Kanal 11 gehen auf eine GVAR in den Spezialfunktionen

Kanal 12: Tilt-Servo2, der eigentliche Limiter für die vertikale Bewegung, das Replace begrenzt die max Bewegung, das ist der Limiter

Hinweis:

Die Kurve darf nur im positive Bereich eingestellt werden (da noch keine Funktion $|a|>b$).



Mischerzeilen: Kanal 10 Pan-Servo1 Kanal 12 Tilt-Servo2 Kanal 11 fährt die Kurve ab

CH10	Que Gewichtung(+100%) (Horizont)
CH11	CH10 Gewichtung(+100%) Kurve(1) (H-Grenze)
CH12	Höh Gewichtung(+100%) (Vertikal)
	R Höh Gewichtung(GV1) Schalter(L1)

GVAR1: Erhält den Werte der Kurve als Zuweisung aus CH11

	Schalter	Aktion	Parameter	Aktivieren
SF1	EIN	Adjust GV 1	Quelle CH11	<input checked="" type="checkbox"/> EIN
SF2	---	Safety CH1	0	<input type="checkbox"/> EIN

Limitier: Vergleich ob Kanal 12 Höhe größer als der aktuelle Kurvenpunkt von GV1

	Funktion	V1	V2
L1	a>b	Höh	GV1
L2	---	---	0

Da ist auch gleich die Sicherheit eingebaut, keine Kollision am Rumpf möglich!

Wenn man seitlich tief steht und jetzt einfach nur horizontal Pan-Servo1 bewegt, wird sich automatisch die Höhe Tilt-Servo2 entlang der Kurve mitbewegen.

Bitte mal testen, das Ding funktioniert.

Beispiel: PPMus - Werte berechnen für Flightcontroller und APM Anwendungen

Wie kann man Weight-Werte, also Verstärkungswerte, in den Mixern direkt berechnen, wenn wir eine bestimmte PPM-Impulsbreite in uns brauchen.
(ganz praktisch bei Copteranwendungen um div Flugmode einzustellen)

Wir gehen von den Normaleinstellungen aus:

Y= -100% bis +100% = 200% , X= 1000us bis 2000us = 1000us, Impuls-Mitte ist 1500us

Dann lautet die lineare Funktion :

$$F(x)=(dy/dx)*Y + b$$

$$F(x)= (1000/200)*Y + 1500$$

$$\text{bzw } Y= (1000/200)*Y + 1500$$

$$\text{und gekürzt } X=5*Y + 1500$$

Y ist der Weight-Wert den wir im Mixer einstellen müssen, damit wir einen gewünschten PPM-Impuls X als us erhalten.

Also Formel umstellen auf Y, damit haben wir: **$Y = (X-1500)/5$** (vereinfachte Formel)

Diese Formel vereinfacht doch einiges und geht viel schneller als probieren.

Beispiel:

Impuls X	Weight Y
1000us	-100
1100us	-80
1200us	-60
1300us	-40
1430us	-14
1500us	0
1560us	+12
1680us	+36
1700us	+40
1800us	+6
1900us	+80
2000us	+100

Anmerkung:

Ganz exkt macht die Taranis 1500us +/-512us also 1024 Stufen bei 200% (-100% bis +100%)

Damit ergibt sich die **exakte Formel $Y = (X-1500) / 5,12$**

Beispiel: APM- Mode mit 6 Stufen bzw mit 2 Schaltern einstellen

Ein Flugcontroller braucht für div Funktionen auf einem Kanal sechs verschieden Steuersignale als PPM-Werte mit vereinfachter Formel $Y = (X - 1500) / 5$ mit exakter Formel $Y = (X - 1500) / 5,12$

1165us = -67%	Absoluter Wert in us und %	-65%
+130 +26	Veränderung in us und %	
1295us = -41%		-40%
+130 +26		
1425us = -15%		-15%
+130 +26		
1555us = +11%		+11%
+130 +26		
1685us = +37%		+36%
+130 +26		
1815us = +63%		+62%

Die 6-Stufen kann man mit einem 6-Stufenschalter oder aber mit 2 Schaltern einstellen. Einem 2-Stufen- und einem 3-Stufenschalter z.B **SF** und **SA**

Das wird dann in den logischen Schaltern und Mischern verknüpft.

Logische Schalter

L1	AND	SA↑	SF↑
L2	AND	SA-	SF↑
L3	AND	SA↓	SF↑
L4	AND	SA↑	SF↓
L5	AND	SA-	SF↓
L6	AND	SA↓	SF↓
L7	---	----	0

Der Mischer erhält dann in der Gewichtung seine benötigten Werte

Mischereinstellungen

CH10	MAX	Gewichtung (-67%)	Schalter (L1)	
	R	MAX	Gewichtung (-41%)	Schalter (L2)
	R	MAX	Gewichtung (-15%)	Schalter (L3)
	R	MAX	Gewichtung (-11%)	Schalter (L4)
	R	MAX	Gewichtung (+37%)	Schalter (L5)
	R	MAX	Gewichtung (+63%)	Schalter (L6)

Das ist mal ein Beispiel wie der Ablauf sein kann, unabhängig davon welche Funktionen die einzelnen Stufen am Flugcontroller auslösen.

Bitte entsprechend anpassen!

Beispiel: 6-Stufenschalter umrechnen auf andere PPMus-Werte

Flighthcontroller oder LED-Controller brauchen ganz bestimmte PPM-Werte (in us) um bestimmte Funktionen auszulösen. Ein 6-Stufenschalter ist gut geeignet um diese Funktionen anzuwählen.

Der Fr-Sky 6-Stufenschalter gibt von sich folgende Werte aus (als S3 eingebaut):

-100%	-60%	-20%	+20%	+60%	+100%	Abstand 40%
1000us	1200us	1400us	1600us	1800us	2000us	Abstand 200us (nicht exakt!)
S31	S32	S33	S34	S35	S36	Neue Schalternamen

Beispiel für LED Funktionen:

1. (<1200µs) Beleuchtung AUS
2. (1200 µs) Landelicht (alle LED's leuchten weiß)
3. (1300 µs) Flugbeleuchtung 1 (die LED's leuchten in den per Setup eingestellten Farben)
6. (1600 µs) Flugbeleuchtung 4 („Dreier“, durchlaufend)
7. (1700 µs) Flugbeleuchtung 5 („füllen und leeren“)
10. (2000 µs) durchlaufender Farbkreis (alle LED's)

Da passen die PPM -Werte des Stufenschalters nicht zu dem was ein LED-Controller oder ein Flightcontroller braucht. Also muss man sich das anpassen oder umrechnen.

Das umrechnen geht vereinfacht so:

Ein Geber/ Poti/ Knüppel gibt -100% = 1000us bis +100% = 2000us aus.

Das sind 200% die auf 1000us aufgeteilt werden. Somit pro 100us 20% bzw 50us= 10%

Mit vereinfachte Formel: $Y = (X - 1500) / 5$

Damit kommt man auf diese vereinfachte Tabelle (exakte Formel unten!)

X	Y
1000us	= -100%
1100us	= -80%
1200us	= -60%
1300us	= -40%
1400us	= -20%
1500us	= 0%
1600us	= +20%
1700us	= +40%
1800us	= +60%
1900us	= +80%
2000us	= +100%

Genauer betrachtet:

Die Taranis erzeugt exakt folgende PPM-Werte:

Servo-Mitte 0% = 1500us -100% = 998us +100% = 2012us

Das sind 1024 Stufen. 1% sind dann genau 5,12us bzw 10% = 51,2us, 20% = 102,4us usw.

Exakte Formel um benötigte %-Werte aus PPMus auszurechnen $Y = (X - 1500) / 5,12$

Beispiel: Benötigt werden 1900us $Y = (1900 - 1500) / 5,12 = +78,125\%$ also +78%

Benötigt werden 1250us $Y = (1250 - 1500) / 5,12 = -48,828\%$ also -49%

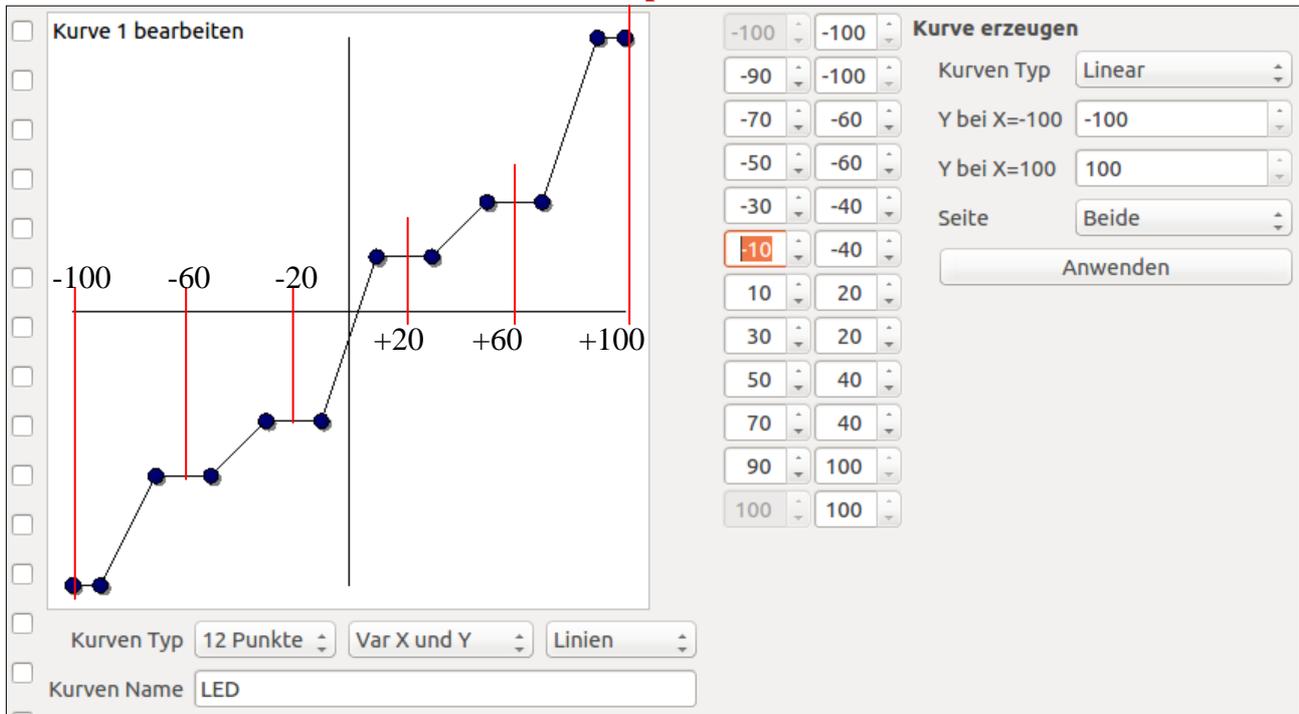
Jetzt können wir damit eine Stufenkurve erzeugen die aus den 6 Stufenwerten des Schalters die richtigen PPM-Werte (in us) für die LED erzeugen kannst.

X-Achse sind die 6 Stufen des Schalters -100% -60% -20% +20% +60% +100%

Y-Achse sind die neuen benötigten %-Werte -100% -60% -40% +20% +40% +100%

Im Programm dann in einer einzigen Mischerzeile mit dieser angepassten Kurve arbeiten.

6-Stufenschalter mit Kurve umrechnen und anpassen für LED-Controller



Auf der X-Achse der 6 Stufen-Schalter (rot) -100% -60% -20% +20% +60% +100%

Auf der Y-Achse die Werte für die LED Ansteuerung -100% -60% -40% +20% +40% +100%

Braucht man andere LED Funktionen, also andere Y- Werte, dann nur die Y-Achse anpassen.

Mischerzeile für Ch5:

CH5 **S3 Gewichtung (+100%) Kurve (1)**

Alternative ohne eine Kurve zu verwenden:

Statt mit einer Kurve und **einer** Mischerzeile kann man die Schalter S31 bis S36 als Mischerschalter verwenden, Mischer-Quelle ist Max, Gewichtung so anpassen dass die Werte für die LED passen und dann mit 6 Mischerzeilen als Replace einzeln aufrufen. → **Benötige %-Werte ausrechnen!**

Mischerzeilen für CH5:

CH5 **MAX Gewichtung (-100%) S31**
 R MAX Gewichtung (-60%) S32
 R MAX Gewichtung (-40%) S33
 R MAX Gewichtung (+20%) S34
 R MAX Gewichtung (+40%) S35
 R MAX Gewichtung (+100%) S36

Beispiel: PPM-Werte im Vergleich Taranis, Spektrum, Graupner, Futaba, Multiplex

Jeder Sender-Hersteller gibt seine Einstellbereiche für die Wege und damit für die Servos in Prozent an. **Diese Prozent-Werte sind nicht vergleichbar!**

Ein Servo wird mit PPM-Signalen angesteuert, das ist entscheidend für die max Wege die ein Servo drehen kann.

Je nach Hersteller und Getriebeuntersetzung, kann trotz gleichem PPM-Wert der Weg unterschiedlich sein. Der eine macht +/-60°, der andere +/-75° oder gar +/-90° Drehwinkel.

Vergleich der max PPM Impulsbereiche

	Min	Mitte	Max	%
Taranis	768us	1500us	2268us	bei 150%
Graupner	900us	1500us	2100us	bei 150%
Spektrum	900us	1500us	2100us	bei 150%
Futaba	950us	1520us	2085us	bei 135%
Multiplex	1050us	16000us	2150us	bei 150% ??

Es gibt auch Multiplex-Sender mit 1520us Mitte, Min 950us, Max 2050us = +/- 100% ??

Taranis		Spektrum/Graupner		Futaba	
Prozent	PPM	Prozent	PPM	Prozent	PPM
-150%	= 732us	-150%	= 900us	135%	= 950us
-125%	= 860us	-125%	= 1000us	120%	= 1016us
-100%	= 988us	-100%	= 1100us	-100%	= 1100us
-50%	= 1244us	- 50%	= 1300us	-50%	= 1310us
0%	= 1500us	0%	= 1500us	0%	= 1520us
+50%	= 1756us	+50%	= 1700us	+50%	= 1730us
+100%	= 2012us	+100%	= 1900us	+100%	= 1940us
+125%	= 2140us	+125%	= 2000us	+120%	= 2024us
+150%	= 2268us	+150%	= 2100us	+135%	= 2085us

Umrechnungen

Taranis	+/- 100%	= +/- 512us
Graupner	+/- 100%	= +/- 400us
Spektrum	+/- 100%	= +/- 400us
Futaba	+/- 100%	= +/- 420us
Multiplex	+/- 100%	= +/- 530us ??

Verhältnis: Spektrum / Taranis = 400us/512us = 0,78125 also 78,125%

d.h. wenn ich in der Spektrum100% eingestellt habe, dann muss ich bei der Taranis 78% einstellen, damit der gleiche PWM-Wert rauskommt.

Beispiel: Spektrum, Kreisel auf 57% eingestellt → 57%*0,78 = 45% bei der Taranis einstellen.
Entsprechend dann die anderen Umrechnungen für Mischergewichtungen, Offset, Kurven.

Beispiel: Telemetriewerte am Sender einstellen (A1, A2, A3, A4, RSSI, Vario)

Jeder Telemetriesensor hat eine feste ID und wird dadurch vom Empfänger erkannt.

Die S-Port-Sensoren werden einfach beliebig in Reihen hintereinander gesteckt.

Der Sender ordnet die Messwerte internen Variablen zu, die eingestellt und dargestellt werden.

Will man diese Telemetriewerte in Mischern weiterverarbeiten, dann können/müssen sie durch eine Vorverarbeitung in INPUTS angepasst werden z.B. auf +/-100%

Selbst wenn keine Sensoren am X8R-Empfänger angeschlossen sind, werden immer

2 Werte automatisch vom X8R an den Sender geschickt. **RSSI / Rx** und **A1**

RSSI bzw **Rx** ist die Empfänger- Feldstärke, die wird immer zurückgeliefert und kann für einen Voralarm z.B. auf **41dBm** und Alarm z.B. auf **38dBm** eingestellt werden.

A1 ist die Empfängerakkuspannung. Wird der Empfänger mit einem BEC versorgt hat er ca 5V fest.

A1 bis A4 sind universell einstellbare Messeingänge, die man in einem weiten Bereich frei einstellen kann (sofern A1 / A2, je nach Empfänger, auch herausgeführt sind).

Genau diesen Bereich kann man in den Progr Schaltern dann auch abfragen.

0,0 bis 13,2V dann hat man in den Progr. Schaltern auch diesen Bereich 0-13,2V zur Auswahl

8,0 bis 12,0V dann hat man in den Progr. Schalter auch nur diesen Bereich 8 bis 12V zur Auswahl

Zuerst in der Telemetrie den A1 bis A4 Bereich einstellen, erst dann kann man ihn abfragen!

Beim **X8R** ist der Analogeingang intern fest mit der Empfängerakkuspannung verbunden.

Er hat keinen zusätzlichen Analogeingang rausgeführt. Er liefert an **A1** die Akkuspannung

Der Empfänger kann mit 4-10V versorgt werden.

Meist hat man 4 bis 5 Zellen NiMH, NiCd also 4,8V-6V oder eine BEC mit 5V

Beim **D8R-II** und **D8R-XP** ist A1 und A2 herausausgeführt und kann als Messeingang verwendet werden. Der Analogeingang A1 kann direkt nur 3,3V verarbeiten.

Mit einem vorgeschalteten 4:1 Teiler sind 13,2V möglich mit 11:1 sogar 36,6V

Dazu gibt es kleine Aufsteckplatinen, oder man macht sich das selbst.

Beim **X6R** kann man per Jumper den A1 auf interne oder auf externe Spannungsmessung legen, auch dort sind direkt nur 3,3V möglich. Somit mit Widerstandsanpassung arbeiten.

Weiteres Sensorwerte:

Der **FAS 40A-Stromsensor** liefert den Stromwert Current und die Akkuspannung Vfas

Die Leistung, Power in Watt und der Verbrauch, Consumption in mAh wird im Sender errechnet und dargestellt. In der Telemetrie-seite, bei Daten, einstellen auf Spannung **FAS** Strom **FAS**, nicht **A1**, **A2**

Der **FVLSS-Sensor** liefert die Werte von Gesamt- und Einzelzellenspannung eines Akkus nach Cell und Cells an den Sender.

Das **Vario** liefert die Höhe, Altitude und die Steig- und Sinkrate Vertical Speed

Für die Variotöne kann man den min-max-Bereich und den Nullschieberbereich einstellen.

Übersicht: Telemetriewerte einstellen und darstellen aus Companion9x

The screenshot shows the 'Telemetrie' (Telemetry) configuration window in Companion9x. The window title is 'Modell 1 bearbeiten - MODEL01'. The interface is organized into several sections:

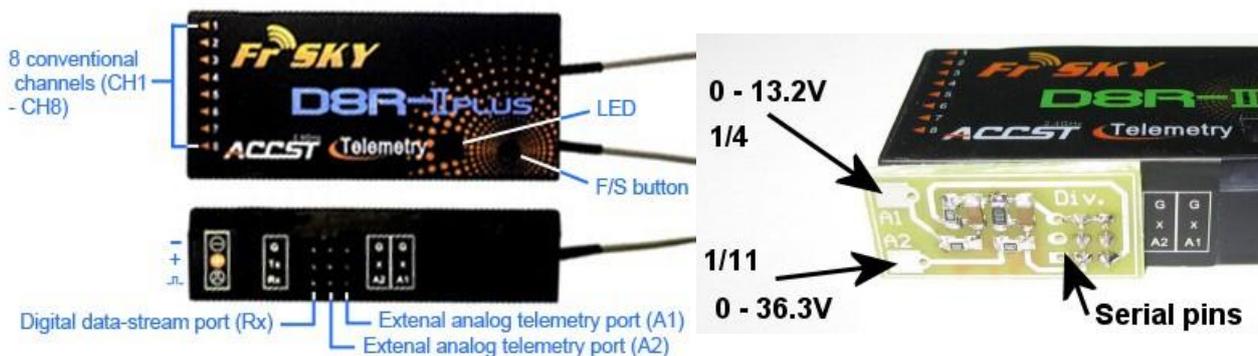
- Navigation:** Configuration, Heli, Flugphasen, Inputs, Mischer, Servos, Kurven, Logische Schalter, Spezial Funktionen, Telemetrie.
- Protocol Selection:** A dropdown menu is set to 'FrSky S.PORT'.
- Protocol A1:**
 - Einheit: Volt(V)
 - Bereich: 13,2
 - Voralarm: 4,50
 - Kritischer Alarm: 4,19
 - Offset: (empty)
- Protocol A2:**
 - Einheit: Amp (A)
 - Bereich: 50,0
 - Voralarm: 0,00
 - Kritischer Alarm: 25,10
 - Offset: 30,00
- Protocol A3:**
 - Einheit: Volt(V)
 - Bereich: 0,1
 - Voralarm: 0,00
 - Kritischer Alarm: 0,00
 - Offset: 0,01
- Protocol A4:**
 - Einheit: Amp (A)
 - Bereich: 1,6
 - Voralarm: 0,11
 - Kritischer Alarm: 0,26
 - Offset: 0,11
- R551:**
 - Voralarm: 39
 - Kritischer Alarm: 41
- RxBatt:**
 - Alarm 1: 4,61
 - Alarm 2: 4,40
- Serielles Protokoll:**
 - Protokoll: FrSky Sensor Hub
 - Einheiten: Metrisch
 - Spq Quelle: A2
 - Rotorblätter: 4
 - Strom Quelle: A3
- Höhenanzeige:**
 - Wario Quelle: 39
 - Wario Grenzen: 41
 - VSpeed: Sink Max -10
 - Sink Min Steig MinSteig Max: -0,5, 0,5, 10
 - Speichern der mAh: (checkbox)
- Verschiedenes:**
 - FAS Offset: 0,0A
 - mAh zählen: (checkbox)
 - Speichern der mAh: (checkbox)
- Telemetrie Bild 1, 2, 3:**
 - Telemetrie Anzeige als: Werte
 - Telemetrie Bild 1: Verbrauch
 - Telemetrie Bild 2: Geschw
 - Telemetrie Bild 3: Strom

Übersicht: ID-Werte für die Frsky Smart-Port Sensoren

Variometer	FVAS-02	: 01	(hatte am Anfang eine falsche ID mit 04)
Spannungsmesser	FLVSS	: 02	
Stromsensor 40A	FCS-40	: 03	
GPS-Sensor	GPS	: 04	
Drehzahlsensor	RPM	: 05	
Serielle Schnittstelle	SP2UART	Host	: 06
Serielle Schnittstelle	SP2UART	Remote	: 07

Dann gibt es immer mehr Smart-Port Sensoren von anderen Herstellern z.B. openTXsensor. Auch die verwenden für ihre Sensoren das Smart-Port Protokoll mit entsprechenden ID-Nummern.

Der Variometer-Sensor kann auch als Interface zum bisherigen Frsky Hub verwendet werden.



Beispiel: Vario einstellen und Höhenansagen aufrufen

Das Frsky Vario wird direkt an den SPORT angeschlossen (nicht verwechseln mit SBus)

Es liefert via Telemetrie die Höhenmesswerte.

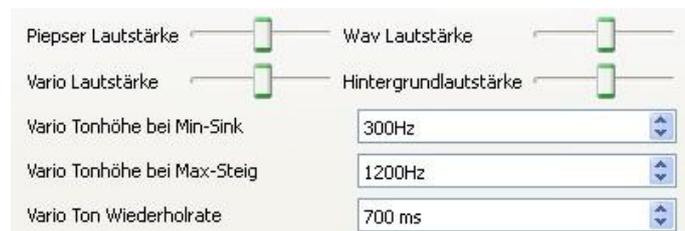
Die **Anzeige** von Höhe und Vertical Speed muss man am Telemetrie Bildschirm aktivieren.

Die **Höhenänderungen** werden im Sender erzeugt und können als **Variotöne** hörbar gemacht werden.



Die Variotöne kann man im Sender-Grundmenü und in Companion einstellen

Um die Höhenansagen und die Variotöne zu aktivieren müssen in den Spezialfunktionen via Schalter Variotöne oder Ansage der Höhe aufgerufen werden



Beispiel:

SA↑ keine Töne und keine Ansagen

SA ▬ nur Variotöne ausgeben

SA↓ Ansage der Höhe alle 5 sec

SF↓ Start Log = Start der Telemetrie-Datenaufzeichnung auf SD-Karte alle 0,1s

	Schalter	Funktion	Parameter	aktivieren
CF1	SA-	Vario		
CF2	SA↓	Play Value	Alt	5s
CF3	SF↓	Start Log	0,1	<input checked="" type="checkbox"/> ON

In der Telemetrie muss der Variotonbereich eingestellt werden

Bereich z.B. von -5 bis +5 m/s das ist das Min und Max der Töne

und z.B. -0,4 bis +0,0 das ist der Bereich wo keine Variotöne ausgegeben werden

(**Nullschieberbereich** festlegen)



Den Bereich um 0,0 ausblenden oder unterhalb bleiben, denn dort sind Rauschsignale die stören!

Beispiel: Servotester mit einstellbaren Wegen und Zeiten in 2 Varianten

Bis opentx 2940 bzw companion V1.52 gabe es auch Templates für die Taranis.
Dort gab es auch einen Servotester als fertige Funktion.
In den Sendern Th9, 9XR und 9XRPro sind diese Template noch vorhanden.

Ein Servotester ist eine praktische Sache wenn er in den Servowegen und in den Servolaufgeschwindigkeiten einstellbar. Deshalb hier nochmal ein Servotester.

Wir verwenden hier die Kanäle **CH15** und **CH16**
Dort läuft der Servotester immer nebenher mit und stört nicht!

Die Wege sind frei mit der Gewichtung einstellbar
Die Laufzeiten aus Verzögerung und Langsam-Funktion ist im Kanal **CH16** einstellbar
Kanal **CH15** ist ein 2. Servotestkanal der ohne Verzögerung sofort umschaltet, Wege mit Gewichtung

Gesteuert wird der Servotester automatisch über den Log. Schalter **L1**

L1 ist hier die Mischerquelle. Ein Log. Schalter als Quelle liefert von sich aus -100% oder +100%
Er schaltet immer um EIN, AUS, EIN, ... in Abhängigkeit vom Wert des **CH16** ob <0 oder >0

L1 a<x CH16 0 ----

CH16 läuft verzögert und langsam hin und her, je nachdem ob **L1** EIN oder AUS ist

```
CH15    L1 Gewichtung (+75%)
CH16    L1 Gewichtung (+100%) Verzögerung (u3: d3) Langsam/ u4: d4)
```

Anwendung: Wo das Servo anschließen?

Wenn wir den Servotester mal schnell brauchen können wir CH15 oder CH16 in einen freien Kanal des Empfängers kopieren und dort Servos anschließen. (hier kann man auch die Wege einstellen)

```
CH6
CH7    CH15 Gewichtung (+100%)
CH8    CH16 Gewichtung (+100%)
CH9
```

Erweiterung: Servotester freigeben oder sperren

Einfach mit einem Schalter den Servotest im entsprechenden Kanal freigeben und sperren

```
CH7    CH15 Gewichtung (+100%) Schalter (SF↓)
CH8    CH16 Gewichtung (+100%) Schalter (SF↓)
```

Servotester Variante 2

Hier verwenden wir einfach den Taktgenerator: Takt 2s EIN und 2s AUS usw.
 Der schaltet damit einen log Schalter EIN und AUS
 Der Rest ist gleich.

	Funktion	V1	V2
L1	Takt	2,0	2,0
L2	---	----	0

CH6	
CH7	L1 Gewichtung(+100%)
CH8	L1 Gewichtung(+100%)Langsam/u2:d2)
CH9	

Mit L1 als Mischerquelle liefert EIN = +100% und AUS = -100%

Die min / max Wege können wir mit der Gewichtung einstellen

CH7 läuft unverzögert

CH8 läuft langsam in 2s auf und 2s ab

Taktzeiten müssen größer als Langsamzeiten sein, sonst kann der Endwert nicht erreicht werden.

Beispiel: Langsame Servo-Bewegungen mit Slow up Slow down im Mischer

Um Slow Up und Slow Down in einem Mischer mit **mehreren** Zeilen verwenden zu können, muss sich die **Mischer-Quelle** ändern, das ist halt mal so!

Es nutzt nichts wenn man nur in der Mischerquelle einen Schalter verwendet und den umschaltet von -100% auf +100%. Es ist immer noch der gleiche Schalter!

Da muss man etwas tricky programmieren, mit Replace geht es auch nicht. Es muss in der Mischerquelle, ein anderer Geber, Schalter, logischer Schalter usw auftauchen.

Das geht z.B. mit 2 programmierbaren Schaltern CS2 und CS3 die mit SA umgeschaltet werden. Im Mischer wird CS2 und CS3 mit Mischer Weight angepasst. mit dem gleichen Schalter SA wird die Mischerzeile aktiviert, nicht mit Replace!

Beispiel: Querruder als Flap in 3 Stufen

Querruder als Flap mit SA in 3 Stellungen 0% -25% -40% mit Slow up Slow down mit CS2 und CS3 als Mischer-Quellenumschaltung

Schalter SA in den Log. Schalter aktiviert CS2 und CS3 (CS1 hier nicht verwendet)

Schalter SA aktiviert auch die entsprechenden Mischerzeilen

Kein Replace, da die Slow up Slow down Zeiten ablaufen müssen und per Schalter umgeschaltet wird.

Das ist keine Flugphasenumschaltung, alles passiert hier noch in der gleichen Flugphase!

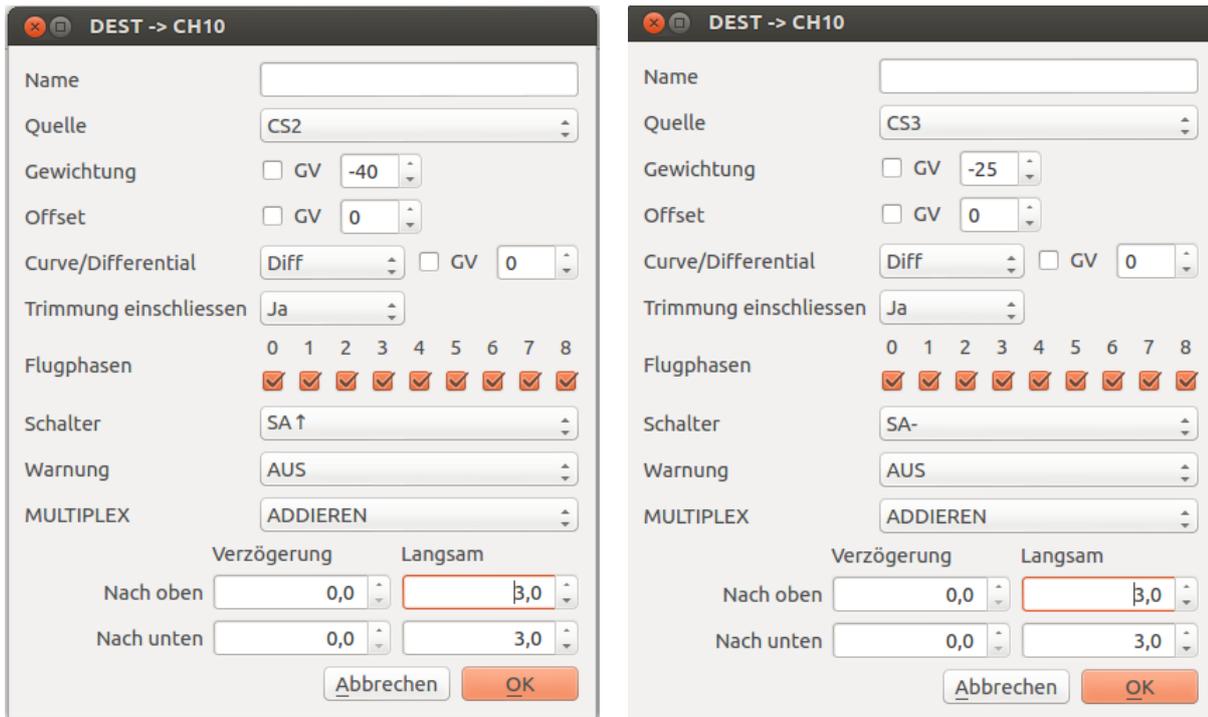
Anmerkung:

Was im ersten Augenblick etwas umständlich aussieht hat seine großen Vorteile bei der Signalvorverarbeitung und der Flexibilität über die frei programmierbare Logik der programmierbaren Schalter und deren weiter Verknüpfungen.

	Funktion	V1	V2	AND	Duration	Verzögerung
CS1	a~x	SA	100	---	0,0	0,0
CS2	a~x	SA	-100	---	0,0	0,0
CS3	a~x	SA	0	---	0,0	0,0
CS4	---	---	0	---	0,0	0,0

```

CH09
CH10    (+100%)Ail
        (-40%)CS2 Schalter(SA+)Langsam/u3:d3)
        (-25%)CS3 Schalter(SA-)Langsam/u3:d3)
CH11
    
```



Die beste Lösung ist aber folgende Möglichkeit:

Mischerverrechnungen und die Funktionen Langsam und Verzögerung trennen.

Das geht so:

1. Mehrere Mischerzeilen in einem freien, oberen Mischer zusammenfassen, z.B. in Ch20.

Mit allem was man dort braucht, Addiere, Replace, Schalter,

Die eigentlichen Mischerverrechnungen und Mischerumschaltungen erfolgen hier.

Dort aber kein Langsam oder Verzögern verwenden!

2. Dann nur diesen einen Kanal (z.B. in Ch20) in dem tatsächlichen benötigten Mischer (z.B Ch5) als **einzig**e Mischerquelle verwenden.

Da jetzt hier nur eine einzelne Mischerzeile steht, kann man hier die Funktionen Langsam und Verzögerung ohne Einschränkung verwendet werden.

Dadurch treten bei direkten Umschaltungen per Schalter keine Sprünge auf!

Beispiel: Flightmode mit Fade-In Fade-Out langsam Ein-und Ausblenden

Mit den Flightmodes kann man auch ganz geschickt langsame Übergänge realisieren.

Mit einem Schalter will ich z.B. die Gasstellung von einem beliebigen aktuellen Wert auf einen fixen Wert umschalten und dann auch wieder auf den Ausgangswert zurückschalten.

Der Vorgang soll nicht abrupt sondern einstellbar langsam in beide Richtungen erfolgen

Also:

Schalter On: Von der aktuellen Gasstellung langsam auf z.B. +50% fix umschalten

Schalter Off: Von den fix +50% wieder langsam zurückauf die aktuelle Gasstellung und das in 1-2sec

Dazu kann man in den Flugphasen die Fade-In und Fade-Out -Übergangs-Zeiten ganz geschickt anwenden.

Schalter SF aktiviert/deaktiviert den Flugmode, dabei werden dann die Fade-In Fade-Out Überblendungen aktiviert.

Flight Mode 0 (Default)	FM 1 (Gas fix)	FM 2	FM 3	FM 4	FM 5	FM 6	FM 7	FM 8
Flight Mode Name	Gas fix						Fade In	1,5
Schalter	SF1						Fade Out	2,0

Und im Mischer dann den Flightmode aufrufen

Konfiguration	Hubschrauber Setup	Flight Modes	Sticks	Mischer
<input type="checkbox"/> Show channels names in mixes				
CH01	(+100%) Thr			
	(+50%) MAX Flight mode (Gas fix)			
CH02	(+100%) Ail			
CH03	(+100%) Ele			
CH04	(+100%) Rud			

Beispiel: GVAR in den einzelnen Flugphasen anwenden und variabel verändern

Es gibt 81 globale Variablen, GVARs. Diese können in Flugphasen/Flugmode ganz geschickt verwendet werden und sparen uns in den Mischerzeilen viel Programmieraufwand. Allerdings versteht man da nicht gleich was, wo, wie abläuft. Deshalb ein Spielbeispiel.

Jede Flugphase hat 9 GVARs, GV1-GV9, die beliebige Werte haben können, Festwerte oder auch veränderliche Werte.

Die Inhalt der GVARs kann man sich in der Simulation anschauen. **FM2** ist aktiv da fett

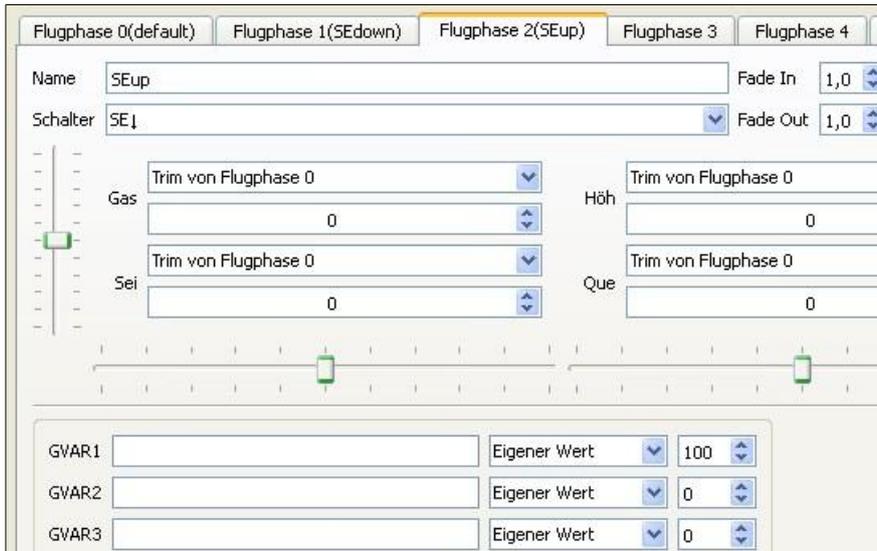
	FM0	FM1	FM2	FM3	FM4	FM5	FM6	FM7	FM8
GV1	0	-50	100	0	0	10	0	0	0
GV2	0	0	0	0	0	0	0	0	0
GV3	0	0	0	0	0	0	0	0	0
GV4	0	0	0	0	0	0	0	0	0
GV5	0	0	0	0	0	0	0	0	0
GV6	0	0	0	0	0	0	0	0	0
GV7	0	0	0	0	0	0	0	0	0
GV8	0	0	0	0	0	0	0	0	0
GV9	0	0	0	0	0	0	0	0	0

Die **GV1** hat hier in **FM0** den Wert **0**, in **FM1** den Wert **-50**, in **FM2** den Wert **+100**

FP0 ist immer aktiv wenn sonst keine andere FP aktiviert ist.

FM1 aktivieren wir mit dem Schalter **SE↑** (am Sender, **SEdown**, da er von uns weg, nach unten zeigt) und vergeben einen Namen. Mit **Fade In** und **Fade Out** erhalten wir einen sanften Zeitablauf für den Übergang. Die **GVAR1** erhält einen festen Wert von **-50**

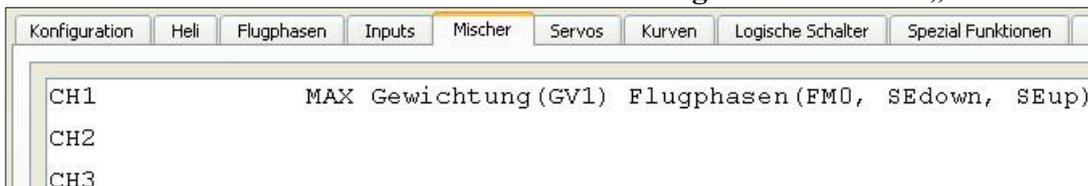
FP2 wird aktiviert durch mit SE↓ (am Sender. SEup zeigt zu uns her, nach oben)
 GVAR1 erhält einen Festwert von +100%. Ansonsten wie in FM1, Fade In, Fade Out



Aufruf im Mischermenü:

Im Mischermenü sieht das dann ganz harmlos aus und man erkennt im ersten Augenblick gar nicht was da eigentlich passiert.

Wir haben hier 3 Zustände des CH1 in einer einzigen Mischerzeile „versteckt“!



MAX ist der Festwert, liefert +100%, **die Gewichtung kommt von GV1**

Die Mischerzeile ist in 3 Flugphasen aktiv, FM0, FM1, FM2 und verhält sich unterschiedlich.

Wenn wir SE betätigen schalten wir damit die Flugphasen um und erhalten je nach Schalterstellung in der Mischer-Gewichtung die unterschiedlichen Werte der GV1 aus den jeweiligen Flugphasen übertragen.

CH1 bewegt sich von -50% 0% +100%, mit sanften Übergängen und ohne ruckeln!

Erweiterung:

Durch ändern des Inhalt von GVAR1 erhält man sofort ein anderes Verhalten von CH1

GVARs müssen keine Festwerte sein, sondern können variabel sein.

Beispiel:

Wir wollen mit S1 arbeiten und indirekt der GVAR1 zuweisen

S1 liefert von sich aus -100% bis +100%, das ist uns zu viel und der falsche Bereich

Deshalb: Signal-Vorverarbeitung in den Inputs verwenden

S1 wird in den Inputs vorverarbeitet und im Bereich angepasst auf -100 bis 0%

Berechnung wie in bei den Mischern $[(\text{Quelle} * \text{Gewichtung}) + \text{Offset}]$

```

Input05
[I6]S1      Gewichtung (50%)  Offset (-50%)  Quelle (S1)
Input07
    
```

Der Wert von Input **[I6]S1** in den Spezialfunktionen der GVAR1 zugewiesen.

	Schalter	Aktion	Parameter	Aktivieren
SF1	EIN	Adjust GV 1	Quelle [I6]S1	<input checked="" type="checkbox"/> EIN
SF2	----	Safety CH1	0	<input type="checkbox"/> EIN

Damit kann man alle 3 Stellungen von CH1 aktiv verändern!

Und die GVAR1 enthält dann ganz unterschiedliche angepasste Werte

Taranis Simulator	Ausgaben	Gvars			
		FM0	FM1	FM2	FM3
GV1		-71	-86	-25	0
GV2		0	0	0	0
GV3		0	0	0	0

Aber auch Vorsicht:

Bei jeder Aktivierung einer Flugphase wird auch die aktuelle Stellung des S1 neu eingelesen

Und in die jeweilige GVAR der aktiven Flugphase eingetragen!!

Bitte mal selber programmieren.

Verblüffend!

Beispiel: Kanalverriegelung, Veränderung des Kanalwert sperren

Will man einen Einstellwert vor Veränderungen schützen, so kann man ihn „verriegeln“.

Am Beispiel verwenden wir das S1 Poti mit dem wir den Kanal 8 einstellen, soweit so gut.

Mit der 2. Zeile rufen wir Kanal 8 selbst, mit seinem aktuellen Wert auf,
und per Schalter SA ersetzen wir (Replace) die 1. Zeile.

Damit ist der Wert verriegelt, egal wie wir jetzt S1 verdrehen.

```
CH8          S1 Gewichtung (+100%)
              R CH8 Gewichtung (+100%) Schalter (SA↓)
CH9
```

Das gleiche Ergebnis kann man auch mit GVARs anwenden.

In den Spezialfunktionen die Globale Variable per Schalter SA aktivieren und per S1 einstellen

	Schalter	Aktion	Parameter		Aktivieren
SF1	SA↓	Adjust GV 1	Quelle	S1	<input checked="" type="checkbox"/> EIN

Im Mischer dann als Quelle Max= +100% als Festwert und Gewichtung als globale Variable

```
CH9
CH10          MAX Gewichtung (GV1)
```

Eleganter geht es mit Trimmast, die auch frei verwendbar sind.

SF1	GasTrim auf	Adjust GV 1	Increment	+1	<input checked="" type="checkbox"/> EIN
SF2	GasTrim ab	Adjust GV 1	Increment	-1	<input checked="" type="checkbox"/> EIN

```
CH9
CH10          MAX Gewichtung (GV1)
```

Im Gegensatz dazu:

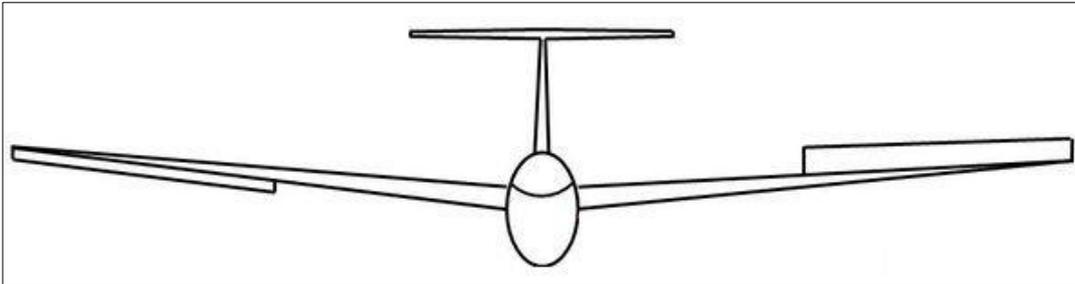
Mit der Override-Funktion in den Spezialfunktionen kann man einen Kanal Per Schalter nur mit einem Festwert überschreiben.

SF3	SA↓	Override CH8	67	<input checked="" type="checkbox"/> EIN
SF4	SA↓	Override CH10	75	<input checked="" type="checkbox"/> EIN

Beispiel: Variable Querruder-Differenzierung, im Flug mit GVAR 0-50%

Beispiel: Variable Querruderwege, im Flug mit GVAR einstellbar 50-100%

Segler von vorne:



Das Querruder nach unten macht einen kleineren Ausschlag als das Querruder nach oben. Dieser Anteil heißt Differenzierung.

Nur wenn das Querruder nach unten geht wird ein Wert addiert, nach oben nicht.

Diff: $-100\% + 40\% = -60\%$

40% Differenzierung, das Ruder geht nur noch bis -60% aus

100% Differenzierung, das Ruder geht gar nicht mehr nach unten 0%

0% Differenzierung, das Ruder schlägt voll aus -100%

Das würde man ganz normal so programmieren.

Normale Einstellungen für 2 Querruder, Differenzierung als Festwert mit 40%

Konfiguration	Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter
CH1			[I1] Gas	Gewichtung (+100%)			
CH2			[I2] Que	Gewichtung (+100%) Diff (40%)			
CH3			[I3] H	Gewichtung (+100%)			
CH4			[I4] Sei	Gewichtung (+100%)			
CH5			[I2] Que	Gewichtung (-100%) Diff (40%)			

In den Inputs die Knüppel für Quer und Höhe mit ca 35% Expo

Konfiguration	Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Spezia
[I1] Gas			Gewichtung (100%)	Quelle (Gas)				
[I2] Que			Gewichtung (100%)	Quelle (Que)	Expo (35%)			
[I3] H			Gewichtung (100%)	Quelle (Höh)				
[I4] Sei			Gewichtung (100%)	Quelle (Sei)	Expo (35%)			
Input05								
Input06								

Differenzierung variabel einstellbar von 0% bis 50%

Mit Poti **S1** im **CH10** als Vorverarbeitung

Spanne ist 0 bis 50% = 50%, Gewichtung ist $50\%/200\% = 0,25 = 25\%$

Offset = Mitte des neuen Bereichs 0 bis 50% = 25%

Somit macht **S1** im **Ch10** nur **0 bis 50%**

CH9	
CH10	S1 Gewichtung (+25%) Offset (25%)
CH11	

Mit diesem **CH10**-Wert versorge ich nun in den Spezialfunktionen die **GV1**

Konfiguration					Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Spezial Funktionen
	Schalter	Aktion	Parameter	Aktivieren								
SF1	EIN	Adjust GV 1	Quelle CH10	<input checked="" type="checkbox"/> EIN								
SF2	----	Safety CH1	0	<input type="checkbox"/> EIN								
SF3	----	Safety CH1	0	<input type="checkbox"/> EIN								

Und dieser **GV1** geht jetzt statt dem Festwert in die Differenzierung rein.

Damit habe ich eine im Flug frei einstellbare Differenzierung von 0 bis 50%

Konfiguration					Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter
CH1	[I1] Gas	Gewichtung (+100%)									
CH2	[I2] Que	Gewichtung (+100%)	Diff (GV1)								
CH3	[I3] H	Gewichtung (+100%)									
CH4	[I4] Sei	Gewichtung (+100%)									
CH5	[I2] Que	Gewichtung (-100%)	Diff (GV1)								
CH6											
CH7											
CH8											
CH9											
CH10	S1	Gewichtung (+25%)	Offset (25%)								

Erweiterung: Auch die Querruderwege im Flug variabel einstellen

Mit dem gleichen Grundprinzip kann auch im Flug meinen Querruderweg einstellbar machen.

Aber Vorsicht: Was soll den mindestens noch Querruderausschlag vorhanden sein?

Hier am Beispiel sagen wir mal 50% sollen min noch da sein und man soll variabel von 50% auf 100% erhöhen können.

Mit Poti S2 und Hilfskanal CH11 als Vorverrechnung

50% bis 100% ist eine Spanne von 50% also Gewichtung $50/200=0,25=25\%$

Offset = Mitte des neuen Bereichs = 50% bis 100% = 75%

CH10	S1 Gewichtung (+25%) Offset (25%)
CH11	S2 Gewichtung (+25%) Offset (75%)

Mit CH11 gehe ich jetzt in die Spezialfunktionen und versorge GV2

Schalter	Aktion	Parameter	Aktivieren
SF1 EIN	Adjust GV 1	Quelle CH10	<input checked="" type="checkbox"/> EIN
SF2 EIN	Adjust GV 2	Quelle CH11	<input checked="" type="checkbox"/> EIN
SF3 ----	Safety CH1	0	<input type="checkbox"/> EIN

Mit **GV2** versorge ich jetzt die Gewichtung, mit **GV1** versorge ich die Differenzierung

Kleine Steigerung: Ich will mit einem Festwert 100% und 40% Diff fliegen können

(wie oben) und dann aber umschalten zum Testen der Funktionen mit **S1** und **S2**

Dazu brauche ich ein Umschaltung der Mischerzeilen an der richtigen Stelle.

R= Replace = Ersetzt im Kanal alle Mischerzeilen die darüber steht

SA↑ in Grundstellung als Normalbetrieb mit Festwerten, bei **!SA↑** mit var. Differenzierung

Kanal	Mischerzeile	Parameter
CH1	[I1] Gas	Gewichtung (+100%)
CH2	[I2] Que	Gewichtung (GV2) Diff (GV1)
	R [I2] Que	Gewichtung (+100%) Schalter (SA↑) Diff (40%)
CH3	[I3] H h	Gewichtung (+100%)
CH4	[I4] Sei	Gewichtung (+100%)
CH5	[I2] Que	Gewichtung (-GV2) Diff (GV1)
	R [I2] Que	Gewichtung (-100%) Schalter (SA↑) Diff (40%)
CH6		
CH7		
CH8		
CH9		
CH10	S1	Gewichtung (+25%) Offset (25%)
CH11	S2	Gewichtung (+25%) Offset (75%)

Damit kann ich jetzt Variabel einstellen:

S1 die Differenzierung 0% bis 50% einstellen

S2 Querruder von 50% bis 100% einstellen

Bitte mal simulieren!

Inputs = Knüppel = Sticks

Natürlich habe ich 35% Expo auf den Stick für Höhe und Quer.

Da könnte man jetzt noch per Schalter die Wege Dualrate / Trirate in 2 / 3 Stufen umschalten.

Konfiguration	Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Spezia
[I1]	Gas	Gewichtung (100%)	Quelle (Gas)					
[I2]	Que	Gewichtung (100%)	Quelle (Que)	Expo (35%)				
[I3]	H h	Gewichtung (100%)	Quelle (Höh)					
[I4]	Sei	Gewichtung (100%)	Quelle (Sei)	Expo (35%)				
	Input05							
	Input06							

Variante: Variable Querruderwege einstellbar in den Inputs (gefällt mir am Besten)

Die einstellbaren Querruderwege (nicht die variable Differenzierung) mache ich nicht in den Kanalmischern sondern in den Inputs und verrechne sie dort als "Trirate" statt Dualrate.

Normaler Festwert: 100% Reduzierter Festwert: 60% Variabler Wert: 50-100%

GV2 via CH11 via S2 Umschaltung via 3-Stufen Schalter SC

Inputs als „Trirate“ Umschaltbar 100%, 60% und variabel

Konfiguration	Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Spezial Funktionen	Telemet
[I1]	Gas	Gewichtung (100%)	Quelle (Gas)						
[I2]	Que	Gewichtung (100%)	Quelle (Que)	Expo (35%)	Schalter (SC↑)				
		Gewichtung (60%)	Quelle (Que)	Expo (35%)	Schalter (SC-)				
		Gewichtung (GV2)	Quelle (Que)	Expo (35%)	Schalter (SC↓)				
[I3]	H h	Gewichtung (100%)	Quelle (Höh)						
[I4]	Sei	Gewichtung (100%)	Quelle (Sei)	Expo (35%)					
	Input05								
	Input06								

CH11 via S2 Einstellbereich 50% bis 100%

Spanne 50 bis 100% = 50% Gewichtung 50/200 = 25%

Offset Mitte des neuen Bereichs von 50% bis 100% = 75%

CH10	S1	Gewichtung (+25%)	Offset (25%)
CH11	S2	Gewichtung (+25%)	Offset (75%)

GV2 wird von CH11 versorgt

Konfiguration					Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Spezial Funktionen
Schalter	Aktion	Parameter		Aktivieren								
SF1	EIN	Adjust GV 1	Quelle	CH10	<input checked="" type="checkbox"/>	EIN						
SF2	EIN	Adjust GV 2	Quelle	CH11	<input checked="" type="checkbox"/>	EIN						
SF3	----	Safety CH1	0		<input type="checkbox"/>	EIN						

Das sieht jetzt ein klein wenig anders aus.
 Da in den Inputs der Querruderwege umgeschaltet werden
 und im Mischer die Differenzierung stattfindet.

Mischer Normalbetrieb mit Differenzierung als Festwert oder Variabel 0-50%

Konfiguration										Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Spezial Funktionen	Telemetrie	
CH1	[I1] Gas Gewichtung (+100%)																		
CH2	[I2] Que Gewichtung (+100%) Diff (GV1)																		
	R	[I2] Que Gewichtung (+100%) Schalter (SA↑) Diff (40%)																	
CH3	[I3] H h Gewichtung (+100%)																		
CH4	[I4] Sei Gewichtung (+100%)																		
CH5	[I2] Que Gewichtung (-100%) Diff (GV1)																		
	R	[I2] Que Gewichtung (-100%) Schalter (SA↑) Diff (40%)																	
CH6																			
CH7																			
CH8																			
CH9																			
CH10	S1 Gewichtung (+25%) Offset (25%)																		
CH11	S2 Gewichtung (+25%) Offset (75%)																		
CH12																			

Tip: Ab openTx V2.00

Anstatt mit einem freien Hilfskanal und einem Mischerzeile kann ich auch in den Inputs eine Signal-Vorverarbeitung machen und die Bereiche umrechnen, mit gleichem Ergebnis für S1 und S2.

```

Input09
[I10] S1      Gewichtung (25%)  Offset (75%)  Quelle (S1)  Kein Trim
[I11] S2      Gewichtung (50%)  Offset (50%)  Quelle (S2)  Kein Trim
Input12
    
```

Das ist hier aber mal egal, viele Wege führen nach Rom.

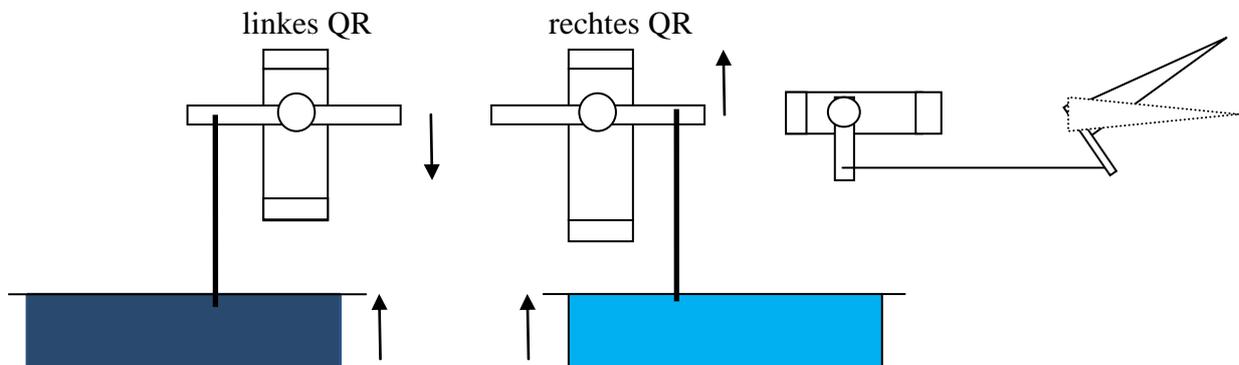
Beispiel: Querruder mit asymmetrischer Anlenkung, Spoiler variabel Speed-Flugphase

Zuerst mal das Grundprinzip:

Das Servo selbst steht auf Mitte = Null. Das Ruderhorn ist gerade aufgesetzt.

Aber die Gestänge sind so verlängert, dass beide Querruder jetzt schon ca 50% nach oben stehen! (Bei Crosslinkanlenkung sind die Gestänge zu kürzen).

Dann alle Klappen im Servo-Limit-Menü so einstellen, dass Min und Max an die mechanischen Limits der Klappe gehen und das Center exakt mittig zwischen den eingestellten Min/Max Werten liegen.



Für das rechte Querruder stellen wir ein:

Um die +50% Stellung des rechten QR zu korrigieren brauchen wir erst eine eigene Mischerzeile die das Querruder wieder auf Mitte stellt. (Feineinstellung mit Servotrim im Strack)
Der Trick: Wir machen die Korrektur nicht im Servo-Menü, sondern mit einer eigenen Mischerzeile!

1. MAX Gewichtung (-50%)

Dann kommt die normale Querrudersteuerung mit dem Knüppel und der Ruderdifferenzierung

2. [I2]Que Gewichtung (40%) Diff (50%)

Dann haben wir den Spoiler als Landehilfe, QR als Spoiler hochfahren, (von -50% auf +80% = 130%)

(Diese Zeile wirkt nur in der Flugphase Landung, Schalter SA↑ aktiv.)

Spoiler variabel per Inputverarbeitung [I5] des Gasknüppels 0 -100% gesteuert, Gas hinten→ Spoiler voll ausgefahren, Querruder bleibt noch wirksam (ca+20%)

3. [I5]Spoi Gewichtung (+130%)

Dann noch den Speedflug, beide Querruder um +5% hochstellen.

(Diese Zeile wirkt nur in der Flugphase Speed, per Schalter SA↓ diese Flugphase aktivieren)

4. MAX Gewichtung (+5%) Flugphase (Speed)

Für das linke Querruder sieht das so aus:

1. MAX Gewichtung (-50%), weil das QR wg dem angepassten Gestänge auf +50% steht

2. [I2]Que Gewichtung (-40%) Diff (50%), Querruder links mit Knüppel und Differenzierung

3. [I5]Spoi Gewichtung (+130%), Spoiler ausfahren

4. MAX Gewichtung (+5%) Flugphase (Speed), QR 5% hochstellen für Speedflug

Damit rechnen die Mischer intern richtig.

Die eigentliche Anpassung der Servodrehrichtung erfolgt wie immer im Servomenü

Jetzt darf man sich aber in der openTX Simulation nicht verwirren lassen.

Auf Grund der Zeile 1 zeigen jetzt beiden Querruder -50% an, tatsächlich stehen die zwei Querruder aber auf Mitte (eventl mit Servo-Subtrim noch sauber in den Strak stellen).

Und so sieht das in den Mischer aus: CH2 rechtes QR, CH5 linkes QR

CH	Mischer
CH1	[I1]Mot Gewichtung(+100%) Langsam/u2:d0)
CH2	MAX Gewichtung(-50%) [I2]Que Gewichtung(+40%) Diff(50%) [I5]Spoi Gewichtung(+130%) Flugphase(Land) Schalter(SA↑) MAX Gewichtung(+5%) Flugphase(Speed) Schalter(SA↓) (Speed)
CH3	[I3]H h Gewichtung(+100%)
CH4	[I4]Sei Gewichtung(+100%)
CH5	MAX Gewichtung(-50%) [I2]Que Gewichtung(-40%) Diff(50%) [I5]Spoi Gewichtung(+130%) Flugphase(Land) Schalter(SA↑) MAX Gewichtung(+5%) Flugphase(Speed) Schalter(SA↓) (Speed)

In den Inputs: [I5] Quelle Gasknüppel, umgerechnet auf 0-100% , Gasknüppel hinten→Spoiler voll ausgefahren, deshalb -50%

Input	Quelle	Gewichtung	Offset	Flugphase	Schalter
[I1]Mot	Quelle (SB)	100%			[Motor]
[I2]Que	Quelle (Que)	100%			
[I3]H h	Quelle (Höh)	100%			
[I4]Sei	Quelle (Sei)	100%			
[I5]Spoi	Quelle (Gas)	-50%	Offset (50%)	Flugphase (Land)	Schalter (SA↑) [Spoiler]
Input06					

In den Flugphasen: (kann man auch ohne Flugphasen machen!)

Flugphase 0(default)	Flugphase 1(Speed)	Flugphase 2(Land)	Flugphase 3	Flugphas
Name: Land				
Schalter: SA↑				
-				
Name: Speed				
Schalter: SA↓				
-				

Das sind hier nur Demowerte um das Prinzip für die Asymmetrischen Querruder zu verstehen. Auch die Motorsteuerung beim Segler kann man anpassen, hier einfach mit dem Schalter SB (0%, 50%,100%) und den Spoiler auf einen Schieber statt Knüppel legen. Da ist vieles Geschmacksache und man muss es Handlingbedingt eh immer anpassen.

Tip: <http://www.rcrcm.com/pdf/RCRCM-Airframes-ail-flap.pdf>

Beispiel: Eigene Ansagetexte als *.wav Dateien erzeugen

In den Taranis-, opentx-, und companion9x-Foren gibt schon viele fertige Textansage-Files für Flugmodelle, Hubis, Schiffe und Autos als deutsche wav-Dateien die man downloaden kann.

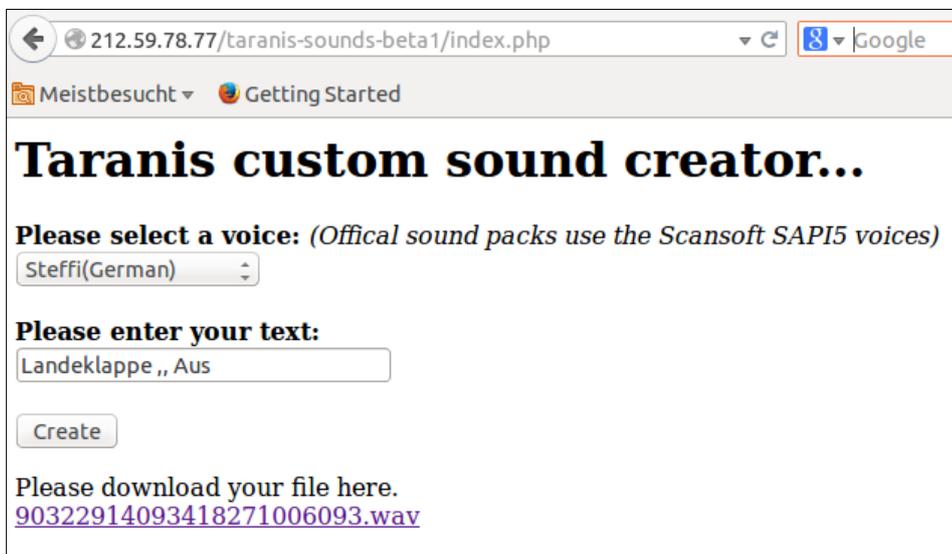
Es gibt viele Möglichkeiten Wav-Dateien für die Taranis selber erzeugen.

- Mit dem Windows XP eigenem Audiorekorder, das ist auch schnell und praktisch
- Auf der opentx Hauptseite: <http://www.open-tx.org/> gibt es 2 gute Programme, leider nur für Win7 **Open Tx Recorder** und **Open Tx Speaker**
Downloads: <http://www.open-tx.org/downloads.html>

Ein sehr gutes Freeware Online Programm um Sprachdateien direkt zu erzeugen

Hier: <http://212.59.78.77/taranis-sounds-beta1/index.php>

Rechner egal, Betriebssystem egal, Win, Mac, Linux, ganz egal, Einfacher geht es nicht
Dann Datei downloaden und Datei umbenennen (max 7-8 Zeichen)



← 212.59.78.77/taranis-sounds-beta1/index.php Google

Meistbesucht Getting Started

Taranis custom sound creator...

Please select a voice: (Official sound packs use the Scansoft SAPI5 voices)

Steffi(German)

Please enter your text:

Landeklappe „ Aus

Create

Please download your file here.
[90322914093418271006093.wav](#)

Sprache umstellen auf Steffi, Text eingeben

Create erzeugt ein passendes *-wav File

Anhören ob ok, eventl ändern

Downloaden und umbenennen auf max 8 Zeichen

Tip: Mit Kommas „, zwischen den Worten kann man Zeitverzögerungen einfügen.

Eingabe: Hundert als Text erzeugt „Einhundert“ als Ansage

Eingabe: 100 als Zahl erzeugt „Hundert“ als Ansage

openTx für Taranis Anleitung Deutsch

- Mit Balabolka geht das auch ganz gut und ist Freeware

- Mit div anderen TTS und Soundprogrammen: Stichwort: **TTS = Text To Speech**

Datei-Format: Wav-Datei PCM, Mono, 8Khz, 16kHz oder 32 KHz, **ohne Anhang, kein ID3-Tag**

Achtung: Kurze Dateinamen verwenden, **max 7-8 Zeichen für den Dateiname** Dateityp: wav

Die Sound *.wav Dateien stehen alle auf der SD-Karte unter Sounds dort muss es passende Unterverzeichnisse geben:

\Sounds\de für Deutsch \Sounds\en für Englisch

Dann gibt es dort noch je ein Unterverzeichnis \System für die internen Sounds des Betriebssystem
Sounds\de\System Sounds\en\System



Einfach mal abspielen, die meisten Namen sind selbsterklärend und müssen genau so heißen

Will man ein paar Ansagen nicht haben, dann diese Datei einfach umbenennen, nicht löschen!

Dann wird die Datei nicht gefunden und es kommen keine Ansagen.

Dafür kommen die internen Warn-, Signal- und Piepstöne

Tip für System-Dateien umbenennen

"**Poti zentriert**" diese Ansage finde ich lästig,

Datei **midpot.wav** umbenennen dann kommt nur kurzer Pieps

"**10sec**" "**10sec**" "**10sec**" beim Countdown, hier ist noch ein Fehler,

Datei **timer10.wav** umbenennen dann kommen nur 10 Pieps

Eigene System-Dateien kann man auch erstellen, müssen aber die gleiche Namen haben, wenn sie einen System-Standardtext ersetzen sollen! Unter 9xforums gibt eine Liste der wav-Dateien

Beispiel: Schalterstellungen und/oder Schalterwert ansagen

Der Unterschied von Spiel Sound, Sag Wert und Spiel Ton:

- Für eigene Ansagen brauchen wir auch eigene Wav-Dateien als Sound-Dateien
- Für die Ansage von Werten brauchen wir nichts extra, das setzt sich openTx selbst zusammen
- Töne sind 16 fertige „Geräusche“, Sirenen, Robot, usw.

Wir wollen uns mal die Schalterstellungen ansagen lassen: SG unten, SG oben, SG mitte

Dazu müssen wir erst mal 3 eigene Ansagen erzeugen, das geht ganz schnell damit:

<http://212.59.78.77/taranis-sounds-beta1/index.php>

Auf Steffi umschalten, dann einfach den Text eingeben, CREATE, und schon erhält man das passende *.wav File zum Download

Datei aufrufen, abspielen und testen bis sie ok ist.

Datei umbenennen (max 8 Zeichen) und auf der SD-Karte unter **Sounds/de** abspeichern.

SG_mitte.wav, SG_unten.wav, SG_oben.wav

Tip:

Hundert --> wird zu "Einhundert", 100 --> wird zu "Hundert"

Nicht SG sondern mit Leerzeichen S G, dann wird jeder Buchstabe einzeln gesprochen

In den Spezialfunktionen können wir jetzt diese 3 wav-Dateien aufrufen, das war es schon.

Konfiguration				Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Spezial Funktionen	Telemetrie
Schalter	Aktion	Parameter	Aktivieren									
SF1	SG ↑	Spiel Sound	SG unten	▶	Keine Wiederholung							
SF2	SG-	Spiel Sound	SG mitte	▶	Keine Wiederholung							
SF3	SG ↓	Spiel Sound	SG oben	▶	Keine Wiederholung							
SF4	SA ↓	Sag Wert	S1		Keine Wiederholung							
SF5	---	Override CH1	0		<input type="checkbox"/> EIN							

Schalterwerte oder Telemetriewerte ansagen

Das ist viel einfacher.

Jeder Schalter liefert von sich aus schon mal 3 Werte -100% 0% +100%

Potis oder Knüppel liefern Analogwerte von -100% bis +100%

Telemetriewerte erhalten automatisch auch noch passende Einheiten angefügt.

Hier mal mit einem logischen Schalter das Poti S2 auf eine Veränderung abfragen

$|d|>=x$ Delta = Differenzfunktion

L1: Wird nur aktiv wenn sich S2 etwas (hier um 2 Prozent) verändert hat.

Konfiguration					Heli	Flugphasen	Inputs	Mischer	Servos	Kurven	Logische Schalter	Spezial F
	Funktion	V1	V2	UND Schalter								
L1	$ d >=x$	S2	2	---								0,0

SF5: Wenn L1 aktiv ist, dann wird der Wert des Potis S1 einmal angesagt.

SF5	L1	Sag Wert	S2	Keine Wiederholung
SF6	---	Override CH1	0	<input type="checkbox"/> EIN
SF7	SB ↑	Sag Wert	SB	Keine Wiederholung
SF8	SB-	Sag Wert	SB	Keine Wiederholung
SF9	SB ↓	Sag Wert	SB	Keine Wiederholung

Genauso geht es mit der Ansage von Schalterwerten hier SB-Schalter bei SF7, SF8, SF9 oder auch mit der Ansage von Telemetriewerten incl. den passenden Einheiten

Das kann man jetzt noch per Logik oder per Schalter sperren und freigeben

Beispiel: Eigene Sprachdateien mit Programm Balabolka erzeugen (Freeware)

Balabolka aktuelle Version downloaden, Entpacken und Programm in Deutsch installieren

Text to speech auf Deutsch umstellen unter SAPI 5

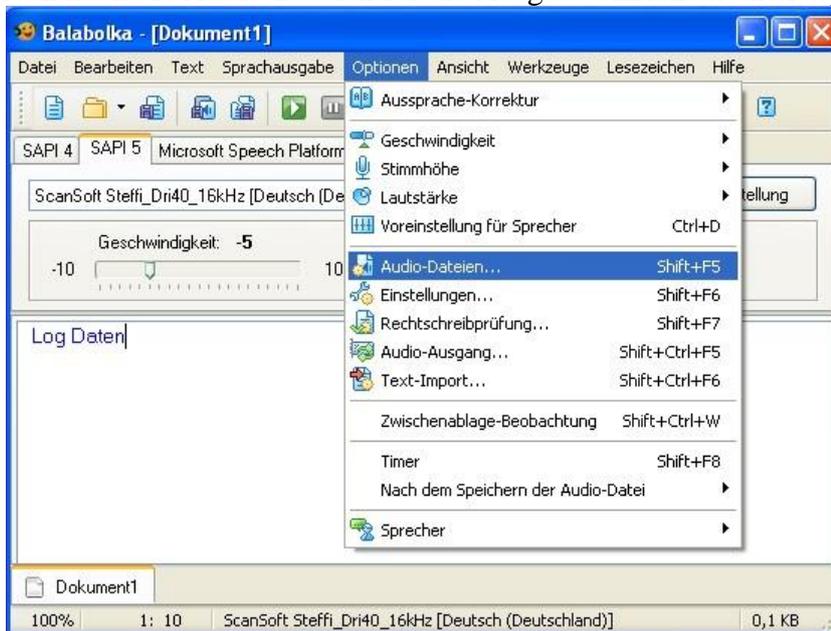
Da kann es passieren dass je nach Betriebssystem XP, Vista, Win 7, Win 8, eine deutsche Sprachansage von Microsoft nachinstalliert werden muss, denn alle Windows-Systeme haben als Standard nur Englisch installiert.

Zumindest braucht man noch die Datei RSSolo4German.zip. Dann entpacken und installieren

Name	Größe	Typ	Geändert am
balabolka.zip	9.356 KB	WinRAR-ZIP-Archiv	01.01.2014 09:58
history.eng.txt	20 KB	CNC Syntax Editor file	02.11.2013 09:50
history.rus.txt	22 KB	CNC Syntax Editor file	02.11.2013 09:52
lhhtsged.exe	2.243 KB	Anwendung	01.01.2014 10:11
license.eng.txt	5 KB	CNC Syntax Editor file	24.04.2012 15:54
license.rus.txt	5 KB	CNC Syntax Editor file	30.07.2013 13:09
readme.txt	3 KB	CNC Syntax Editor file	02.11.2013 09:53
RS5olo4German.zip	21.213 KB	WinRAR-ZIP-Archiv	01.01.2014 10:27
RS5olo4GermanSteffi.exe	21.973 KB	Anwendung	24.10.2005 11:17
setup.exe	9.415 KB	Anwendung	02.11.2013 11:12

Damit hat man als SAPI 5 die ScanSoft Steffi_Dri_16Khz als Sprache installiert und kann sie aufrufen.

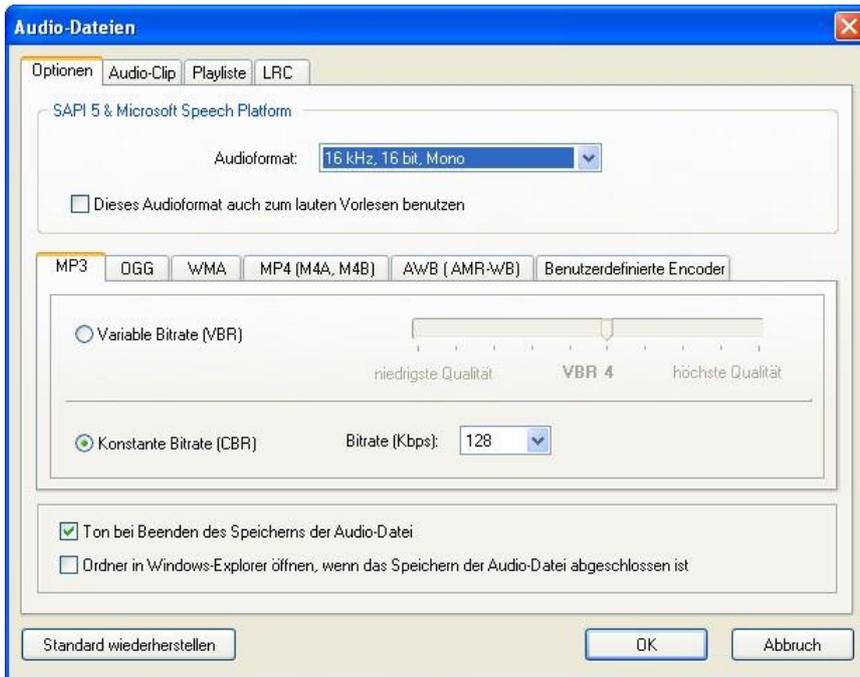
Nun noch die Audio-Dateien für die Ausgabe einrichten unter Optionen, Audio-Dateien...



Wir müssen eine wav Datei erzeugen als **Mono**, 16KHz, 16 bit, **Kein Stereo!** (8Khz oder 32 KHz geht auch).

Keine ID3 Tags oder sonst was einrichten, wir brauchen eine reine, nackte WAV-Datei!

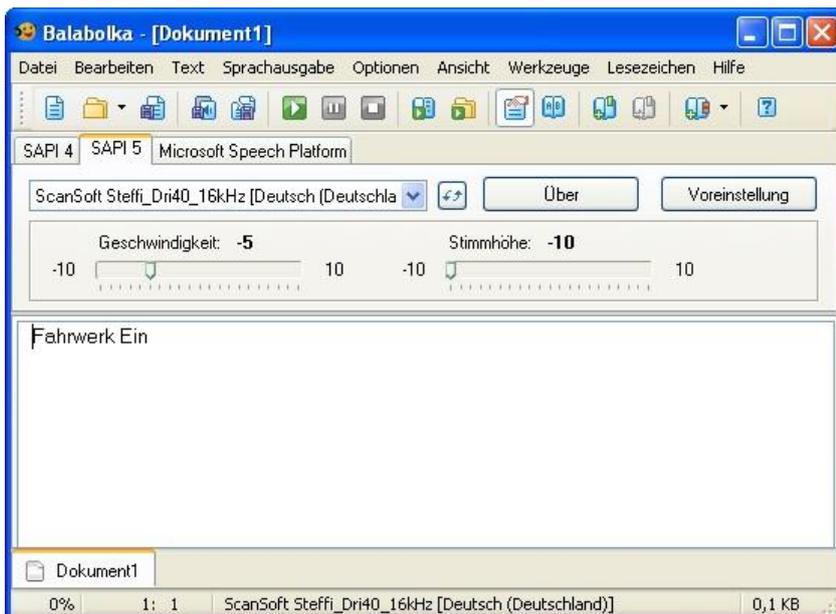
openTx für Taranis Anleitung Deutsch



Dann können wir den ersten Text eingeben und testen.

Also Text eingeben, dann Text markieren bzw Cursor nach ganz vorne und mit grünen Pfeil mal ablaufen lassen.

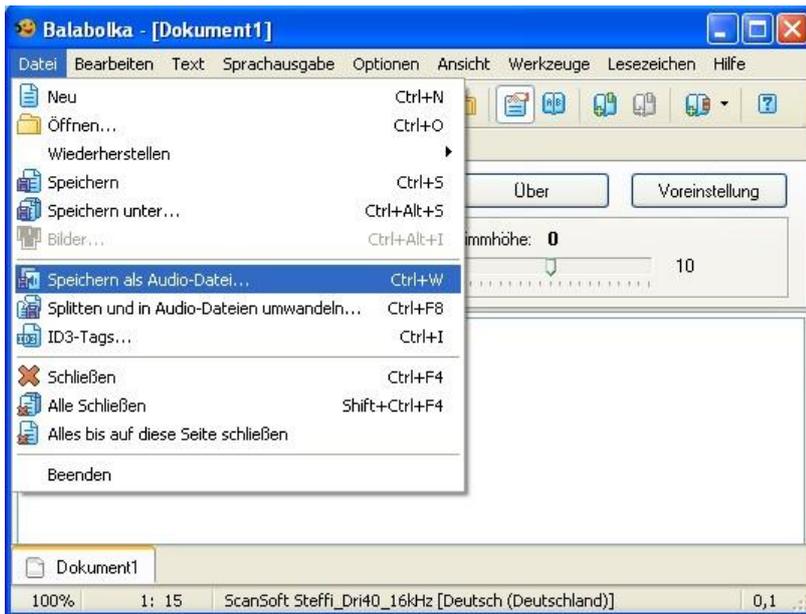
Geschwindigkeiten und Stimmhöhen anpassen bis es ok ist.



Wenn das ok ist dann unter Datei, Speichern als Audio-Datei...

Am besten ein eigenes Unterverzeichnis anlegen und dort abspeichern

openTx für Taranis Anleitung Deutsch



So und jetzt nochmal:

Kurze Namen verwenden, max 7-8 Zeichen, nicht mehr, die SD-Karte kann nur 7-8 Zeichen

Alles was auf der SD-Karte steht muss unter **Sounds\de** oder/und unter **Sounds\de\System** rein

Alles was in **Sounds\de** steht da kann man den Namen und Inhalt frei vergeben.

Alles was in **Sounds\de\System** steht da **muss** der Name so beibehalten werden, damit der Prozessor auf diese Systemmeldungen zugreifen kann. Der Inhalt kann aber beliebig sein!

Alles Sound-Files die auf der SD-Karte stehen müssen auch im PC unter Companion9x stehen, damit man richtig programmieren kann und genau die gleichen Files auswählen kann.

Dazu in companion9x das Verzeichnis unter Einstellungen richtig einrichten.



Beispiel: Modellname einmal bei Aufruf ansagen lassen

Beim Aufruf eines Modells kann man sich das Modell ansagen lassen.

Genauer: Man kann steuern, dass eine bestimmte *.wav Datei einmal ausgeführt wird.

Die *.wav-Datei erstellt man wie oben beschreiben, mit Balabolka oder dem Taranis custom sound creator.

Der Name der Wav-Datei muss ganz exakt gleich sein wie der Name des Modells, das man aufruft, keine Leerzeichen, keine Sonderzeichen

Der Inhalt kann beliebig sein!

Die Datei darf max 7-8Zeichen lang sein, keine Sonderzeichen, keine Umlaute, keine Leerzeichen enthalten. Also nicht Delta 2 sondern Delta-2

→Darauf sollte man schon achten wenn man das Modell neu anlegt!

Diese *.wav Datei muss auf der SD-Karte im Verzeichnis **Sounds/de/** stehen.

In den Spezialfunktion muss als **SF1 (an erster Stelle!)** der Aufruf des Modells stehen.



SF1 One Funktion einmal bei Aufruf des Modells ausführen

Sag Text <Modellname>

1x: einmal ausführen (im Gegensatz zu **!1x** d.h. einmal aber nicht beim Start)

Alternative und Erweiterung :

Auf SD-Karte ein weiteres Verzeichnis anlegen mit dem Modellname (z.B. ASW17)

Sounds/de/Modellname/ also **Sounds/de/ASW17/**

und dort dann **alle** wav-Dateien für diese Modell reinstellen.

Modellname und wav-Dateiname müssen exakt gleich sein, wie oben beschrieben.

Automatischer Aufruf wie oben beschreiben: **SF1 ONE Sag Text ASW17 1x**

Dann kann man sich auch Flugphasen automatisch ansagen lassen.

Dazu gibt es zur Wav-Datei 2 zusätzlich Parameter **-ON** und **-OFF**

Auch hier muss die *.wav Datei exakt so heißen wie die Flugphase, max 6-7 Zeichen lang
z.B: Landung-ON.wav Thermik-ON.wav Speed-OFF.wav

Aber Achtung: Dateiname und Parameter zusammen nur maximal 10 Zeichen!

(6-7 für den Namen der Flugphase und dann noch 3-4 für den **-ON -OFF** Parameter)

\Soundes\de\ASW17\Landung-ON.wav

Beispiel: Splashscreen für Taranis anpassen und erzeugen

Splashscreen für Taranis im BMP-Format mit 212x64 Pixel S/W (eigentlich 4bit=16 Graustufen)

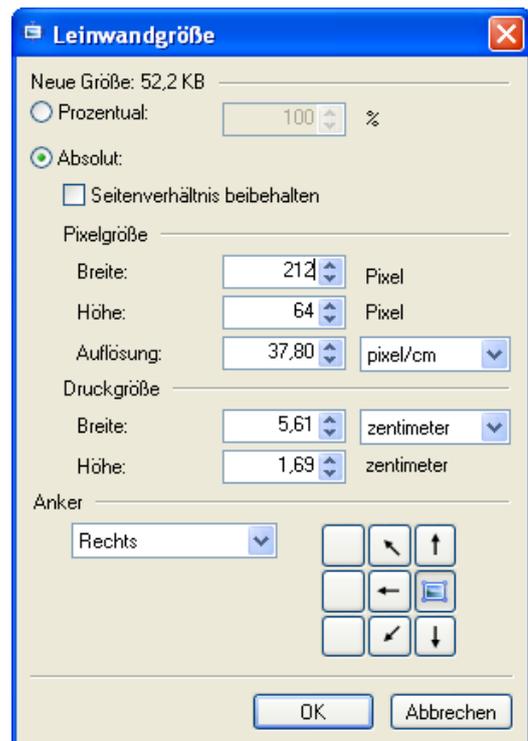
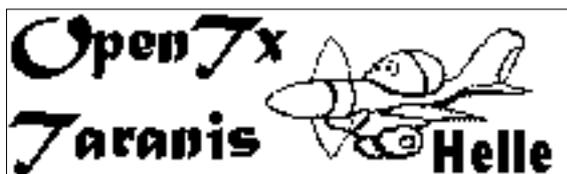
Das bisherige 9x-Format ist 128x64, da gibt es hunderte sehr schöner Splashscreen

Link: <http://openrcforums.com/forum/viewforum.php?f=43>

Wenn man die in companion9x lädt werden die aber auf 212x64 verzerrt, das ist Mist!

Mit dem **Programm Paint.net** kann man die 128x64 Formate reinladen, dann mit Bild, Zeichenbereichgröße auf 212x64 einstellen, Teilbild nach rechts Mitte wählen, dann hat man links freien Platz für eigenen Text

Das geht recht flott, Rest ist wie jedes Zeichenprogramm Farbe Vordergrund, Farbe Hintergrund, Pinsel, Ausschnitte usw. Helligkeit und Kontrast anpassen da ja nur Schwarz/Weiß möglich. Abspeichern unter BMP-Format, Bit-Tiefe auf Auto-detect



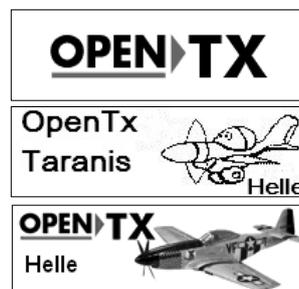
Beispiel: Splashscreen (Start-Screen) einfügen oder ersetzen

Startbilder müssen ein ganz bestimmtes Format haben:

Taranis: 212x64 Punkte, max 4 Bit (16 Graustufen)

Th9, 9XR: 128x64 Punkte, 2 Bit (schwarz, weiß)

- In den **Profilen** kann man eine Startbild auswählen
- Unter **Editiere Sender Start Bild** kann man ein vorhandenes Startbild in der Firmware ersetzen.



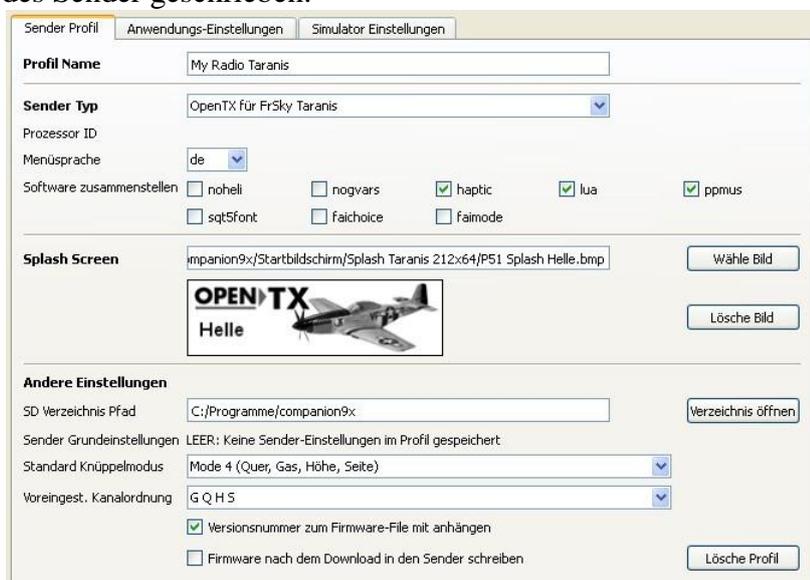
Das sind aber 2 völlig verschiedene Verfahren!

Downloaded man eine neue Version einer Firmware ist dort immer der  Startscreen enthalten.

Splashscreen in den Profilen:

Profile sind Grundeinstellungen um eine Firmware mit div Optionen zusammenzustellen.

Hier wird ein Startbild vorab ausgewählt, **aber erst beim direkten** Flashen/Brennen in den Sender (das geht wie bisher auch mit dem DFU-Util) **zusammen mit der Firmware** in den Flash-Speicher des Sender geschrieben.



Editiere Sender Start Bild:

Hier wird **das vorhandene Startbild** in einer **vorhandenen Firmware** ersetzt.

Das muss von „**Hand**“ gemacht werden. Damit kann diese Firmware dann auf die **SD-Karte kopiert** werden, via Bootloader geflasht und der eigene Startscreen wird beim Einschalten sichtbar.

Ansonsten bleibt es beim Symbol 

Symbol:



Beispiel: Textdateien als Checkliste auf das LCD-Display bringen

Beim Start des Senders, bzw beim Aufruf eines Modells können eigene, kurze Texte zur Anzeige gebracht werden. Das ist für alles Mögliche gut, Einstellungen, Schalter prüfen, Namen,

Dazu bedarf es ein paar einfacher Regeln.

Es müssen einfache, kurze Text-Dateien sein, die mit einem einfachen Editor erstellt und im ANSI-Format abgespeichert werden (z.B. mit dem Windows Editor).

Am Besten die Schriftart Courier, Normal, 12Pkt dann hat man eine Blockschrift

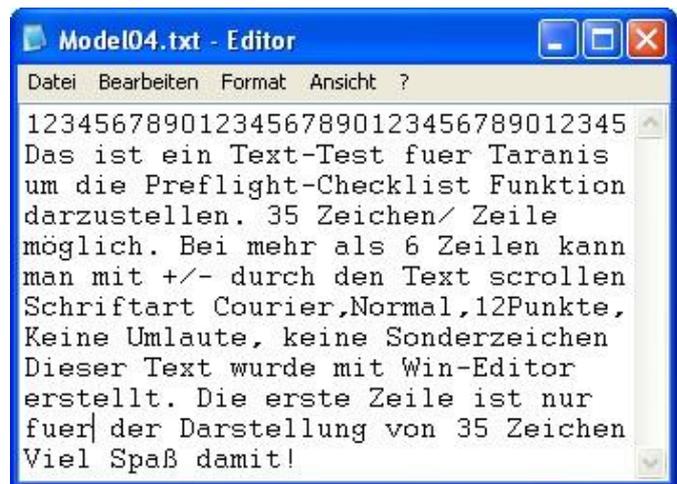
Keine Umlaute ä,ü,ö, keine Sonderzeichen

Pro Zeile 35 Zeichen, nicht mehr!

Editor muss CR/LF beim Zeilenumbruch einfügen, sonst werden die Zeilen nicht richtig dargestellt!

Bei mehr als 6 Zeilen kann man Scrollen

Der Dateiname muss **ganz genau** so heißen wie das Modell für das der Text ist.



z.B.

Model01.txt

Model02.txt

Delta.txt

Hexakopter.txt

Diese Dateien müssen auf der SD-Karte im Unterverzeichnis **MODELS** stehen

In den Modelleinstellungen 2/13 muss das Häkchen **Display Checklist** gesetzt sein

Dann erscheint nach der Gas-Warnng und der Schalterwarnung der Text

Beispiel:



Beispiel: Input Signalvorverarbeitung, statt Mischer mit einem freien Hilfskanal

Die Inputs wurden stark erweitert.

Die Signalverarbeitung in den Inputs und die Mischerverarbeitung im Kanal sehen fast gleich aus. Vieles was man bisher nur in einem freien Hilfskanal und einer Mischerzeile vorberechnen konnte, kann man jetzt im Signal-Input direkt vorverarbeiten.

Hier: Edit Input10: S1 Bereich umrechnen von +/-100% auf 50% bis 100%

Gewichtung = $\text{Spanne}/200 = 50/200 = 25\%$

Offset = Mitte den neuen Bereichs 50 bis 100% = 75%

Selbst Kurven, Differenzierungen oder Expofunktionen sind möglich

und das auch noch ein- oder beidseitig.

Trimmungen in den Inputs **und** den Mischer aktivieren, damit sie zum Kanal „durchgereicht“ werden.

Input Signalvorverarbeitung

Mischer Vor-Verarbeitung im Kanal

[I10] S1	Gewichtung (25%)	Offset (75%)	Quelle (S1)	Kein Trim
Input11	Gewichtung (50%)	Offset (50%)	Quelle (S2)	Kein Trim

Bei den Inputs kann pro Input-Signal aber immer nur jeweils 1 Zeile aktiv sein!

Hat man mehrere Zeilen (z.B Dualrate umschaltbar) muss man jede Zeile per Schalter aktivieren / deaktivieren.

Bei den Mixern **können pro Kanal mehrere Zeilen aktiv** sein, da man sie verrechnen kann Addiert, Multipliziert, oder Replace = Ersetze

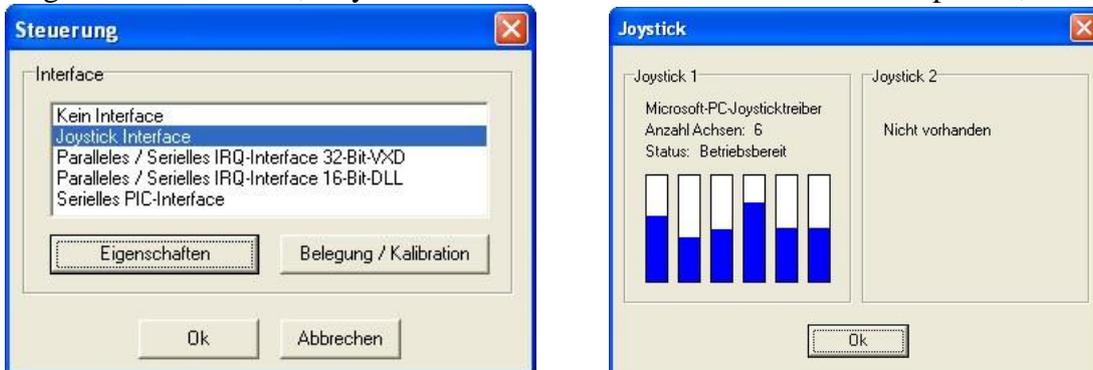
Beispiel: Taranis Sender am Flugsimulator anstatt eines Joystick verwenden

Das ist ab opentx V2.05 besonders einfach und man braucht kein PPM to USB Interface!

In der Firmwareszusammenstellung muss man Joystick auswählen, dann geht das automatisch. Sender einschalten, hochfahren, ein (Simulator)-Modell wählen, Modell muss im Schülermodus sein, damit es PPM-Signale liefert!

Erst jetzt USB einstecken, damit wird der Sender automatisch als Standard PC-Joystick erkannt.

Flugsimulator aufrufen, Joystickinterface auswählen und die Kanäle anpassen, das wars.



Hier: Der FMS-Flugsimulator kann nur 6 Kanäle auswerten

Andere Flugsimulatoren können alle 8 Kanäle und 8 Schalter auswerten.

Von der Taranis kommt: Kanal 1-8 und als Analog-Werte und Kanal 9-16 als Schalter

Hintergrund: Die Taranis wird unter Windows automatisch als HID Gamecontroller erkannt. Siehe: Start, Systemsteuerung, System, Hardware, Gerätemanager, dort überprüfen.



Damit kann man auch den Sender unter Companion am Simulator verwenden!

Tip: Man kann auch mit einem ForceFly Profi-Joystick ein Modell fliegen.

<http://emrlabs.com/index.php?pageid=1>

Varianten des USB Verbindungsaufbau mit Taranis X9D, X9E

- 1. Taranis ausgeschaltet** => USB anschließen => STM32 Bootloaderfunktion aktiv (DFU, Zadig)
- 2. Taranis eingeschaltet** => USB anschließen => Joystickfunktion aktiv
- 3. Taranis Trims halten, einschalten** => USB anschließen => SD-Datenträger, Update, Boot

Teil E Companion V2.xx und openTx V2.xx

Für die Installation braucht man Administratorrechte, da Laufwerke eingerichtet werden!

Die bisherige Vollversion am PC ist Companion9x mit V1.52 und die opentx im Sender ist r2940 r2942 r2943 mit Modell-EEProm V215. Diese kann man weiterhin verwenden.

Jetzt gibt es den Nachfolger:

openTx Companion V2.00 und openTx V2.00 mit vielen Erweiterungen (V2.017, V2.10)

Diese gibt es hier, falls nicht ein automatisches Update geladen wird.

Die opentx Hauptseite: <http://www.open-tx.org/>

Companion: <http://downloads-20.open-tx.org/companion/> *.exe Datei für Companion

Opentx: <http://downloads-20.open-tx.org/firmware/binaries/>

*.bin Dateien für Taranis *.hex-Dateien für Th9x und 9XR

Die Dateinamen enthalten die jeweils gewählten Optionen.

Index of /			
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	
 companion/	03-Jun-2014 00:17	-	
 favicon.ico	04-Jun-2014 17:00	1.1K	
 firmware/	04-Jun-2014 20:47	-	
 index.php	20-May-2014 10:11	54	

 companionInstall_1.99.4.exe	28-May-2014 13:56	8.9M
 companionInstall_1.99.5.exe	30-May-2014 12:55	8.9M
 companionInstall_1.99.6.exe	31-May-2014 00:43	8.9M
 companionInstall_1.99.7.exe	01-Jun-2014 01:22	8.9M
 companionInstall_2.0.0.exe	03-Jun-2014 00:17	8.9M

Index of /firmware/binaries	
 Parent Directory	
 opentx-taranis-de.bin	
 opentx-taranis-noheli-lua-en.bin	
 opentx-taranis-lua-haptic-sqt5font-se.bin	
 opentx-taranis-ppmus-de.bin	
 opentx-taranis-en.bin	
 opentx-9x-heli-templates-en.hex	

Auf das aktuelle Datum gehen, dann kommt man in die Unterverzeichnisse für CompanionInstall und die Betriebssystemdateien für die Sender (*.bin Dateien)

Bitte immer nur die aktuellsten Versionen laden, da täglich Fehler behoben werden und Erweiterungen erfolgen!

Wer schon eine Companion9x V1.52 installiert hat, muss auf 2-3 Dinge achten.

Die bisherige EEPROM Version ist V215 . d.h. alle Modelle auf dem PC und auf der Taranis sind in diesem Format gespeichert.

Die V2.00 hat als EEPROM die Version 216.

Diese ist nicht kompatibel zu V215. Es gibt keine zurück von V216 auf V215!

openTx für Taranis Anleitung Deutsch

Somit zuerst seine Modelle vom Sender auf den PC sichern und dann unter einem anderen Namen und anderem Verzeichnis zusätzlich abspeichern

Ansonsten die Version 2.00 normal installieren, aber unter einem anderen Namen, z.B. CompanionV200 statt Companion9x.

Nach der Installation der neuen Version V2.00 einen anderen Speicherpfad für die EEPROM Modelldateien verwenden, damit nicht versehentlich die V215 zerstört wird.

Damit hat man beide Versionen getrennt auf dem PC und das ist auch gut so.

Wer nur mal die neue Version testen und simulieren will kann das am PC tun und lässt auf dem Sender seine r2940 drauf, die er wie bisher mit Companion 1.52 bedient.

Er kann aber auch die Version opentx200.bin auf den Sender flashen und damit direkt arbeiten.

CompanionV2.00 ist komplett eingedeutscht, wer bisher mit Companion9x V1.52 gearbeitet hat findet sich schnell zurecht. Manche Funktionen gibt es erst ab opentxV2.00 z.B. beim Vario sind jetzt die Töne und Frequenzen frei einstellbar.

Die neuen Modell-Template Funktionen brauchen die LUA-Option, da sie mit einem LUA-Skript arbeiten. Dazu muss auf der SD-Karte das Unterverzeichnis **/SCRIPTS/WIZARD** angelegt werden.

Dort alle LUA Skripte+Bilder für Modellgenerator reinkopieren.

Beim Anlegen eines **neuen** Modells wird daraus dann das LUA-Skript automatisch gestartet.

Wer den Sender zum ersten mal auf opentxV2.00 flasht (so wie bisher mit DFU-Util und installiertem Zadig-Treiber unter V1.52) und dann neu startet sieht, dass die Modelle automatisch von V215 auf V216 konvertiert werden. Das kann durchaus mal 1 min dauern, wird aber angezeigt. Man kann jederzeit wieder auf r2940 zurückflashen, muss aber dann auch die Modelle im Sender zuerst löschen und dann wieder im V215 Format zurückladen.

Nach der Installation kann man die Testversion aufrufen und sie startet mit dem neuen Logo



Das Hauptmenü hat eventl. andere Symbole (sind einstellbar!) aber sonst die gleichen Funktionen

CompanionTx V2.00x Start und Senderprofil anlegen

Nach der Installation von Companion V2.00 (aktuelle V2.015) muss man ein paar Dinge einrichten damit Companion und die Simulation am PC genau so laufen wie am Sender selbst.

Senderprofil anlegen:

Damit wird die richtige Software für den eigenen Sender ausgewählt, die Optionen zusammengestellt und man kann sie dann downloaden und abspeichern.

Wer mehrere Sender hat bitte ein passendes Verzeichnis anlegen!

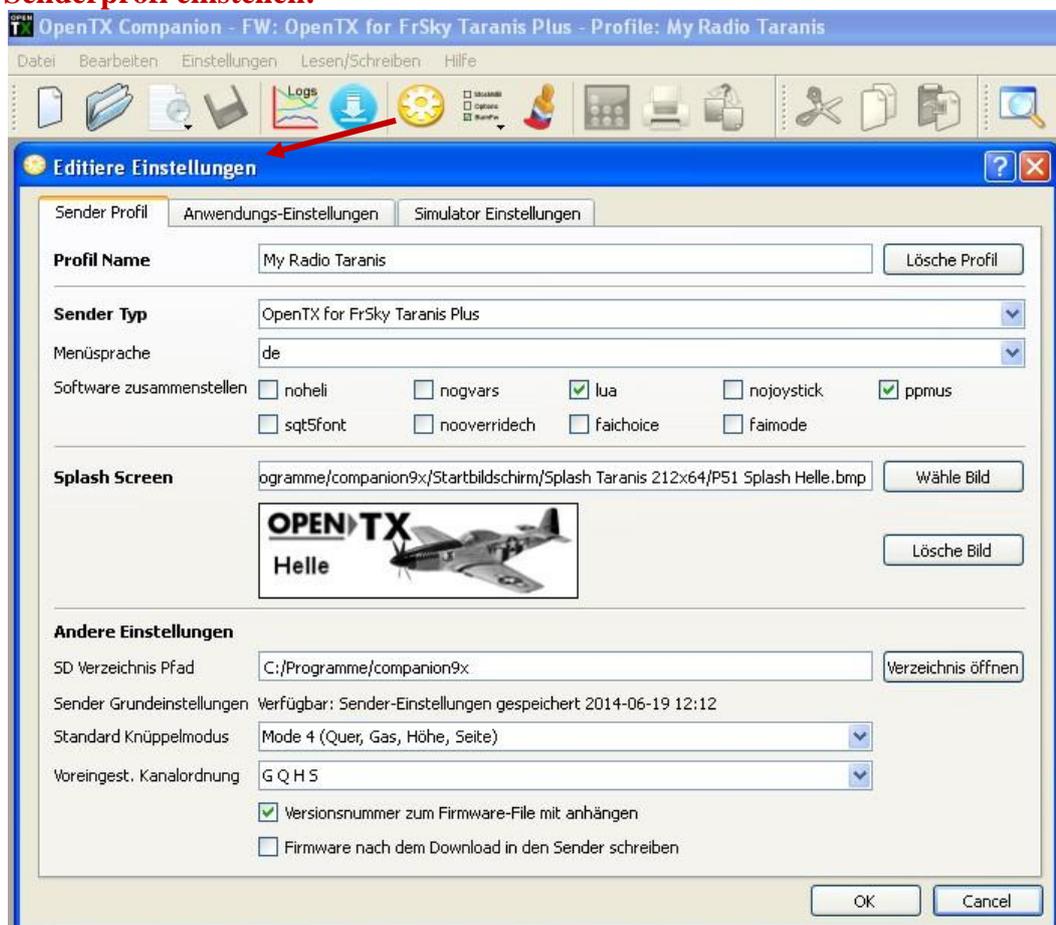
→ **Aufpassen dass wirklich der richtige Sender ausgewählt und zusammengestellt wird!**

Taranis oder Taranis Plus, oder 9XR 64k, X9R 128k, 9XRPro oder eigene Namen verwenden.

Da ist absolut wichtig, denn damit wird das Verhalten von Companion, der Simulation und der Firmware die man downloaded eingestellt. Man kann auch mehrere Senderprofile erstellen.



Senderprofil einstellen:



Nun werden wir das Senderprofil mit Werten versorgen.

Sendertyp, Menüsprache Deutsch (das ist nicht die Ansagesprache und Soundsprache)

Die Senderoptionen die für die Firmware zusammengestellt werden.

Splashscreen-Verzeichnis usw.

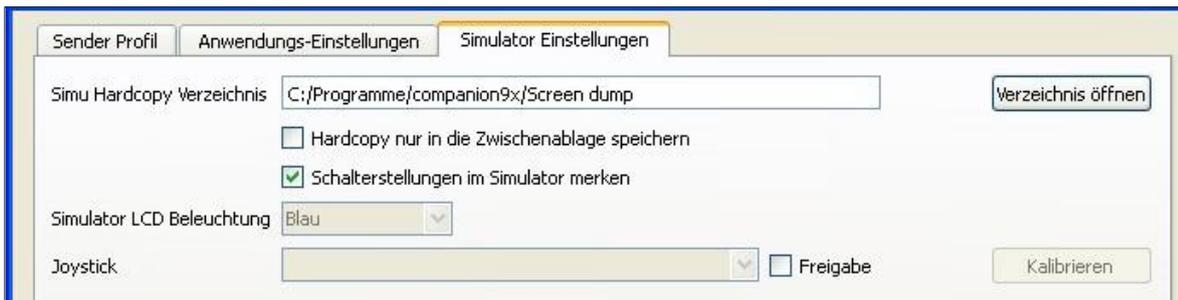
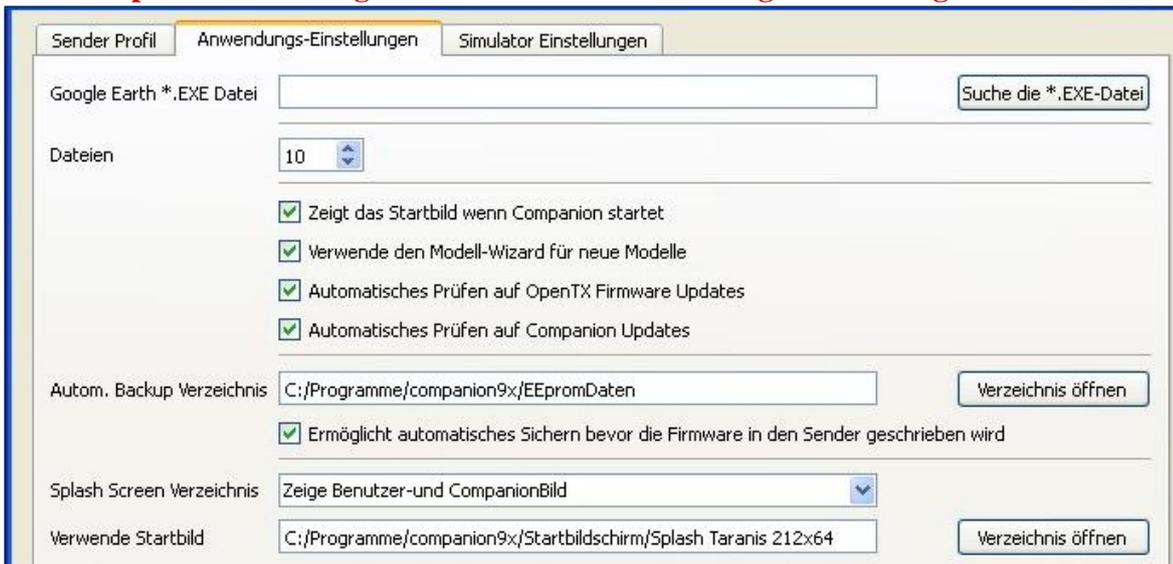
Der SD-Verzeichnis Pfad hat nichts mit der SD-Karte auf dem Sender zu tun!

Wir brauchen für die Simulation auf dem PC die exakt gleiche Verzeichnisstruktur mit den gleichen Dateien wie auf der SD-Karte auf dem Sender!

Wir greifen nicht auf den Sender zu, der Sender braucht nicht angeschlossen sein.
Also Verzeichnisse anlegen und Dateien dort rein.

Am Einfachsten kopiert man sich die SD-Karte auf den PC in den SD-Verzeichnispfad. z.B. C:\Programme\Companion2xx\SD-Karte und hat dann dort alle Dateien und Verzeichnisse zusammen.

Auch ein paar Anwendungs- und Simulationseinstellungen sind nötig.



Um am PC simulieren zu können, ohne dass ein Sender angeschlossen ist, muss auf dem PC die SD-Karte als Kopie mit allen Verzeichnissen und Dateien des Senders vorhanden sein. Falls etwas nicht funktioniert, dann erst man hier suchen!

Vorsicht wer mehrere Sender hat!

Aktiviertes Profil (Taranis statt Th9x), Firmwaredownload (9XR statt Th9x), tatsächlich angeschlossener Sender (TH9x) und Modelldaten mit Einstellungen (von der Th9x) müssen zusammenpassen!

Sonst zerschießt ihr euch eventl eure Modelldaten oder der Sender reagiert nicht mehr.

Firmware downloaden

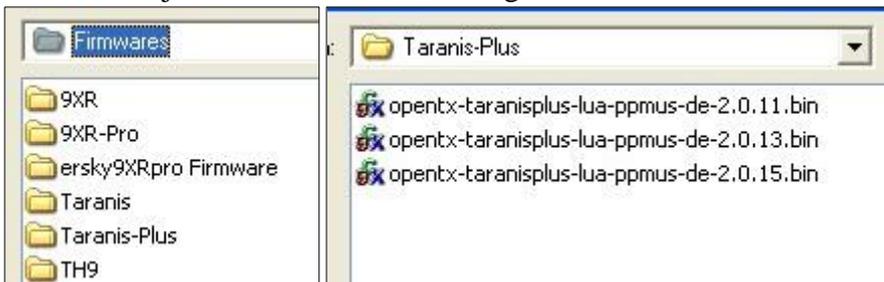


Jetzt brauchen wir für unseren Sender die richtige Firmware.

Das wird am Server automatisch zusammengestellt und hängt von unseren Einstellungen im Senderprofil ab.

Darum ist das Senderprofil so wichtig!

Ich habe für jeden Sender am PC ein eigenes Firmware-Verzeichnis eingerichtet.



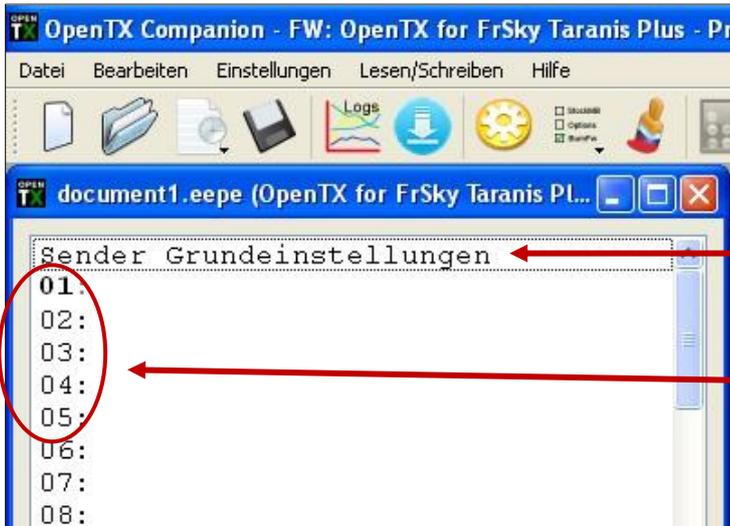
Hinweis: Beim Download der Sendersoftware aus CompanionTx ist der Name sehr lang und enthält alle Optionen.

Auf der SD-Karte und am LCD-Display sind aber nur ca 16-20 Zeichen lesbar. Deshalb den langen Dateinamen umbenennen und kürzen z.B. OpenTx_V2015 bevor man ihn auf die SD-Karte schreibt.

Damit haben wir jetzt die passende Sendersoftware als *.bin Datei zum flashen des Senders

openTx für Taranis Anleitung Deutsch
Jetzt kann man ein neues Modell anlegen und den Sender simulieren

Neues Modell anlegen

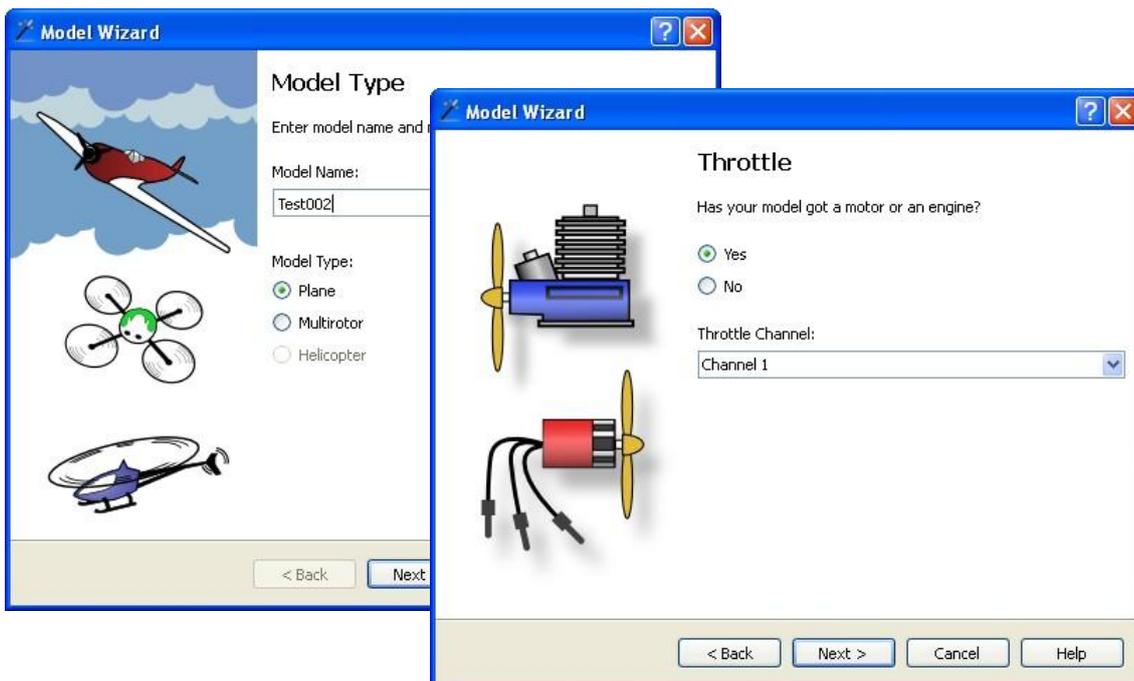


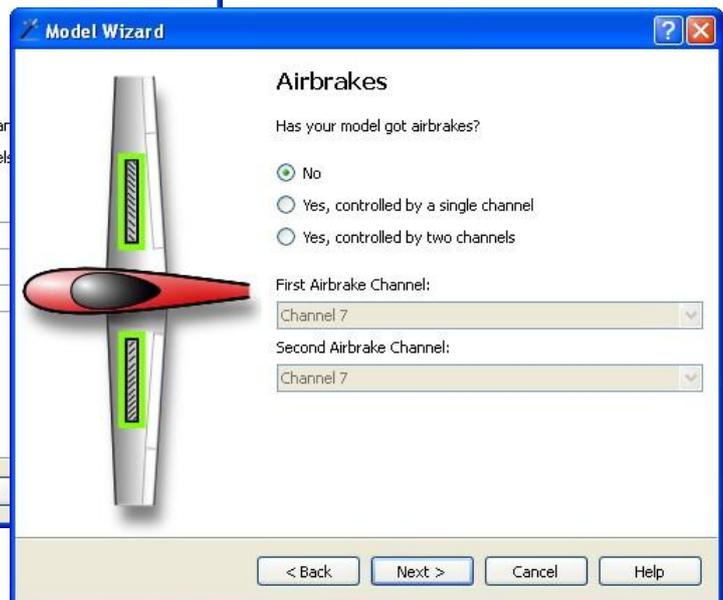
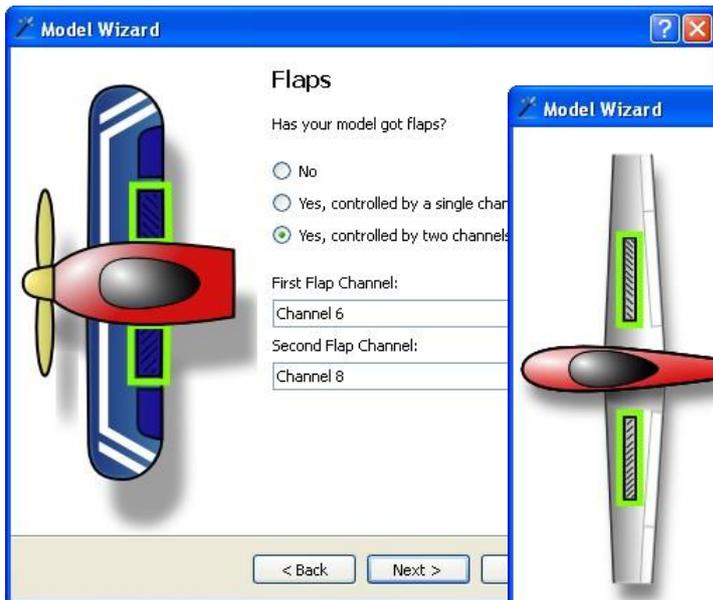
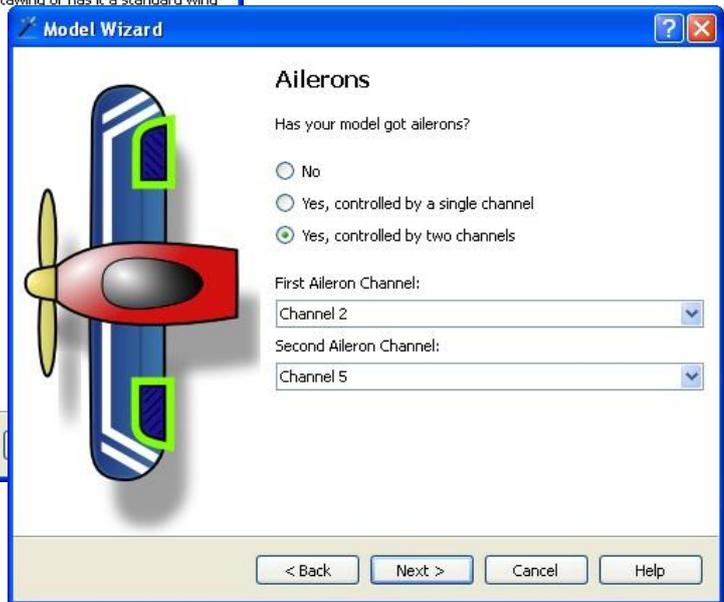
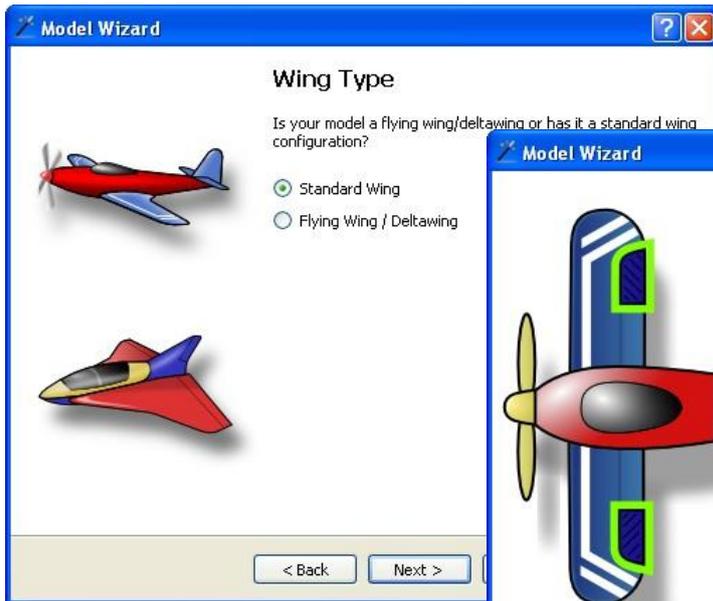
Die Sender Grundeinstellungen
(für alle Modelle)

Modelle neu anlegen und der Wizard
wird gestartet

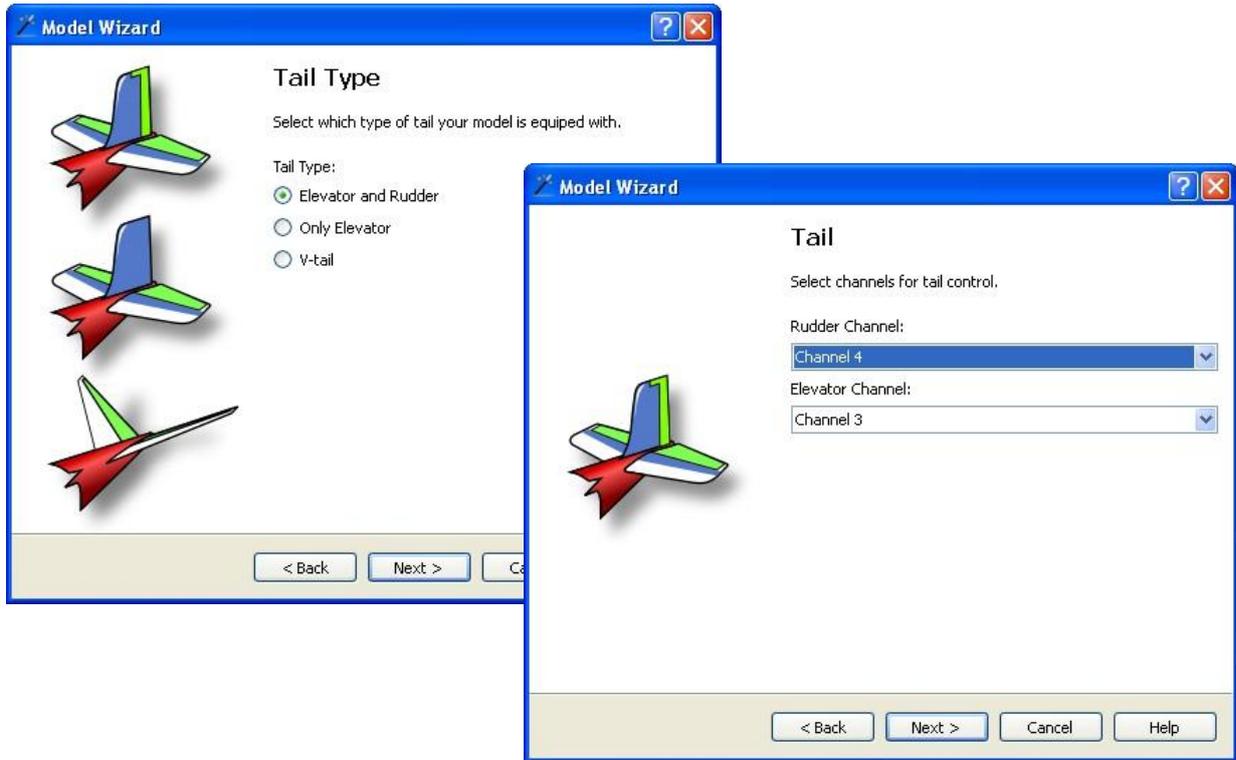
Der neuen Modell Wizard ab companion V2.00

Wird ein neues Modell angelegt, startet automatisch der neue Modell Wizzard und führt durch die möglichen Ruder und Kanaleinstellungen.

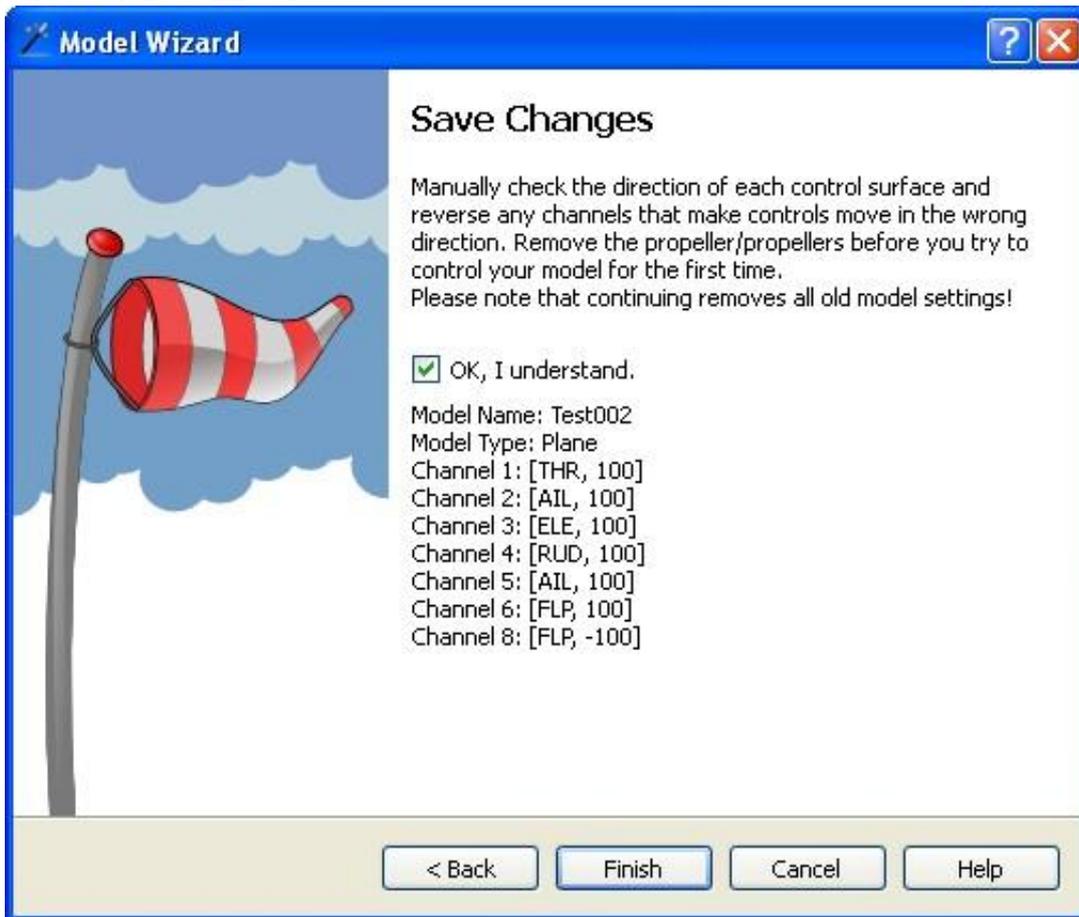




openTx für Taranis Anleitung Deutsch



Und so sieht dann vorab aus:



In den Inputs und in den Mischer werden dann diese Werte eingetragen:

Input	Source	Weight
Thr	Source (Thr)	Weight (100%)
Ail	Source (Ail)	Weight (100%)
Ele	Source (Ele)	Weight (100%)
Rud	Source (Rud)	Weight (100%)
Input05		
Input06		
Input07		
Input08		

Channel	Mixer	Weight	Switch
CH1	[I1] Thr	Weight (+100%)	
CH2	[I2] Ail	Weight (+100%)	
CH3	[I3] Ele	Weight (+100%)	
CH4	[I4] Rud	Weight (+100%)	
CH5	[I2] Ail	Weight (+100%)	
CH6	MAX	Weight (-100%)	Switch (SA↑)
		Weight (+100%)	Switch (SA↓)
CH7			
CH8	MAX	Weight (+100%)	Switch (SA↑)
		Weight (-100%)	Switch (SA↓)
CH9			

Edit Settings

Radio Profile: Application Settings | Simulator Settings

Profile Name: My Radio

Radio Type: Taranis

General Settings: EMPTY: No radio settings stored in profile

SD Structure path: C:/Programme/companion9x

Splash Screen: companion9x/Startbildschirm/Splash Taranis 21



Default Stick Mode: Mode 4 (AIL THR ELE RUD)

Channel Order: T A E R

Append version number to FW file name

Offer to write FW to Tx after download

Die Inputs- Geberreihenfolge kommt aus den Companion Grundeinstellungen hier Mode 4
 Die Mischer- und Kanalreihenfolge aus den Companion Kanalvoreinstellungen hier TAER

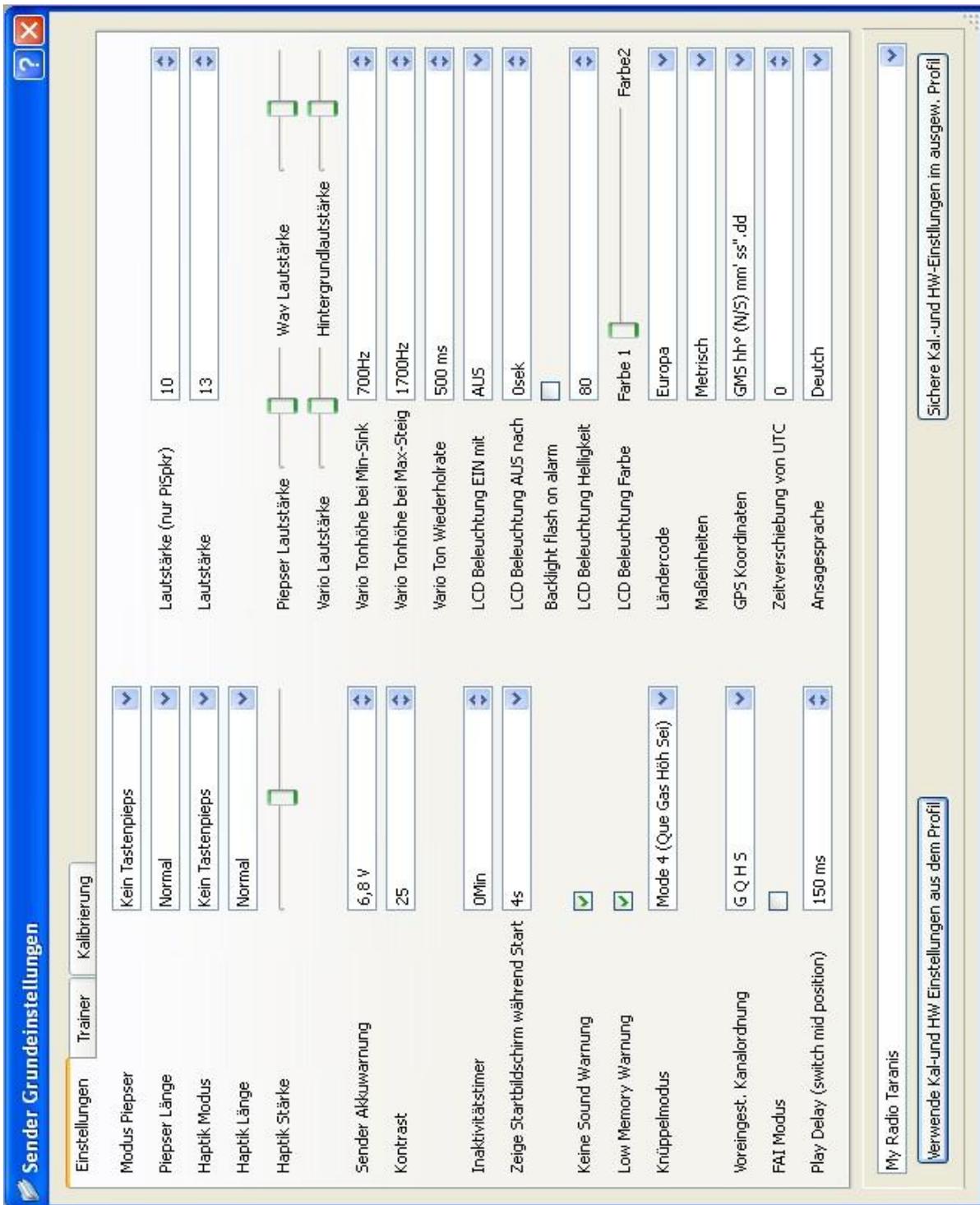
Das Ergebnis muss aber immer angepasst werden!

In den Gewichtungen und in den Servo-Drehrichtungen, das kann nie passen!
 Keine Automatik weiß wie das Servo eingebaut ist, wie das Ruderhorn angelenkt wird
 ob das Servo bei positivem Impuls rechts oder links läuft und wie sich das Ruder tatsächlich bewegt.

Es gibt aber „3 goldene Regel“ die das Mischer-Programmieren sehr erleichtert

- Positive Gebersignale müssen ein Ruder nach oben oder nach rechts bewegen.
- Das rechte Querruder ist das erste, ist positiv und geht nach oben.
- Zuerst müssen sich alle Mischerzeilen in der Simulation mathematisch richtig bewegen, erst dann wird am und mit dem Modell Kanal für Kanal einmalig per Servoumkehr Laufrichtungen und Servowege so angepasst, dass es „richtig“ läuft. Nicht vorher!

Sender Grundeinstellungen für alle gemeinsame Einstellungen



Hier kann man sich auch die Kalibrierwerte aus dem Sender ins Profil abspeichern, dann braucht man nicht jedesmal die Knüppel und den Akku abgleichen.

Aber Vorsicht, Werte prüfen! Vertrauen ist gut, Kontrolle ist besser.

Modelleinstellungen, beachte die überlange Seite unten für das ext. HF-Modul und Trianer

Modell 2 bearbeiten :MODEL02

Konfiguration: Heli | Flugphasen | Inputs | Mischer | Servos | Kurven | Logische Schalter | Spezial Funktionen | Telemetrie

Modell

MODEL02 | Modellbild

Timer 1: 00:00 | AUS | Count Down | Kein | Jede Minute | Not persistent | (00:00:00)

Timer 2: 00:00 | AUS | Count Down | Kein | Jede Minute | Not persistent | (00:00:00)

GasTimer Quelle: Gas | Gas Leerlauftrim | Gas Warnung | Gas Warnung | Vollgas hinten?

Trim Auflösung: Fein | Erw. Wege 100% --> 150% | Erw. Trim 25%--> 100% | Anzeige Checkliste

Zentrierpiepston: Sei Höh Gas Que S1 S2 S3 LS RS

Warnungen

Schalter Warnungen: SA SB SC SD SE SF SG

Poti Warnungen: AUS | S1 S2 S3 LS RS

Internes HF Modul

Protokoll: FrSky XJT - D16 | Start: CH1 | Kanäle: 8

Failsafe Mode: Halten letzte | Empfänger Nr.: 2

1	2	3	4	5	6	7	8
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
9	10	11	12	13	14	15	16
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

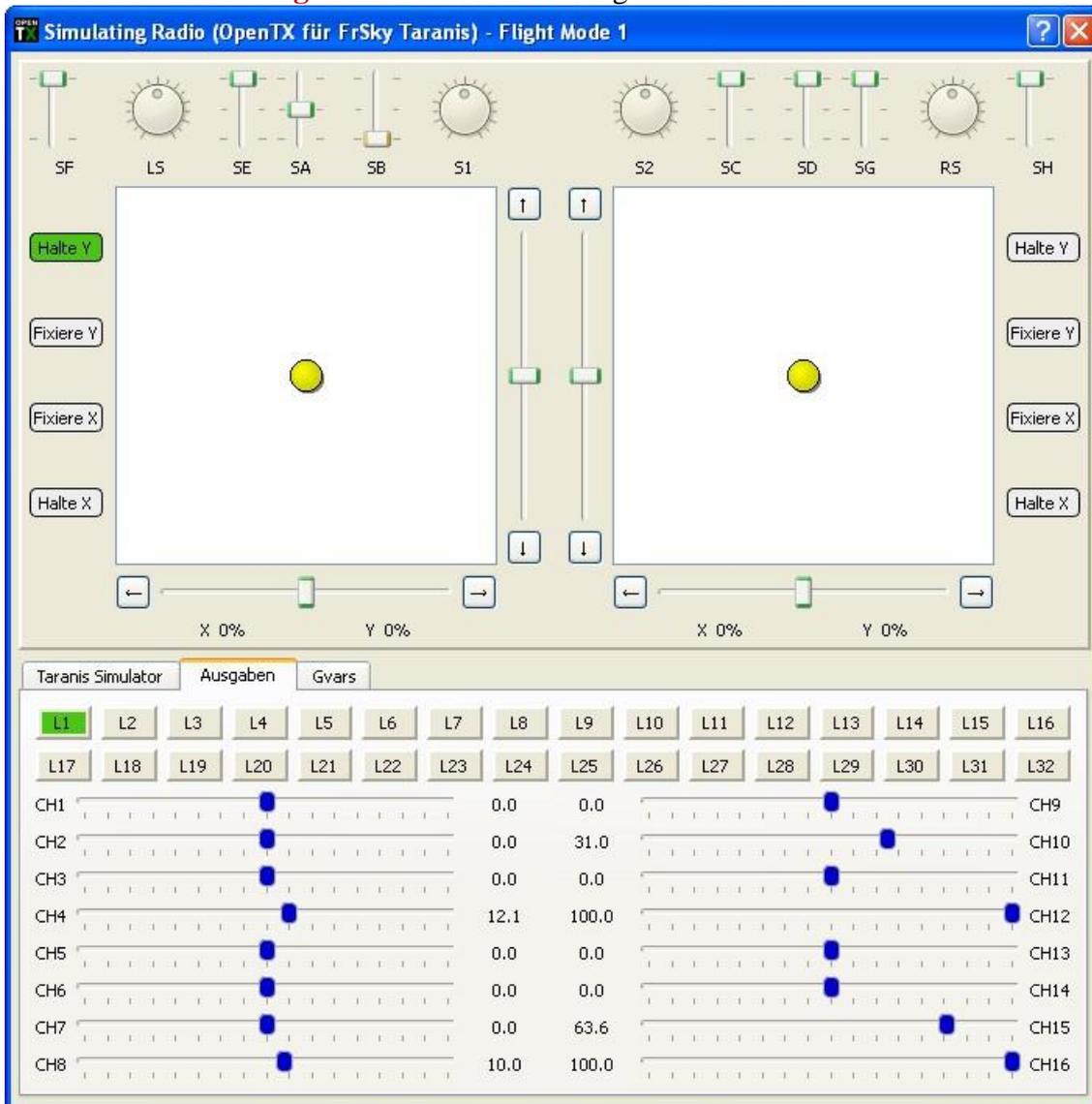
Externes HF Modul

Protokoll: OFF

Trainer Port

Simulation

Simulationsdarstellung mit allen Gebern und logischen Schaltern



Taranis Sender Simulation anstatt Simulation der Ausgaben

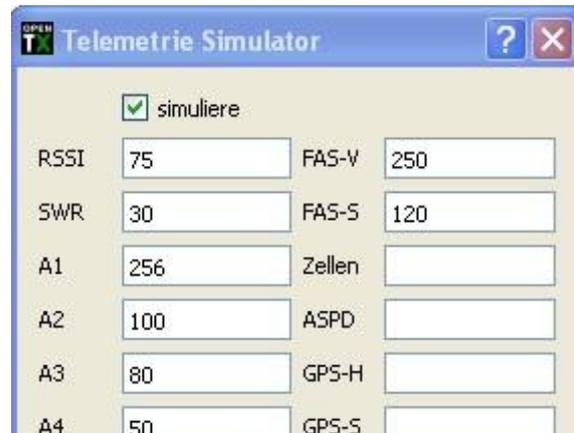


F4, F5, F6 Simulation von Telemetriewerten, Trainer und Debugger für LUA

Ab Companion V2.0.15 kann man mit Funktionstasten am PC weitere Werte simuliert werden.

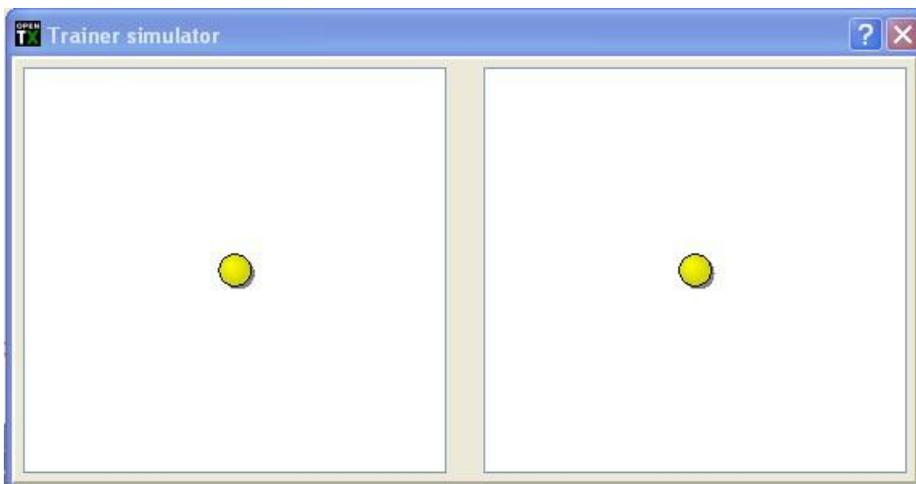
F4 Telemetriedaten simulieren

Um einfache, feste Telemetriewerte zu erzeugen, die dann in der Sendersimulation und Programmierung ausgewertet werden. (Funktionen werden ständig ausgebaut)

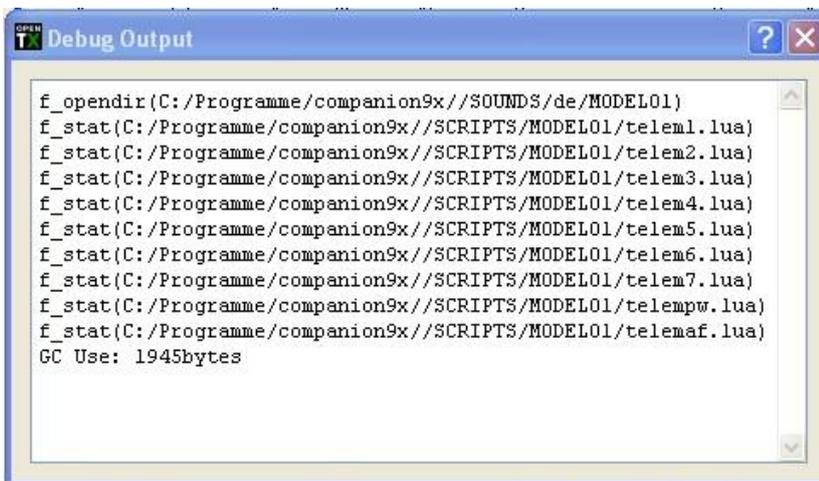


F5 Trainer simulieren

Damit kann eine einfache 4-Kanal Fernsteuerung simuliert werden. Die Werte kann man im Trainereingang oder als TR1-TR4 weiterverarbeiten.



F6 Debugmodus für Fehlermeldungen und LUA-Programmierung



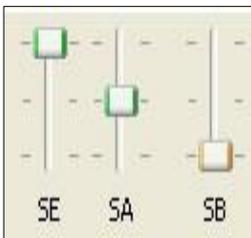
Meldungen des Debuggers erleichtern die LUA Programmierung und geben Fehlermeldungen aus

Bei LUA-Fehlern mal den SD-Karte Verzeichnispfad kontrollieren!

Übersicht der Globale Variablen (9x9 = 81) mit den jeweiligen Werten je Flugphase

Taranis Simulator	Ausgaben								
	Gvars								
	FM0	FM1	FM2	FM3	FM4	FM5	FM6	FM7	FM8
GV1	0	0	22	0	0	0	0	0	0
GV2	0	0	25	0	0	0	0	0	0
GV3	9	0	-24	0	0	0	0	0	0
GV4	-7	0	-24	0	0	0	0	0	0
GV5	0	0	0	0	0	0	0	0	0
GV6	0	0	0	0	0	0	0	0	0
GV7	0	0	0	0	0	0	0	0	0
GV8	0	0	0	0	0	0	0	0	0
GV9	0	0	0	0	0	0	0	0	0

Vergleich Schalterstellungen in Companion, PC-Simulation und am Sender



SE↑ Down- Stellung, Schalter am Sender zeigt von mir weg, nach unten

SA— die Mittelstellung ist klar

SB↓ Up- Stellung, Schalter am Sender zeigt zu mir her, nach oben



Die Telemetrie Seite Einstellungen und Darstellung der aller Werte (bis openTx V2.0.17)

The screenshot displays the 'Telemetrie' configuration window in openTx. The window title is 'Modell 2 bearbeiten :MODEL02'. The left sidebar contains tabs for 'Konfiguration', 'Heli', 'Flugphasen', 'Inputs', 'Mischer', 'Servos', 'Kurven', 'Logische Schalter', 'Spezial Funktionen', and 'Telemetrie'. The main area is divided into several sections:

- Protokoll:** A1, A2, A3, A4, RSSI, Höhenanzeige.
- A1:** Einheit (Volt(V)), Bereich (0,0), Voralarm (0,0), Kritischer Alarm (0,0).
- A2:** Einheit (Volt(V)), Bereich (0,0), Voralarm (0,0), Kritischer Alarm (0,0).
- A3:** Einheit (Volt(V)), Bereich (0,0), Voralarm (0,0), Kritischer Alarm (0,0).
- A4:** Einheit (Volt(V)), Bereich (0,0), Voralarm (0,0), Kritischer Alarm (0,0).
- RSSI:** Vario Quelle (45), Vario Grenzen (42).
- Höhenanzeige:** VSpeed (-10), Sink Max (-0,5), Sink Min (-0,5), Steig (0,5), MinSteig (0,5), MaxSteig (10).
- Verschiedenes:** Höhenanzeige in der Infozeile (unchecked), Serielles Protokoll (Frisky Sensor Hub), FAS Offset (0,0A), Spg Quelle (A1), Rotorblätter (2), mAh zählen (checked), Strom Quelle (---), mAh (0mAh), Speichern der mAh (unchecked).

At the bottom, there are three tabs: 'Telemetrie Bild 1', 'Telemetrie Bild 2', and 'Telemetrie Bild 3'. Below them is a dropdown menu 'Telemetrie Anzeige als Werte' and four empty input fields.

Die Telemetrie Flugdaten auswerten mit dem Daten-Logger

Alle Flugdaten können auf der SD-Karte, im Verzeichnis /LOGS, aufgezeichnet werden.

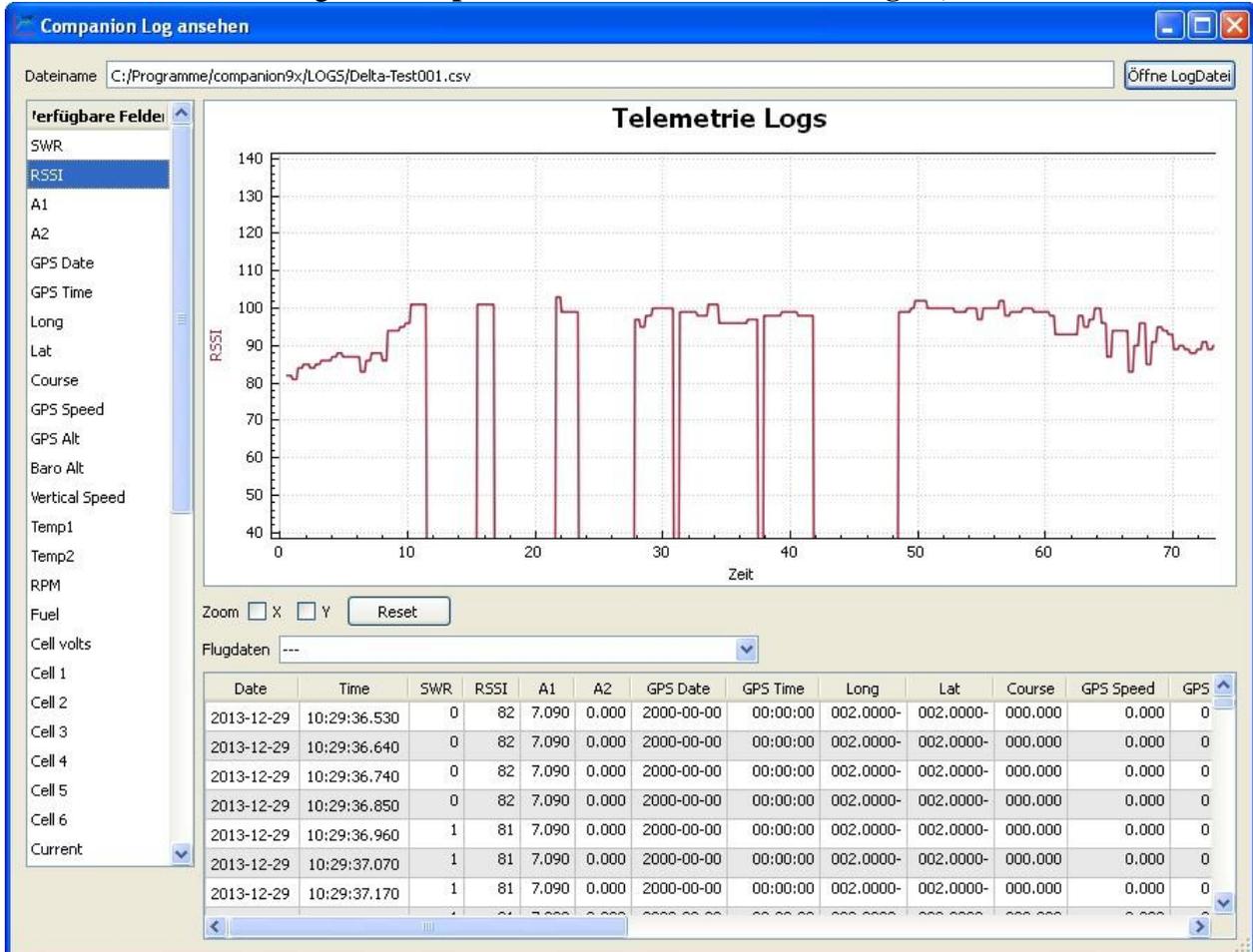
Dann die Datei in Companion übertragen und auswerten.

Aufgezeichnet werden **alle** Daten die aus der Telemetrie ankommen.

Selbst wenn kein Sensor angeschlossen ist, werden A1 und RSSI, SWR, Sender Akku gespeichert.

Die Aufzeichnungsrate / Takt kann eingestellt werden von 0,1s bis 10s

Starten der Aufzeichnung, in den **Spezial Funktionen mit Start Log 0,1s**



Es wird dabei eine Excel kompatible CSV-Datei erzeugt.

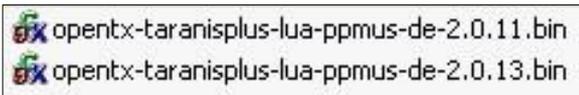
Der neue Bootloader ab openTx V2.00

Der neue Bootloader wird bei einem Senderupdate auf opentx V2.00 automatisch installiert!

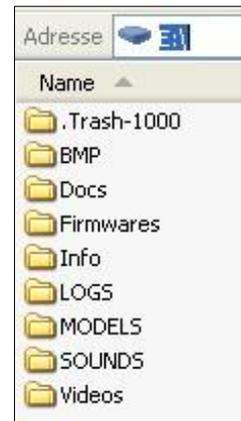
d.h. Um ein Senderupdate zu machen braucht man dann keinen DFU- und Zadig-Treiber mehr!
Die Sender-Update-Datei wird dann einfach auf die SD-Karte kopiert,
(hier ins Laufwerk E:\) ins **Unterverzeichnis \FIRMWARES**
Es können mehrere Versionen der Sendersoftware auf der SD-Karte gespeichert sein.

Wer eine Taranis Plus hat oder schon openTx V2.00 drauf hat braucht nichts ändern.

Hinweis:



opentx-taranisplus-lua-ppmus-de-2.0.11.bin
opentx-taranisplus-lua-ppmus-de-2.0.13.bin



Beim Download der Sendersoftware aus CompanionTx ist der Name sehr lang und enthält alle Optionen.

Auf der SD-Karte und am LCD-Display sind aber nur ca 16 Zeichen lesbar. Deshalb den langen Dateinamen umbenennen und kürzen z.B. OpenTx_V2013 bevor man ihn auf die SD-Karte schreibt.

Der neue Ablauf ist jetzt so:

Sender ist ausgeschaltet, kein USB-Kabel ist gesteckt!

Die beiden unteren Trimmrasten der Knüppel zusammen nach innen drücken und halten
Taranis einschalten, er fährt hoch und es meldet sich der Bootloader mit 2 Optionen

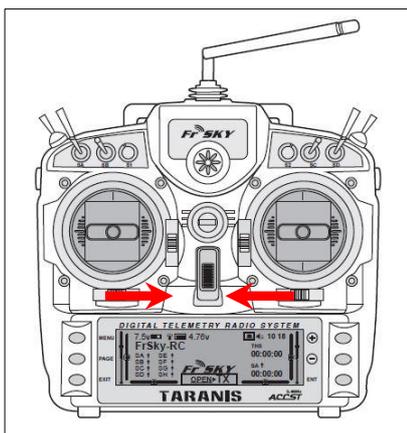
a) Firmware-Update

Auf der SD-Karte die Firmware auswählen und bestätigen.

Flashen startet und nach wenigen Sekunden ist die neue Firmware auf dem Sender. Fertig!

b) USB-Verbindung

Erst jetzt das USB-Kabel anstecken, damit wird die SD-Karte mit allen Unterverzeichnissen für den PC sichtbar und es kann darauf zugegriffen werden um Modelle oder ein Firmwareupdate zu übertragen.



Welche Version man auf dem Sender hat kann man einfach prüfen. Sender Grundeinstellungen (Menü Long, dann mit Page auf 4/8)



Hintergrundwissen: Es gibt 2 Bootloader!

1. Der Prozessor hat einen fest installierten internen STM32-Bootloader den wir bisher mit DFU-Util und dem Zadig Treiber angesprochen haben. **Das gibt es weiterhin wie gehabt.**

Sender ausgeschaltet lassen, USB Kabel anstecken,
Ablauf dann wie gewohnt, Firmware via PC an Sender übertragen.

2. Dann hat jetzt die Firmware openTx selbst einen Software-Bootloader installiert, der dann aufgerufen wird wenn wird die 2 Trimmasten halten und dann erst denSender einschalten.

Er greift dann für das Firmwareupdate direkt auf die SD-Karte zu,
Vorteil: keine Zadigtreiber mehr nötig

Wer noch keine openTx drauf hat muss sich das einmalig selber updaten

→ Dazu gibt es extra eine exakte Anleitung!

Varianten des USB Verbindungsaufbau mit Taranis X9D, X9E

1. **Taranis ausgeschaltet** => USB anschließen => STM32 Bootloaderfunktion aktiv (DFU, Zadig)

2. **Taranis eingeschaltet** => USB anschließen => Joystickfunktion aktiv

3. **Taranis Trims halten, einschalten** => USB anschließen => SD-Datenträger, Update, Boot

Neues Verfahren für das OpenTX Firmwareupdate auf Taranis ab OpenTx V2.00

Mit openTx 2.0 kommt **zusätzlich** ein neues Verfahren um den Sender mit einer neuen Firmware zu flashen. Dann ist kein spezieller Treiber (DFU-util, Zadig-Treiber) mehr nötig!
(Das bisherige, direkte Flashen des Senders mit DFU-util gibt es weiterhin!)

Dazu sind aber am Sender noch ein paar Vorarbeiten nötig:

CompanionTX starten, Taranis einschalten, hochlaufen lassen und dann per USB mit PC verbinden. Es melden sich 2 Wechseldatenlaufwerke E: und F: oder F: und G: Ein Laufwerk davon ist die SD-Karte

Auf der SD-Karte ein neues Unterverzeichnis einrichten mit **\Firmwares**

USB abmelden und abziehen, Taranis ausschalten

Jetzt noch einmal, **letztmalig**, die neue Bootloader-Firmware **wie bisher** mit DFU- und installiertem Zadig-Treiber von companion9x auf den Sender flashen und starten.

Ab jetzt wird es sehr einfach!

Ein neues Firmware-update wird nur noch auf die SD-Karte in das Verzeichnis \Firmwares kopiert. Von dort holt sich dann die Taranis das Update und flasht es selbstständig.

Der neue Ablauf ist jetzt so:

Sender ist ausgeschaltet, kein USB-Kabel ist gesteckt!

Die beiden unteren Trimmrädchen der Knüppel zusammen nach innen drücken und halten
Taranis einschalten, er fährt hoch und es meldet sich der Bootloader mit 2 Optionen

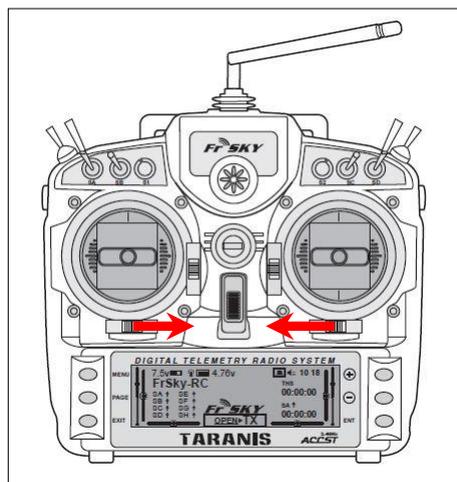
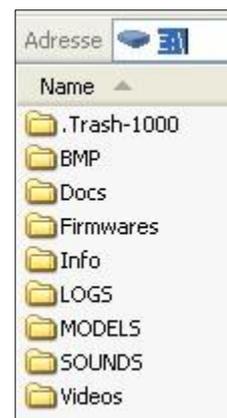
a) Firmware-Update

Auf der SD-Karte die Firmware auswählen und bestätigen.

Flashen startet und nach wenigen Sekunden ist die neue Firmware auf dem Sender. Fertig!

b) USB-Verbindung

Erst jetzt das USB-Kabel anstecken, damit wird die SD-Karte mit allen Unterverzeichnissen für den PC sichtbar und es kann darauf zugegriffen werden um Modelle oder ein Firmwareupdate zu übertragen.



Den Bootloader selbst updaten:

Gelegentlich kann es vorkommen, dass man den Bootlaoder selbst auch updaten muss, wenn er z.B. neue Funktionen erhalten hat.

Auch das geht ganz einfach wenn man schon openTx V2.x drauf hat und ohne dass man einen Zadig-Treiber braucht.

Der Bootloader ist immer mit dabei und Teil der Firmwaredatei opentxV2??.bin

Wenn man nur die Firmware updatet wird der Bootloader selbst nicht überschrieben!

Bootloader updaten:

Sender einschalten, hochfahren,
Sender Grundeinstellungen,
auf die SD-Karte2/8 gehen,
dann zu FIRMWARES
dort die aktuelle Firmware auswählen und ENTER drücken,
dann kommt ein Auswahlmenü

```
SD CARD 2/8
opentx-9xrpro-203.bin
opentx Flash BootLoader
opentx Delete
opentx-haptic-10a-200.bin
opentx-haptic-lua-FFMus200.bin
opentx1994_3005.bin
opentx207.bin
```

mit ENTER starten, fertig.

Hinweis: Option Joystick in der Firmware aktiviert / deaktiviert

- Ohne Option Joystick, kann man direkt per USB Kabel verbinden ohne vorher im Sender in den Bootloadermodus zu gehen und es melden sich die 2 Laufwerke.
- Mit Option Joystick, **muss** im Sender der Bootloadermodus aktivert werden, dann erst per USB verbinden und es melden sich die 2 Laufwerke.

Varianten des USB Verbindungsaufbau mit Taranis X9D, X9E

1. **Taranis ausgeschaltet** => USB anschließen => STM32 Bootloaderfunktion aktiv (DFU, Zadig)
2. **Taranis eingeschaltet** => USB anschließen => Joystickfunktion aktiv
3. **Taranis Trims halten, einschalten** => USB anschließen => SD-Datenträger, Update, Boot

Hilfe: Falsche OpenTx Software geflasht, LCD dunkel, nichts geht mehr, was tun?

Es kann mal vorkommen dass man die falsche Software auf die Taranis flasht und nichts geht mehr (Opentx für Taranis anstatt für Taranis Plus).

Keine Panik das kann man ganz einfach lösen.

Wenn man eine Taranis oder Taranis Plus mit mindestens OpenTx Version V2.00 hat, geht das ganz einfach, da dort ein eigener Bootloader vorhanden ist, den man nicht zerstört hat. Allerdings muss man jetzt den Bootloader blind bedienen, da man nichts mehr am LCD sieht. Zum Blind flashen darf nur eine einzige Firmware-Datei auf der SD-Karte, Verzeichnis Firmwares, drauf sein, weil man ja nicht sieht was als erste Datei am Display zur Auswahl angezeigt wird.

Oder: Wie bisher mit vorinstalliertem Zadig-Treiber und DFU-Util diese neue Version flashen.

Richtige Software downloaden:

Companion starten. Ein neues Senderprofil für Taranis oder TaranisPlus einrichten und die Optionen zusammenstellen.

Richtige Software downloaden, diese umbenennen auf max 16-20 Zeichen, z.B. OpentxPlusV2012 und dann auch nur diese opentx-Software auf die SD-Karte /FIRMWARES des Senders laden.

Bootloader blind bedienen:

Den Affengriff machen, d.h. die beiden unteren Trimmastasten zusammendrücken, Sender einschalten und erst dann die Trimmastasten nach 1 sec loslassen.

Am LCD-Display steht, was du nicht siehst: Bootloader Vx.xx und Write Firmware

Jetzt Sender per USB mit PC verbinden, es melden sich die SD-Karte mit allen Verzeichnissen.

Am LCD-Display steht, was du nicht siehst: USB Connected

1. von der SD-Karte alle Dateien unter /FIRMWARES erst mal auf PC sichern.
 2. Auf der SD-Karte unter /FIRMWARES alle Dateien löschen und nur die neue richtige Software opentx Vx.xx draufkopieren.
 3. USB abziehen
- Am LCD-Display steht, was du nicht siehst: Bootloader Vx.xx und Write Firmware

Tip: Falls man gar keine Verbindung vom Sender zum PC per USB mehr herstellen kann, dann die SD-Karte aus dem Sender entnehmen, am PC wie oben beschrieben die SD-Karte vorbereiten, dann wieder in den Sender rein. Die zwei Trimmastasten halten und Sender einschalten. (Am LCD-Display steht, was du nicht siehst: Bootloader Vx.xx und Write Firmware)

Jetzt blind genau so vorgehen:

Enter kurz drücken, 1 sec warten, Enter länger drücken
10-15 sec warten, das sollte zum flashen reichen

Exit kurz drücken, 1 sec warten, Exit länger drücken, ca 1 sec halten
Das wars, der Sender sollte sich jetzt normal melden und hochfahren.
Eventl doch noch Sender ausschalten. Sender einschalten

Merkhilfe Kanal und Schalterbelegung

Model _____

Channels		Notes
1	9	
2	10	
3	11	
4	12	
5	13	
6	14	
7	15	
8	16	
Receiver _____		

11/2013 by VoBo

Firmwareupdate der Sensoren, Empfänger, HF-Module

Das hat nichts mit einem Update von openTx zu tun!

OpenTx ist nur die Bedien-und Programmieroberfläche für den Sender.

Hier geht es um updates der S-Port Sensoren, der Empfänger und der eigentlichen HF-Sendersoftware im internen oder externen XJT-Module

Das kann man mit dem PC machen, oder im Maintenance-Mode auch direkt alles vom Sender aus.

Variante 1: Mit dem PC die Smart-Port Sensoren, Empfänger, HF-Module updaten

Vorab, Gefahr, Ärger! Nicht einfach den FrSky FrUSB-3 Adapter einstecken, ohne dass vorher der Treiber geladen wurde!

Die Vorarbeiten:

Für das Firmwareupdate der Smart-Port Sensoren, Empfänger und HF-Module mit dem PC brauchen wir etwas Hardware, Software und das alles in der richtigen Reihenfolge!

Als Hardware den FrSky FrUSB-3 Adapter

Eine Dioden-Anpassung für die serielle Schnittstelle RX und TX (kaufen oder selber machen)

Den Silab-Treiber. Der muss zuallererst am PC installiert werden, noch bevor der USB Adapter eingesteckt wird! (obwohl Windoff motzt und sagt der sei nicht zertifiziert usw.)

Dann erst den USB Adapter reinstecken, der wird jetzt sofort richtig erkannt und ein Com-Port zugewiesen.

Nun im Gerätemanager nachschauen welcher freie Com-Port Nummer der Silab-Treiber erhalten hat (bei mir ist es eben gerade COM14)

Das eigentliche Update in genau dieser Reihenfolge machen

Das Frsky Smart-Port Update Programm starten und den Com-Port eintragen

Die eigentliche Firmware für den Sensor zum Update laden, Filename *.frk

Wer hier die falsche *.frk Datei verwendet kann sich die die Geräte abschießen!

Das Programm sucht jetzt den Sensor

Jetzt sofort den Sensor richtig einstecken, dann wird er auch gleich gefunden!

Achtung: Zwischen Programm Start zum Sensor suchen und dem Anstecken hat man nur 3-4 sec Zeit, sonst wird der Sensor nicht gefunden. Da muss man schnell sein.

Dann den Dowload starten.

Das dauert überraschend lange, so 20-30s, dann ist es fertig.

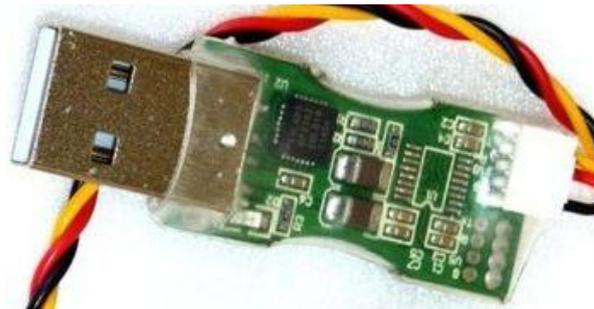
Und wir können den Sensor abstecken.

Link zu Frsky für diese Programme, Treiber, Firmware, Tools, Sourcecode, Manuals:

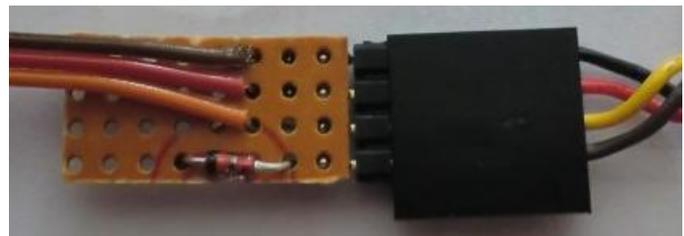
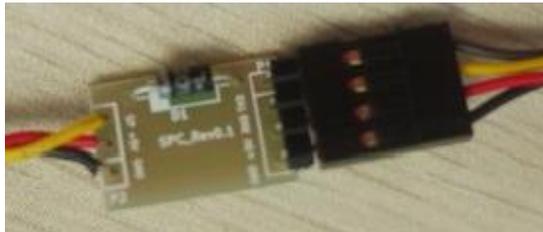
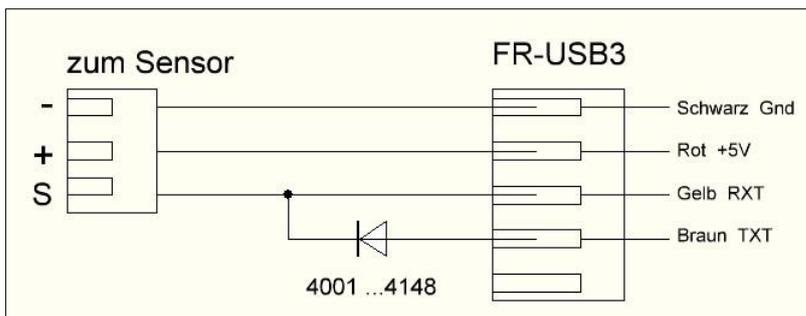
<http://www.frsky-rc.com/download/>

Das folgende Beispiel ist für ein update des Vario-Sensors

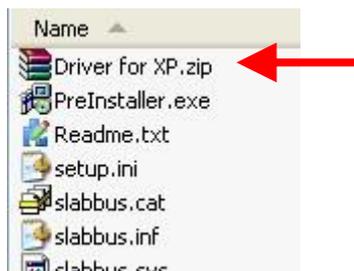
openTx für Taranis Anleitung Deutsch
FrSky FrUSB-3 Adapter mit Silab USB Baustein



**Zum Update braucht man auch diese Diodenanpassung:
Mit Diode 1N4001 zum Selbermachen auf Lochraster, oder fertig kaufen SPC**



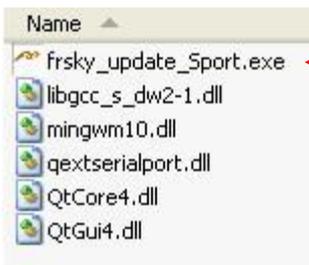
**Silab-Treiber installieren
Pre-Installer.exe**



**dann erst Fr-USB3 einstecken
Damit wird der Adapter erkannt und ein
Com-Port zugewiesen
Silicon Labs (bei mir ist das COM14)**

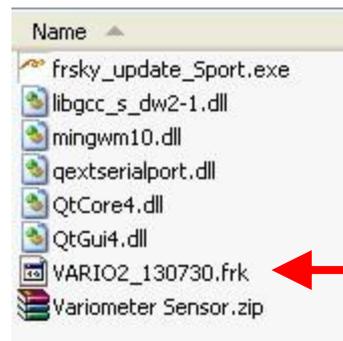


Das UpdateProgramm für S-Port Sensoren und HF-Module frsky_update-Sport.exe



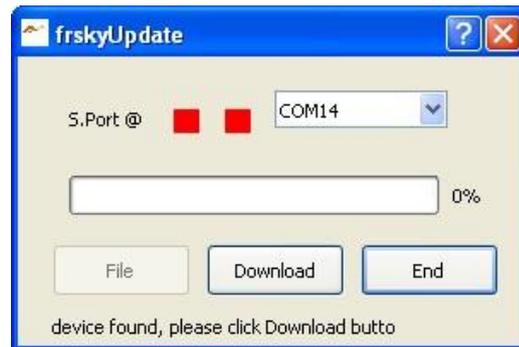
Windoff weist dann einen virtuellen Com-Port zu (bei mir COM14), den muss man sich im Gerätemanager raussuchen! Siehe: Systemsteuerung, System, Hardware, Gerätemanager, Anschlüsse COM

Programm starten, Com-Port eintragen und das neue update File *.frk laden
Wer hier die falsche *.frk Datei verwendet kann sich die die Geräte abschießen!



Jetzt sofort den Sensor anstecken, Dazu hat man nur ca 4sec Zeit!

damit wird er auch gleich gefunden



Update starten, Sensor blinkt dabei ganz langsam und wir sind fertig



Mit dem PC die X-Empfänger und das externe HF-Modul updaten (ETSI V1.8.1)

Alle bisher in der EU verkauften 2,4GHz RC-Komponenten, Sender und Empfänger, hatten ein HF-Übertragungsverfahren das nach ETSI V1.7.1 arbeitet.

Seit 01.01.2015 dürfen in der EU nur noch RC-Sender und Empfänger mit ETSI EN 300 328 V1.8.1 verkauft werden.

Das gilt aber nur für die EU. Der Rest der Welt fliegt weiterhin mit ETSI V1.7.1

Die Systeme sind nicht kompatibel untereinander.

Ein neuer Sender nach ETSI V1.8.1 arbeitet nicht mit einem Empfänger nach ETSI V1.7.1 zusammen und umgekehrt und auch nicht mehr mit einem D-Empfänger.

Man wird also irgendwann seine Komponente auf den neuen Stand updaten müssen oder alles weiterhin mit dem alten Standard betreiben, oder einzelene neue Komponenten zurück auf V1.7.1 flashen. (auch das geht)

Man muss nicht updaten, Bestandsschutz ist gegeben.

Für das Update braucht man für die HF-Module und für die Empfänger eine neue Firmware (das hat nichts mit openTx zu tun) die man auf der Homepage von Frsky downloaden kann:
<http://www.frsky-rc.com/download/index.php?sort=How%20To>

Dort dann auf die Begriffe ETSI EN 300 328 V1.8.1 oder EU achten.

Es ist auch immer ein Manual dabei, das erklärt wie man das macht, was man für eine Software und welche Hardware man dazu braucht.

Das Prinzip ist einfach:

Das Update wird immer am S-Port-Stecker der HF-Module und der Empfänger durchgeführt.

Dazu brauchen wir die gleichen Hardware und Software wie sonst auch, wenn wir ein update an am S-Port der Telemetrie-Sensoren machen.

Siehe Beispiel: **Smart-Port Sensoren Firmware updaten**

Hardware:

Der Frsky **FrUSB-3 und der Diodenadapter** wird für alle Software-Update aller Smart-Port Sensoren und auch für das EU-Update EN3003 328 ETSI V1.8.1 aller HF-Module und Empfänger benötigt!

http://www.frsky-rc.com/product/pro.php?pro_id=37

Software für das Update:

Treiber für Win XP/7/8 und Software zum updaten

http://www.frsky-rc.com/download/index.php?sort=Tool&cate_id=0&pro_id=0

Die FrSky *.frk Dateien für HF-Module, Empfänger und Sensoren gibt es hier:

<http://www.frsky-rc.com/download/>

http://www.frsky-rc.com/download/index.php?sort=Firmware&cate_id=0&pro_id=0

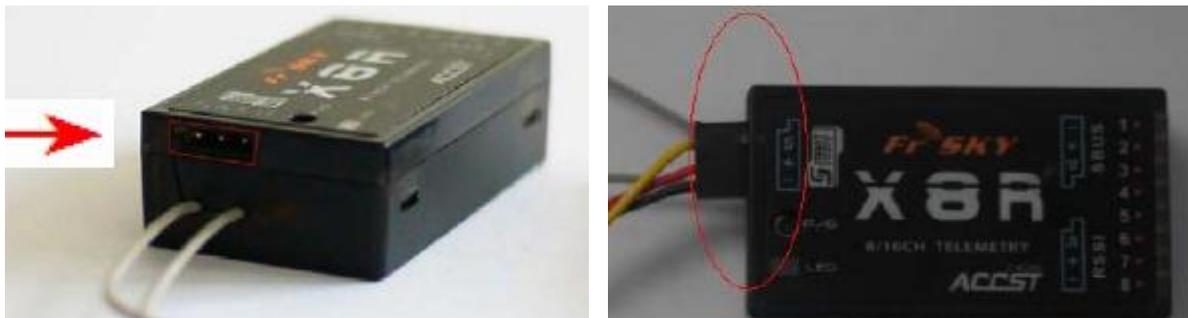
Wer hier die falsche *.frk Datei verwendet kann sich die die Geräte abschießen!

Von Frsky gibt es 3-4 Varianten für die HF-Software ETSI V1.7.1 oder ETSI V1.8.1

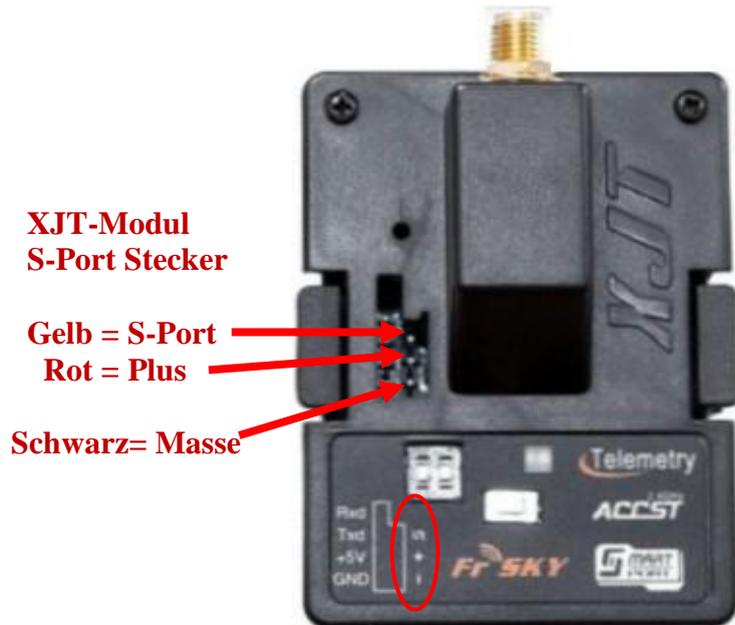
- Weltweit als Non EU Versionen, mit ETSI V1.71 für alle X-Empfänger, alle D-Empfänger
- 3 EU-Versionen: Nur am Versions-Datum zu unterscheiden 22./ 23.01.2015 und 13.04.2015
- EU-Version V1.8.1 X-Empfänger mit V1.8.1, D-Empfänger mit V1.7.1 Version 150123
- EU-Version V1.8.1 X-Empfänger, keine D-Empfänger mehr Version 150413
- EU-Version V1.8.1 X-Empfänger, keine D-Empf. S-Bus Fehler behoben Version Juli 2015

Wer seine D-Empfänger weiterhin nutzen will, kann die passende Software auf Sender HF-Modul und Empfänger flashen.

Steckerbelegung Servokabel für Upgrade am S-Port Stecker des Empfängers



S-Port Stecker am externen HF-Modul Firmware für Firmwareupdates



Mit dem PC das Sender-interne XJT HF-Module Updaten (Pins im Modulschacht)

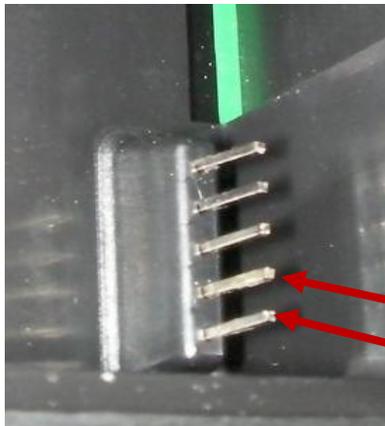
Variante 1:

Auch dazu benötigen wir wie vorher, den PC mit vorinstalliertem Silab-Treiber, das Programm frsky_update-Sport.exe und den FrUSB-3 Adapter mit Diodenanpassung.

Aber hier müssen wir den Plus abziehen und isolieren!

Nur Masse und das S-Port Signal anstecken.

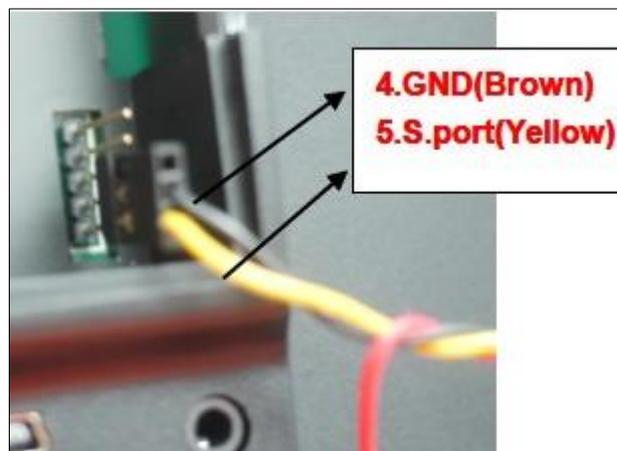
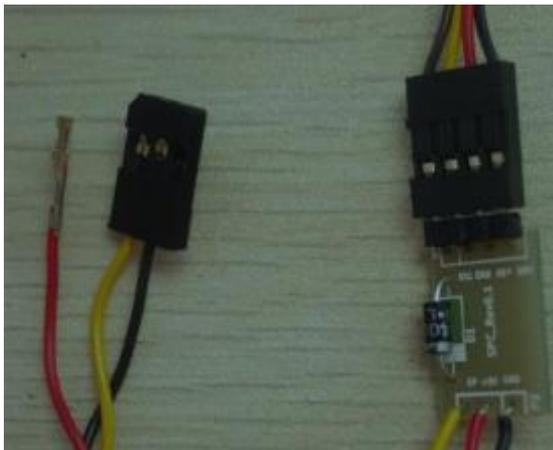
Im Modulschacht ist der S-Port Anschluß des internen XJT-Moduls als 5. Pin herausgeführt



4. Masse (Schwarz)
5. S-Port (Gelb)



Es darf nur Masse (Schwarz / Braun) und das S-Port Signal (Gelb / Weiß) eingesteckt werden. Auf keine Fall der Plus, also Rot am Stecker ziehen!



Progamiertablauf dann wie oben

Wer hier die falsche *.frk Datei verwendet kann sich die die Geräte abschießen!

~~Varainte 2: Mit dem Sender Sport-Sensoren, Empfänger, HF-Module updaten~~

~~Entwurf: (noch nicht klar wann/ob das so in openTx V2.10 integriert wird)~~

~~Ab V2.017 gibt es auch die Möglichkeit ein Firmware-update des internen HF-Moduls, des externen HF-Moduls, aller X-Empfänger und aller Telemetrie-Sensoren direkt vom Sender aus zu machen, ohne PC oder sonstiger Hardware und Programmieradapter.~~

~~Dazu gibt es den Maintenanc-Mode = Wartungsmode von Mike Blandford. Dabei wird der Bootloader um besondere S-Port Update- Funktionen erweitert.~~

~~Auf der SD-Karte braucht man ein zusätzliches Verzeichnis **FIRMWARE** Dort kommen nur die *.frk Dateien für die Updates der S-Port Gräte, der Empfänger und der XJT HF-Module rein~~

~~Die *.frk Dateien bekommt man hier unter: Download/Firmware~~

~~http://www.frsky-rc.com/download/index.php?sort=Firmware&cate_id=0&pro_id=0~~

~~Nicht verwechseln mit **FIRMWARES** dort kommen nur die openTx???.bin Dateien rein, für das update von OpenTx.~~

~~Zuerst brauchen dir die passende Software für die Taranis oder die TaranisPlus. Der Maintenance Mode ist nicht in der normalen openTx V2.017 enthalten, sonder muss extra compilert werden. Hierzu gibt es 2 fertige deutsche Versionen. Im Beitrag #2 zum Download. <http://fpv-community.de/showthread.php?63626-V2-0-17-Maintenance-Mode-im-Bootlaoder-f%FCr-S-Port-update-direkt-vom-Sender-aus>~~

~~In dieser Zip-Datei sind 2Versionen von openTX V2.017 enthalten:~~

~~Für Taranis: openTxV2017T_DE_Maintenance.bin~~

~~Für TaranisPlus: openTxV2017TPlus_DE_Maintenance... .bin~~

~~Zuerst mal die richtige Datei auswählen und umbenennen, wg kurzer Namen~~

~~Taranis: V2017T_DE_SPortupdate.bin~~

~~TaranisPlus: V2017TPlus_DE_SPortupdate.bin~~

~~Die richtige Datei auf die SD-Karte kopieren, ins Verzeichnis **FIRMWARES** kopieren.~~

~~Dann per normaler Bootloader-Funktion diese neue angepasste openTx V2.017 Firmware flashen.~~

~~Dann den Bootloader selbst updaten (siehe Handbuch Seite 327).~~

~~Damit erst wird der Bootloader um den Maintenance Mode erweitert!~~

~~**Solange der Bootloader nicht mehr geändert wird bleibt diese erweiterte Funktion erhalten!**~~

Jetzt haben wir einen erweiterten Bootloader

1. OpenTx Update per Bootloader-Funktion (wie bisher)

Beide Trimmaster nach innen drücken, halten, dann einschalten

Da greift der Bootloader auf **FIRMWARES** zu, für Updates der OpenTX-Firmware

2. S-Port-Update per Bootloader-Funktion:

Beide Trimmaster nach außen drücken, halten, dann einschalten

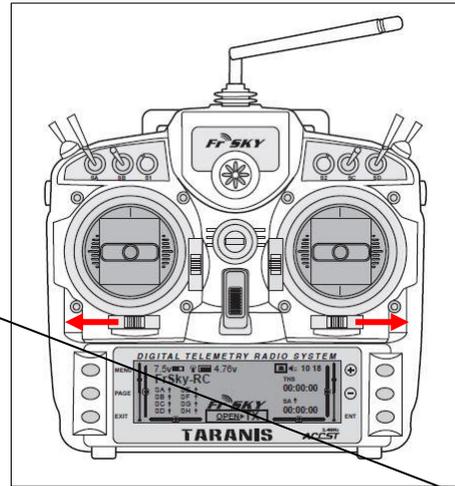
Da greift der Bootloader auf **FIRMWARE** zu. Für Updates von allen Telemetriesensoren, X-Empfängern, internes und externes XJT-Modul, direkt vom Sender aus, am S-Port-Pin im Modulschacht, ohne extra Hardware. Die Tastenbelegung ist da aber etwas anders.

Wartungsmodus aufrufen:

Die beide unteren Trimmraster nach **außen** halten, dann erst den Sender einschalten.

Damit kommt man in den Wartungsmodus mit 3-5 Auswahlmöglichkeiten

- > Update Bootloader selbst
- > Update Internes XJT-Modul,
- > Update am S-Port Pin für
 - Externes XJT-Modul,
 - X-Empfänger
 - Telemetrie-Sensoren.



Varainte 2: Mit dem Sender S-Port-Sensoren, Empfänger, HF-Module updaten

Ab openTxV2.10

Das geht jetzt noch einfacher: **Keine extra Maintenance-Mode im Bootloader nötig!**

Einfach auf der SD-Karte ein neues Verzeichnis anlegen z.B: **/SPORT_Updates** oder ähnlich
Dort die benötigten *.frk Dateien für das Update von S-Port Geräten reinkopieren.

Dann am Sender dieses Verzeichns aufmachen und die richtige *.frk Datei auswählen.
OpenTx erkennt dass es *.frk Dateien sind und bringt nun eine Auswahl:

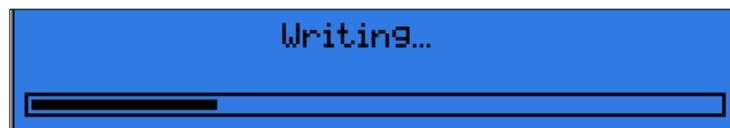
Flash externes SPORT-Gerät: für alle Sensoren, Empfänger, externes HF-Modul
oder

Flash internes XJT-Modul: für das interne XJT HF-Modul

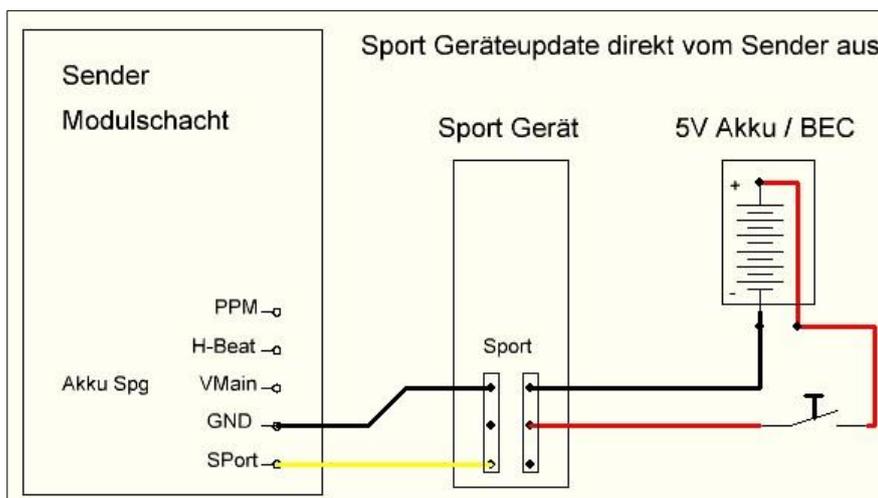
Und schon startet das SPORT-Update

Vorsicht, Gefahr, Aufpassen!

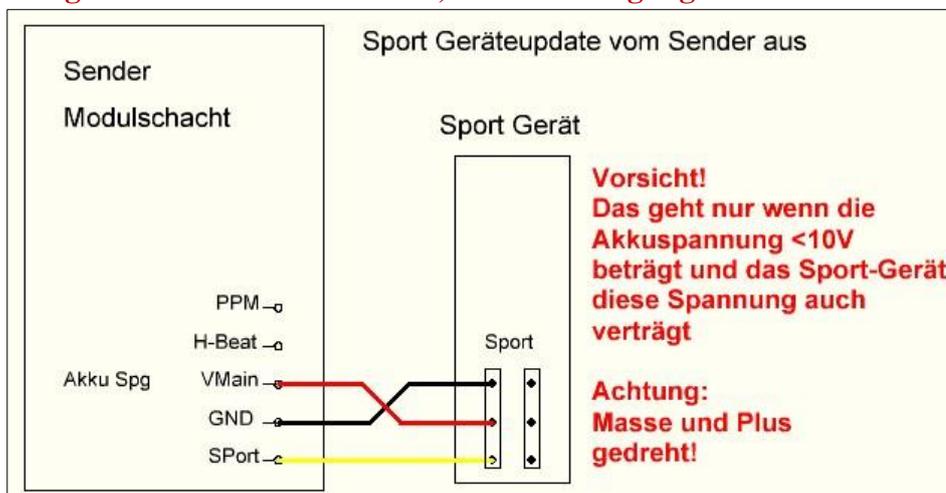
Immer die richtige *.frk Dateien verwenden, sonst schießt ihr euch die Geräte ab!
Exakt auf die Namen und Nummern achten!



Verkabelung: Sender an S-Port Geräte, Stromversorgung über extra 5V Akku und Schalter
Falls das S-Port Gerät an einer externer Stromversorgung hängt Ablauf exakt einhalten!
Im Sender richtige FW im SD-Kartenmenu aufrufen. Dann **Externes Gerät flashen** bestätigen.
ERST JETZT Sofort das S-Port Gerät einschalten. Das muss innerhalb von 2-3 sec erfolgen
damit das S-Port Gerät in den Bootmodus kommt. Dann wird die FW installiert.



Verkabelung: Sender an S-Port Geräte, Stromversorgung direkt vom Sender



Mit dem PC SPORT-ID Adresse ändern, Telemetrie Übertragungsrate anpassen

Alle Telemetriesensoren haben erst mal eine feste ID. Diese kann man ändern und neu vergeben, wenn man z.B. 2 gleiche Sensoren verwenden will. Auch die Übertragungsrate der Telemetriesensoren kann in Grenzen verändert werden.

Dazu braucht man die gleiche Hardware wie oben:

FrUSB-3/FUC-3 (FrSky Upgrade Cable) = USB-Interface mit Kabelsatz

SPC (FrSky Smart Port Converter) = Diodenadapter

Der Treiber muss vorher eingerichtet sein, bevor das USB-Interface zum Ersten mal angesteckt wird!

Dann das Frsky- Programm: FrSky_SPORT_Tool_.exe

Gibt es hier:

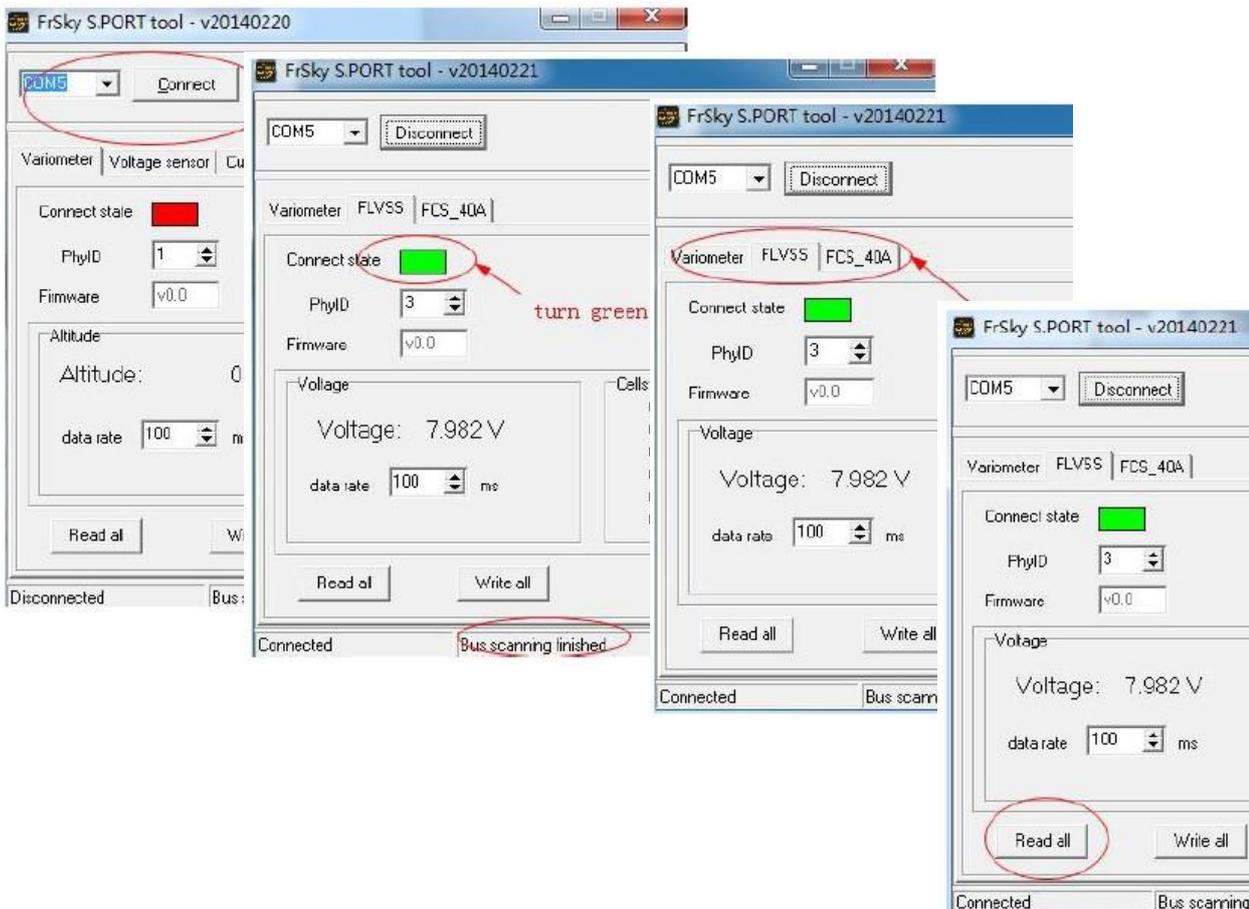
<http://www.frsky-rc.com/download/view.php?sort=Tool&down=143&file=S.Port%20Tool>

Man braucht keinen SCC = Servo Channel Changer!

Die Telemetriesensoren werden am S-Port angesteckt

Ansonsten ist der Ablauf gleich wie oben.

Programm starten, virtuellen COM-Port suchen und eintragen



Übersicht der Sensoren mit ID, Sub-ID, APP-ID und Periode

SENDER	PHY_ID	GROUP NUMBER	APP_ID	PERIOD (100ms)	ENALBE
VARIO2	0	0	Altitude	2	
			Altitude Rate	1	
FLVSS	1	0	Battery voltage	3	
FAS-40S	2	0	Battery voltage (Ampere sensor)	5	
			Current	5	
GPS	3	0	GPS Latitude/ Longitude	10	
			GPS Altitude	10	
			GPS Speed	5	
			GPS Course	5	DISABLE
			GPS TIME	100	
RPM	4	0			
SP2UART_H	5	0	SP2UART	2(10ms)	
			ADC3	5	DISABLE
			ADC4	5	DISABLE
SP2UART_R	6	1	SP2UART	2(10ms)	
			ADC3	5	DISABLE
			ADC4	5	DISABLE
FAS-150S	7	0	Current	5	

REMARK:

The group number of VARIO2 is forced to 0, although it can be changed to any other data.

The enable status of application ID is available for GPS, SP2UART and FAS-150S only at present, VARIO2, FAS-40S, and FLVSS are forced to setting ENABLE

Teil F Der FrSky Pultsender X9E

X9E Ein erster Überblick

Seit August 2015 gibt es den neuen Frskysender X9E als Pultsender.

Das ist eine erweiterte X9D Taranis mit zusätzlichen Funktionen in anderem Gehäuse.

Drehgeber als Eingabe rechts ersetzt die 3 Tasten **+ - Enter**
2 zusätzliche Schiebepotis in der Mitte
8 Schalter und 2 Poti kann man zusätzlich einbauen
Bluetooth BT4.0 ist eingebaut
Ein Farb LCD-Display oben (an der Sonne sieht man leider nichts!)
usw

Eingebaut ist ein Akku mit 8 Zellen NiMH 9,6V mit niedriger Selbstentladung „Eneloop“
Andere Akkus, Lipo, Life, mit 2 - 3 Zellen sind möglich, Spannungsbereich von 6V – 15V

Das mitgelieferte DC-Netzteil hat 18V 200mA
Die Ladeschaltung für 8 Zellen NiMH ist im Sender eingebaut

Die Echtzeituhr RCT hat eine Stützbatterie Typ CR1220 3V/ 35mAh (unter dem Sender-Akku)

Tips:

Falls sich die Echtzeituhr immer wieder resetet (und die Uhrenbatterie ok ist),
dann den Elko 100uF/16V gegen 470uF/16V tauschen.
Siehe Linke Seite, neben der kleinen Drehgeberplatine

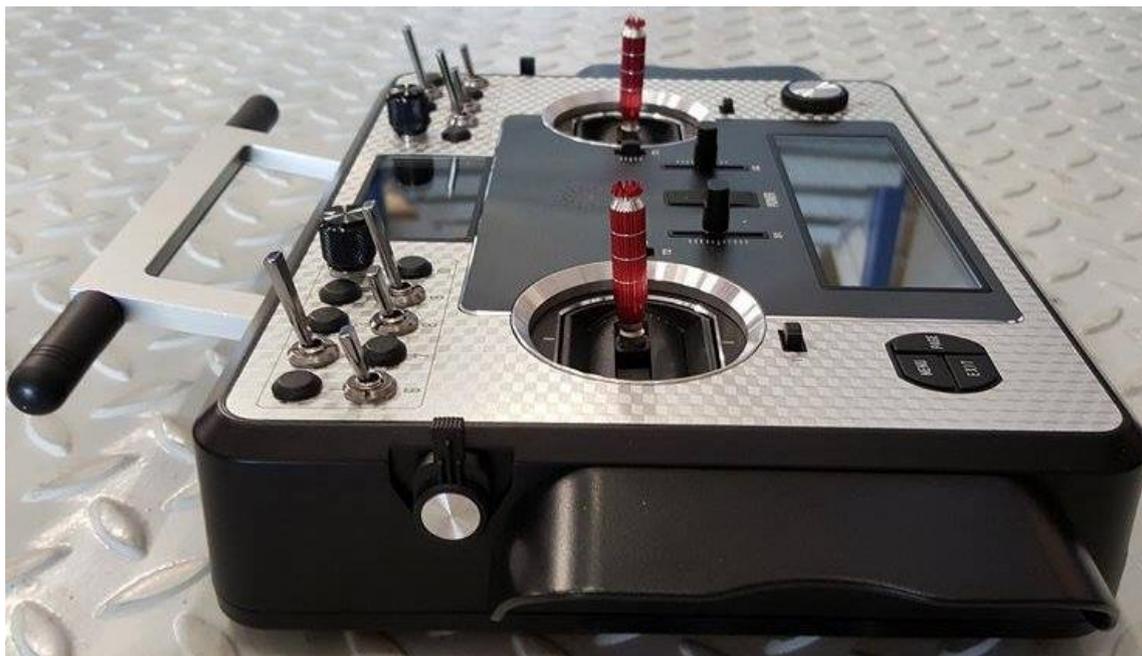
Die Schrauben für die Halterung des Senderakkus sind recht kurz,
man kann längere Schrauben von Servohalterungen verwenden.

Der Senderakku ist nur mit Doppelklebeband in der Halterung fixiert.
Kann sein er löst sich bei starker Hitze. Mit Tesaband umwickeln hilft.

**Um alle Funktionen des 9XE nutzen zu können braucht man
mindestens openTx V2.1x auf dem Sender und Companion V2.1x auf dem PC**

**openTx V2.1 unterscheidet sich vor allem in den Telemetriefunktionen,
der Rest sind normale Anpassungen und Erweiterungen**

X9E Ansichten und Einblicke

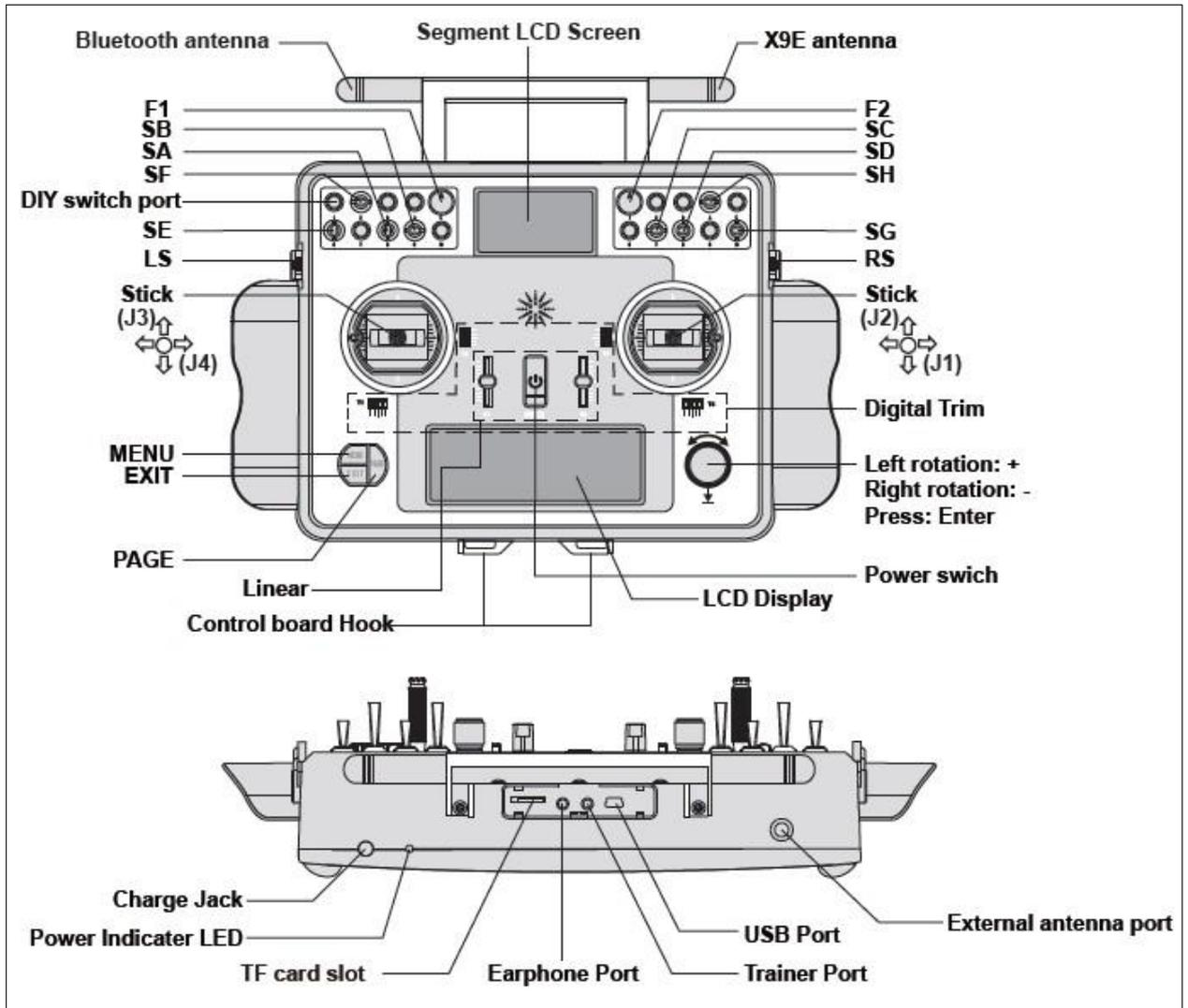


FrSky X9E Sender Stand August 2015 First Batch

openTx für Taranis Anleitung Deutsch
Frontseite SD-Kartenslot , Audio-Buchse, DSC-Buchse, USB-Anschluss



Senderübersicht der Bedienelemente



Man beachte:

am X9D sind S1, S2 die Potis oben

am X9E sind S1, S2 die Fader in der Mitte F1, F2 sind die Potis oben (anstatt S1, S2)

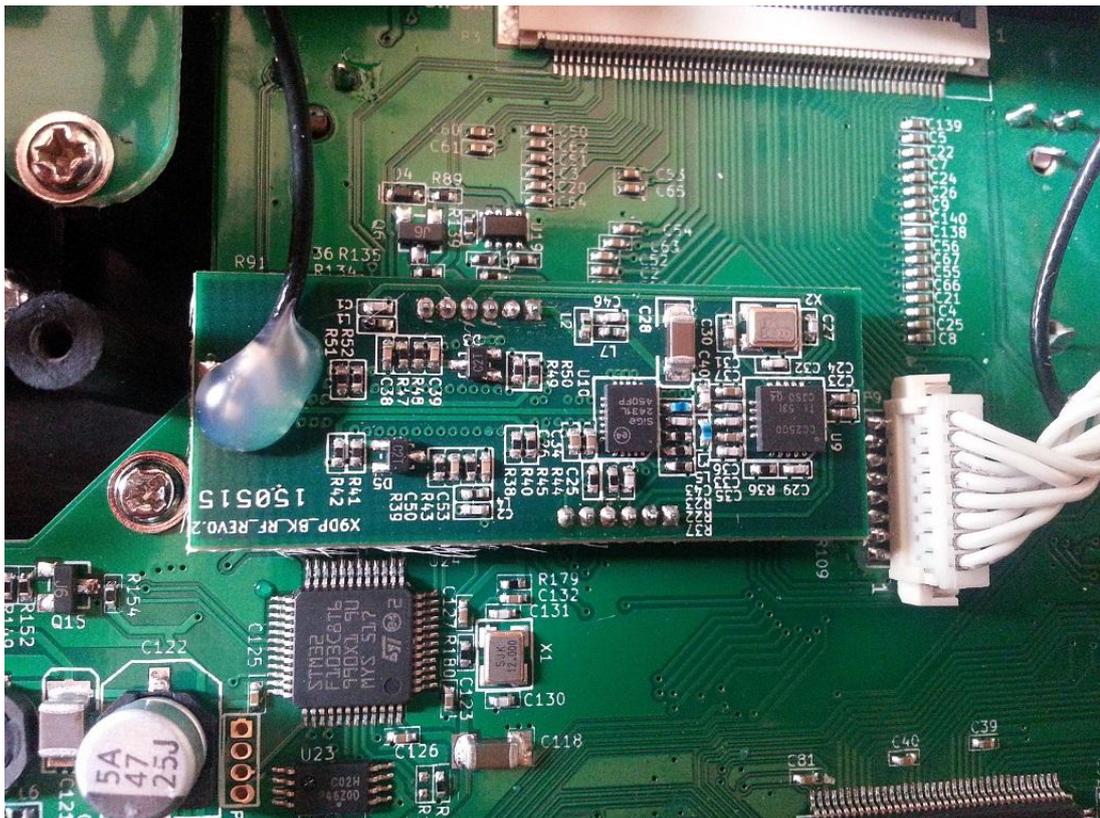
Das ist ärgerlich wenn man Modelle von der Taranis auf die X9E überträgt

openTx für Taranis Anleitung Deutsch
Innenaufbau, Akku 8 Zellen NiMH oder beliebigen Akku von 6-15V möglich



Kleine Platine links ist der Drehgeber, daneben sieht man den Elko 100uF/16V

Das interne XJT HF-Modul

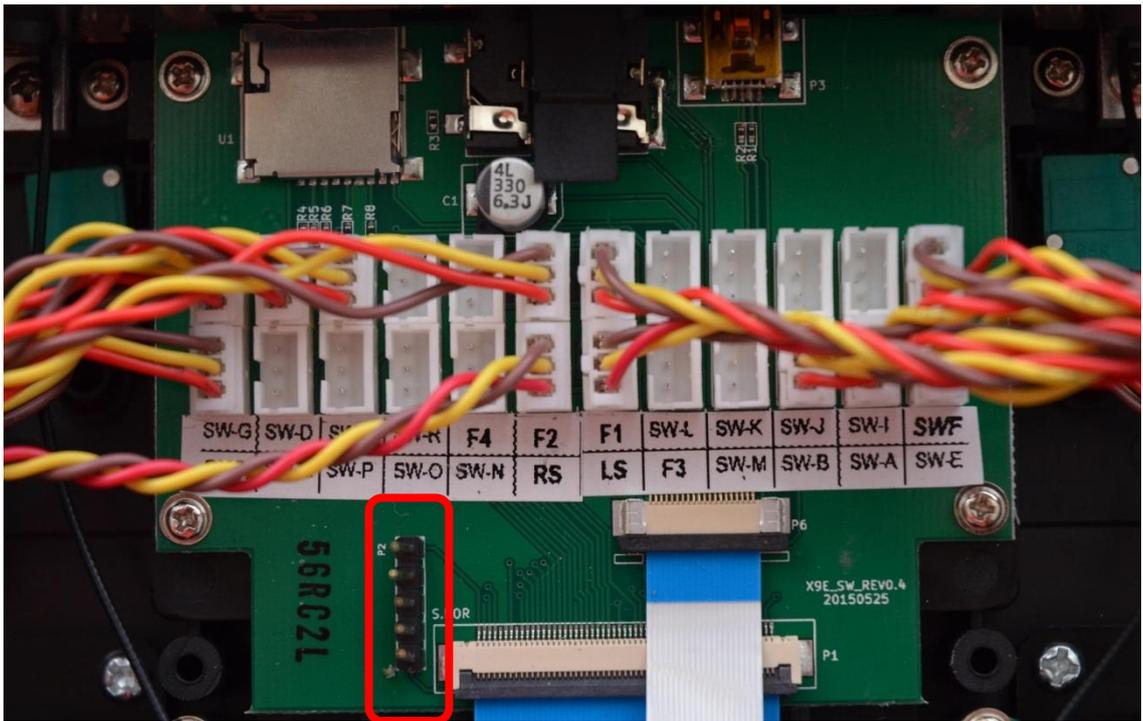


openTx für Taranis Anleitung Deutsch
Hauptplatine unter der Akku Halterung mit Batterie für Echtzeituhr

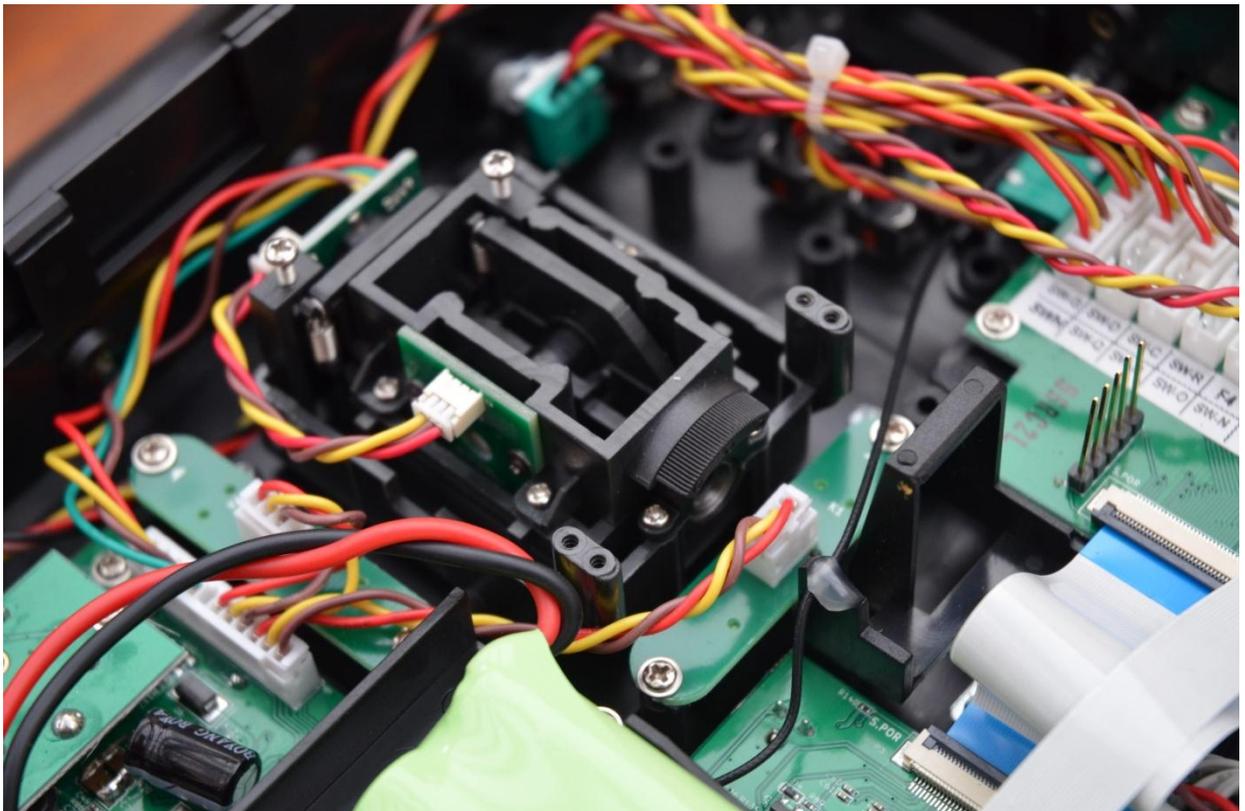


Echtzeituhr RTC Stützbatterie Typ CR1220 3V/ 35mAh
Falls sich die RTC immer wieder resetet, Elko 100uF/16V gegen 470uF/16V austauschen

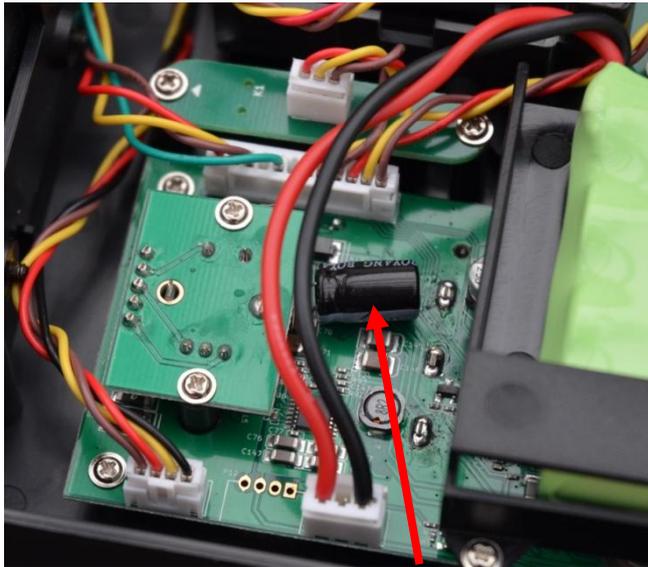
Freie Stecker für zusätzliche Schalter und Potis, Stiftleiste für weiteres HF-Modul



Knüppelaggregat mit Potis



Akkuanschluss und Spannungswandler



Drehgeberplatine und Elko 100uF/16V

Bluetoothmodul BT4.0



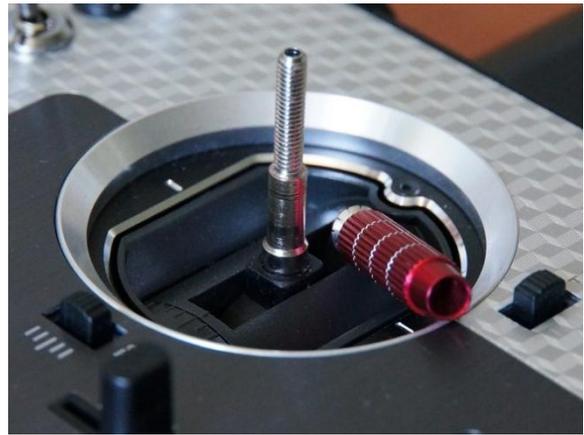
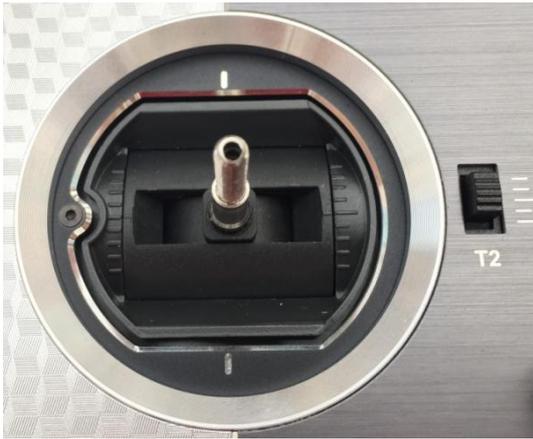
X9E mit OpenTx V2.1x als Bediensoftware, Drehgeber ersetzt 3 Tasten



Farb LCD-Anzeige im Detail



**Knüppel mit M4, und Bohrung für Knüppelschalter (geht nicht ganz durch Kunststoffteil)
Für Knüppelschalter Kunststoffteil mit 1,0 bis 1,5mm durchbohren**



Höhenverstellung des Knüppels mit Innen-Imbus

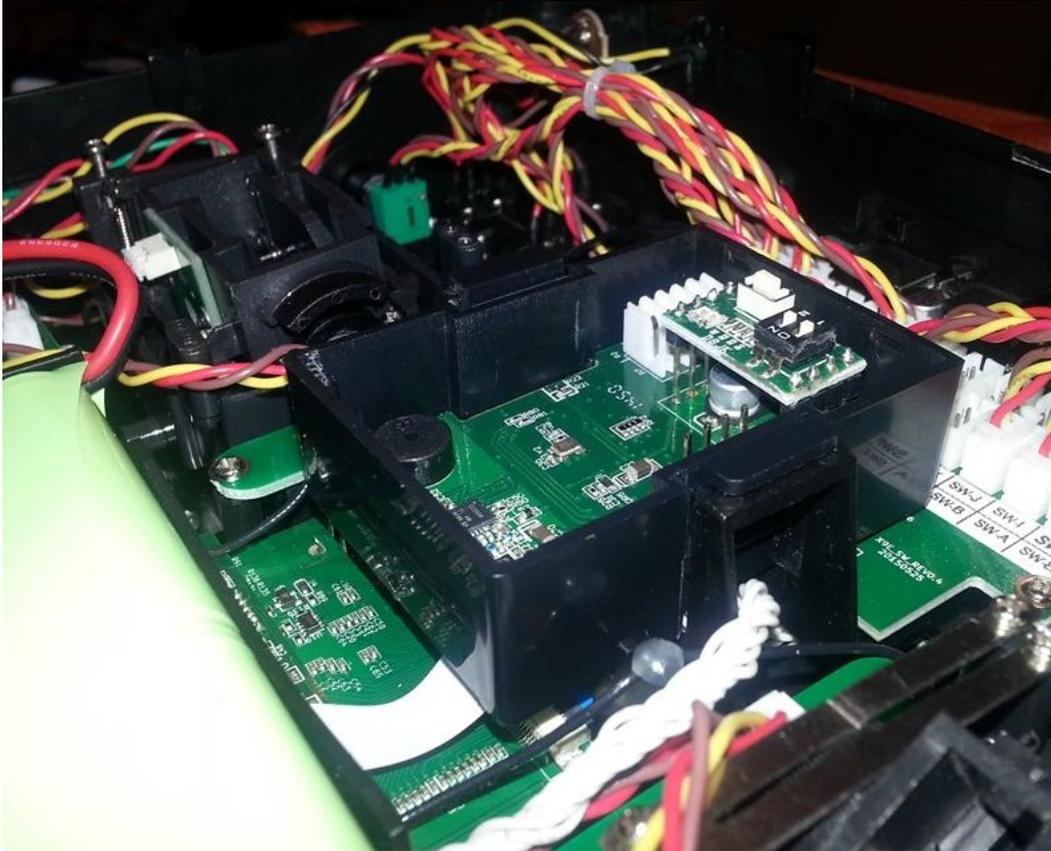
X9E mit 2 Knüppelschalter von mw-modellbauservice

Die Sticks sind von mw-modellbauservice, 40mm Länge und unten ein 5mm langes Sackloch mit 5mm Durchmesser, das passt dann exakt.
Der Einbau war einfach, man muss nur die Knüppeldurchführung im Kunststoff komplett durchbohren mit 1,0 bis 1,5mm, das sind nur ein paar zehntel Material.



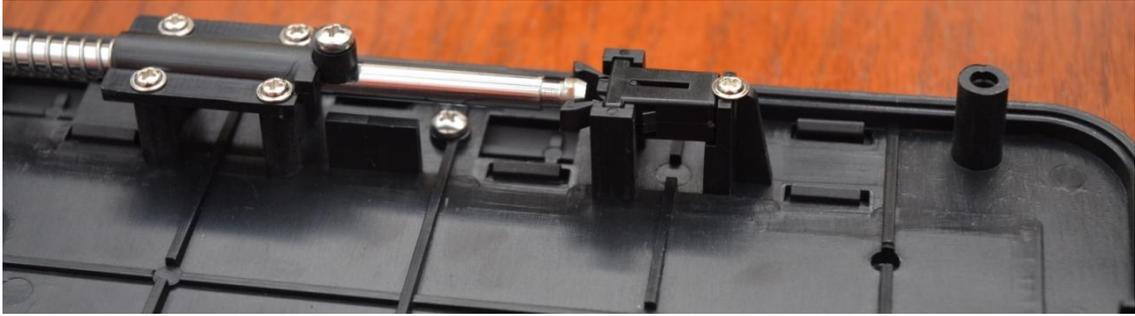
Im X9E wird ein zusätzliche HF-Modul innen eingebaut und die Antenne extra rausgeführt

XJT HF-Modul im Steckplatz eingebaut als „externes HF-Modul“



Die Sendeantenne des externen Moduls muss verlängert werden, der Deckel entfernt werden. Das ist nicht optimal!

openTx für Taranis Anleitung Deutsch
Mechanik des Tragebügels mit Klappmechanismus



Alternatives Sender-Pult mit Jeti –Bügeln zum umklappen (Quelle RC-Groups)



Teil G Die Sender 9XR Pro, 9XR, Th9x mit OpenTx V2.0x

Den Sender 9XR Pro mit openTx V2.0x flashen.

Der Sender 9XR Pro wird mit der Software ersky9xr von Mike Blandford ausgeliefert, die ähnlich wie openTx funktioniert, im Detail aber etwas anders ist. Er hat den gleichen kleinen LCD-Screen wie die 9XR und Th9x, aber einen 32bit Prozessor, viel Speicher, alle Schnittstellen, USB, SD-Karte und braucht immer ein HF-Modul. Der Schaltplan ist offen zugänglich und man kann einiges selber erweitern.



Zum flashen sind ein paar kleine Hürden zu überwinden.

Zuerst brauchen wir mal die openTx-Version für den **9XR Pro** (nicht für die 9XR)

Die bekommen wir über Companion, dort ein Profil für 9XR Pro erstellen, aller benötigten Optionen auswählen, dann downloaden.

Damit haben wir mal die openTx für den 9XR Pro.

Ablauf:

Auf dem Sender ist eine SD-Karte mit dem Unterverzeichnis **FIRMWARE** Dort ist das Betriebssystem **ersky9xr** abgelegt. Der installierte Bootloader von **ersky9x** greift auf genau dieses Verzeichnis **FIRMWARE** zu.

Dort kopieren wir jetzt die openTX-Version für den 9XR Pro rein. Dann können wir von dort auch auf openTx updaten, wie bei der Taranis auch. Die beiden unteren Trimmasthen nach innen halten, dann einschalten. Opentx auswählen, flashen, fertig. Wenn wir den Bootloader nicht verändern, also nicht updaten, ist das ok und alles ist gut.

Aber falls man den Bootlaoder von openTx einmal updated greift er auf **FIRMWARES** zu!

Man beachte das "S"! **Das ist leider der feine aber große Unterschied!**

Also auf der SD-Karte noch ein Unterverzeichnis **FIRMWARES** anlegen und dort auch noch mal die openTx-Version für den 9XR Pro draufkopieren, dann kann nichts schiefgehen.

Weitere openTx updates dann immer in **FIRMWARES reinkopieren!**

Hintergrund:

Es gibt im englischen kein Plural für Firmware, trotzdem haben die französischen Entwickler versehentlich **FIRMWARES** statt nur **FIMWARE** genommen.

openTx für Taranis Anleitung Deutsch

Die Sender 9XR, Th9x mit OpenTx V2.0x bedienen und flashen

Auch diese Sender arbeiten mit openTx V2.0x und können geflasht werden.
Beachte: Je nach Ausführung haben sie nur 64k oder 128k Speicher!



Bezeichner und Bedeutungen

Eingabewerte (in Rot, so wie sie auch in den Menüs auftauchen)

1. Sticks (cross-drivers, levers, gimbals): Steuerknüppel/Kreuzknüppel:

- 1. **Rud** (Rudder) **StR** (Seitenruder) **S**
- 2. **Ele** (Elevator) **HöR** (Höhenruder) **H**
- 3. **Thr** (Throttle) **Gas** (Gasknüppel) **G**
- 4. **Ail** (Ailerons) **QuR** (Querruder) **Q**

2. Potentiometer:

- **P1** – Poti links vorne, Bezeichnung “HOV PIT”
- **P2** – Poti rechts vorne, Bezeichnung “HOV. THR”
- **P3** – Poti links oben, Bezeichnung “PIT TRIM” oder ”AUX. 2” “POT. 7”

3. Schalter: (immer in Großbuchstaben!)

- **THR** - "Throttle Cut" Schalter, links vorne unten,
(nicht zu verwechseln mit Throttle Stick **Thr** (Gas-Knüppel)).
- **RUD** – Ruder Dualrate “RUD DR” = Seitenruder Dualrate
- **ELE** - Elevator Dualrate “ELE DR” = Höhenruder Dualrate
- **ID0, ID1, ID2, 3POS** – 3-Wege Schalter rechts oben, ID0 =vorne, ID1 =mitte ID2=hinten. Heißt aber auch “F MODE” oder “AUX 3”oder “MIX”.
- **AIL** - Aileron Dualrate “AIL DR” = Querruder Dualrate
- **GEA** – Landing gear “GEAR” = Fahrwerk Schalter
- **TRN** – Trainer Taster, das ist kein Schalter sondern ein Taster!
- **L1 .. L16** – Softwareschalter, virtuelle, programmierb.Schalter

Das Symbol **"!"**. stellt eine logisches NOT oder INVERS dar.

Schalter haben 2 Zustände Ein oder Aus, 1 oder 0, betätigt oder nicht betätigt, also **“Normal”** oder **“Invers”** .

Wenn man den z.B. den Schalter **"ELE"** Elevator Dualrate (Höhenruder Dualrate) im Menü auswählt, so kann er als **"ELE"** in Stellung vorne/unten oder als **"!ELE"** in Stellung hinten/oben ausgewählt werden.

Man muss nochmal darauf hinweisen dass in der Firmware jede Funktion oder Schalter total frei zugeordnet werden kann. Es gibt keine feste Schalterzuordnungen, das sind hier nur die Bezeichner wie sie eben am Sender für die Schalter angebracht sind.

Man kann mit jedem Schalter alles machen z.B. mit dem 3 Wege-Schalter auf alle Ruder Dualrate aufschalten. Oder mit dem ELE-Schalter den Motor ein-und ausschalten.

Alles ist frei verfügbar und belegbar!

Man beachte auch die Schreibweisen:

Alle Schalter sind in Großbuchstaben,

(THR, RUD, AIL, ELE, GEA, TRN, ID0, ID1, ID2)

Alle Analogwerte (Sticks und Potis) in gemischter Schreibweise

(Thr, Rud, Ail, Ele, P1,...).

Achtung: Genau das wird bei Mischeranweisungen oft verwechselt!

Eingaben/Werte editieren

Die Bedienung für Th9x und 9XR ist exakt gleich.

Es gibt am Sender 6 Tasten um durch die Menüs zu navigieren und zu editieren
Rechts die 2 Tasten MENU und EXIT, Links die 4 Kreuztasten als Cursor.
Sie werden hier in der Anleitung immer in eckigen Klammern gesetzt z.B. [MENU]

Manche Funktionen werden durch einen längeren Tastendruck (ca. 1Sec) aufgerufen
dann steht da z.B. [MENU LONG] alle anderen durch einen kurzen Tastendruck.

Dann ist da noch das Problem mit den 2 Tasten "+" und "-" die sind leider etwas
verwirrend und müssten eigentlich vertauscht werden. Das kann man auch machen,
wenn man den Sender öffnet und die Tasten vertauscht.

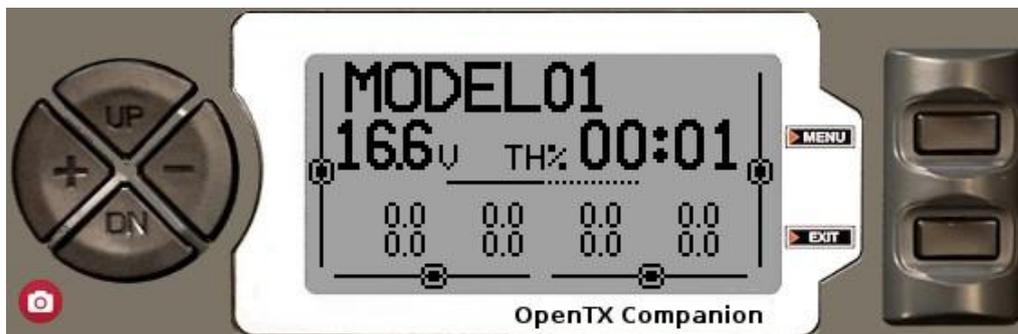
Wir verwenden hier aber die [RIGHT] und [Left] das ist dann logischer und praktischer
in der Anwendung.

[RIGHT] = rechte Taste, erhöht Zahlenwerte, Inc.

[LEFT] = linke Taste, verringert Zahlenwerte, Dec.

Das verwirrt zunächst, vereinfacht aber vieles.

Einfach nicht weiter drüber nachdenken



Die wichtigsten Tastenfunktionen aus dem Hauptbildschirm

[RIGHT LONG] öffnet das Menü für alle Modelleinstellungen.

[LEFT LONG] wechselt in die Grundeinstellungen des Senders

[DOWN LONG] wechselt in die Darstellung der Telemetrie-Anzeigen

[UP LONG] wechselt in die Statistik und Debug Anzeigen des Senders

Ist man in einem Haupt-Menü (der ist Cursor links oben, Seitenanzeige invers z.B. 2/12)

kann man mit [LEFT] / [RIGHT] durch alle Seiten durchblättern

Das Grundprinzip lautet:

Mit den 4 Cursor Tasten in eine Zeile, Spalte gehen, das wird invers dargestellt.

mit [MENU] in den Eingabemodus wechseln, das blinkt dann

mit [UP]/[DOWN] Werte ändern oder auswählen,

mit [MENU] Wert übernehmen und Eingabemodus verlassen.

oder [EXIT] Abbruch und Eingabemodus verlassen.

Editieren und abspeichern

Grundsätzlich gilt, dass geänderte Werte sofort abgespeichert werden. Man kann also den Sender ausschalten und alles ist schon gespeichert (mit Option **pgbar** sieht man wenn er abspeichert).

Alle Werte werden im internen EEPROM des Microcontroller abgespeichert. Trotzdem kann es dabei zu einer kurzen Verzögerung kommen den das abspeichern dauert ein paar Millisekunden. Man sollte also mit dem Ausschalten des Senders ca. eine Sekunde warten.

Es gibt keine "UNDO" Zurück-Funktion, jede Veränderung ist sofort gültig

In der Regel gilt (Es gibt Ausnahmen!):

Wenn ein Wert **Invers** dargestellt wird kann man nicht durch drücken von Links/Rechts [Right] / [Left] Werte verändern, sondern muss **zuerst** mit [MENU] in den Editiermodus wechseln, dann blinkt der Wert und dann erst kann man mit [LEFT]/[Right] bzw. [UP]/[DOWN] Werte verändern.

Zum Verlassen des Editiermodus [MENU] oder [EXIT] drücken

Die Ausnahmen vereinfachen Eingaben und ermöglichen schnellere, sofortige Eingaben in den Hauptmenüs.

Navigation durch Zeilen und Eingaben in mehreren Spalten

Mit den [UP] / [DOWN] / [LEFT] / [RIGHT] Tasten bewegt man den Cursor entlang der Zeilen und Spalten des jeweiligen Menü. Eingabebereiche werden dann invers dargestellt.

Die [MENU] Taste wird für den Start (Eingabebereich blinkt dann) und das Ende des Editiermodus benötigt (der Eingabebereich blinkt nicht mehr).

Mit der [EXIT] Taste verlässt man den Editiermodus
Nochmal [EXIT] drücken und man ist wieder im aufgerufenen Bildschirm
Oder mit [EXIT LONG] kommt man sofort in die Hauptanzeige zurück.

Zeilen in denen Eingaben nur an einer Stelle/Spalte möglich sind

Wenn man mit den ([UP]/ [DN] Cursor in eine Eingabe-Position kommt, die nur eine Spalte hat, kann man mit [RIGHT] / [LEFT] den Werte **sofort** ändern. Man muss nicht extra mit [MENU] in den Editiermodus wechseln.

Mit [MENU LONG] kann man dann bei bestimmten Menüs feste Zahlenwerte direkt durch globale Variablen (GV1-GV5) ersetzen, nochmals [MENU LONG] setzt dann den Originalwert wieder. Im Editiermodus ist der Cursor invers dargestellt und blinkt.
Um den Editiermodus zu verlassen einfach [MENU] oder [EXIT] drücken.

Werte in einer Checkbox ein/ausschalten

Mit den 4 Cursortasten ([UP], [DN], [LEFT], [RIGHT]) steuert man durch die Zeilen und Spalten, dabei werden die Eingabe-Positionen invers dargestellt.
In einer Checkbox wird mit Druck auf [MENU] die Funktion ein-oder ausgeschaltet. Das geht aber **auch** gleich mit [RIGHT] ein, und mit [LEFT] aus.

Eingabe abschließen

Alle Änderungen werden sofort in den Einstellungen dargestellt, Änderungen sofort abgespeichert und wirken sich am Sender sofort aus.

Wert Änderungen werden mit [EXIT] oder [MENU] abgeschlossen. Es gibt keine Undo-Funktionen, man kann also nicht einfach wieder zu den vorherigen Werten zurück.

Bearbeiten von Zeilen

In den Open9x Menüs sind manche Eingabe mit Zeilen zu ergänzen/löschen z.B. bei den Modellisten, Mischern, DR/Expo, den Schaltern usw.

In all diesen Fällen ist das Vorgehen immer gleich

Einfügen von Zeilen

Mit den Cursorstasten [UP] und [DOWN] auf die Zeile gehen und mit [MENU] die Zeile auswählen, diese wird dann invers dargestellt.

Mit [LEFT LONG] wird eine **neue Zeile darüber** eingefügt

Mit [RIGHT LONG] einen **neue Zeile darunter** eingefügt.

Löschen von Zeilen

1. Zeile mit den Cursorstasten [UP] und [DOWN] anwählen
2. Mit [MENU] die Zeile auswählen, wird dann invers dargestellt
3. Mit [EXIT LONG] wird dann diese Zeiler gelöscht

Editieren von Zeilen/Werten Untermenü aufrufen

1. Zeile mit den Cursorstasten [UP] und [DOWN] anwählen
2. Mit [MENU] die Zeile auswählen, wird dann invers dargestellt
3. [MENU LONG] ruft dann das Untermenü auf und man ist im Editormodus

Im der Modellauswahlliste (1/12) ist immer das Modell mit dem Stern „*“ aktiv.

Kopieren von Zeilen/Werten

1. Zeile mit den Cursorstasten [UP] und [DOWN] anwählen
2. Mit [MENU] die Zeile auswählen, wird dann invers dargestellt
3. Mit den Cursorstasten an die Zielposition gehen
4. Mit [MENU] kopiert dann an die aktuelle Zeile

Verschieben Zeilen/Werten

1. Zeile mit den Cursorstasten [UP] und [DOWN] anwählen
2. mit [MENU] die Zeile auswählen, wird dann invers dargestellt
3. nochmal [MENU] die Zeile wird umrandet dargestellt
4. mit den Cursor das Ziel anwählen
5. [MENU] verschiebt dann diese Zeile

Texte eingeben

In manchen Seiten/Bereichen muss man Texte eingeben, (Modellname, Name der Flugphase usw.)

1. Mit [MENU] in den Editiermodus, einzelnes Zeichen wird invers dargestellt
2. Mit [LEFT] und [RIGHT] den Cursor positionieren
3. Mit [UP] und [DOWN] das Zeichen ändern, Ziffern, Sonderzeichen, (nur Großbuchstaben, Ziffern und ein paar Sonderzeichen).
4. Mit [LEFT LONG] wird von Groß auf Kleinbuchstaben gewechselt und umgekehrt
5. Mit [RIGHT LONG] wird von Groß auf Kleinbuchstaben gewechselt und umgekehrt und dann gleich zur die nächsten Position gesprungen.
6. Beenden mit einfachem ([MENU] oder [EXIT])

Arbeiten mit Auswahlwerten

In Open9x gibt es auch die Möglichkeit Schalterstellungen, Potis, Sticks usw. direkt abzufragen. z.B. Schalterstellungen beim Einschalten, Mittenposition der Potis durch kurzes Piepsen, Auswahl der Flugphasen die in Mischer oder Dualrate/Expo aktiv sein sollen.

In den Menus gibt es dazu Zeichenketten z.B. (01²34) für die Flugphasennummern FP0-FP5 oder (RETA123) bzw. (SHGQ¹23) für die Mittenpositionen von Sticks und Potis.

Jedes Zeichen korrespondiert dabei mit einem Element für das es steht.

Ist ein Element **aktiv** wird es **invers** dargestellt, nicht aktiv als normale Darstellung.

Das kann man einstellen, indem man mit den Cursors [LEFT]/[RIGHT] die Position anwählt, dann wird diese Position wieder invers blinkend dargestellt. Ein kurzer Druck auf [MENU] und man kann diese Position jeweils aktivieren oder deaktivieren.

Verlassen des Editiermodus durch [MENU] oder gleich durch [UP] oder [DOWN]

Flashen von Th9x und 9XR mit AVR ISP-Schnittstellen

Der Th9x hat keine AVR ISP Programmier-Schnittstellen eingebaut, die muss man selber einbauen,

Der 9XR hat die 6-polige AVR ISP Schnittstelle schon fertig eingebaut!

Alls notwendige findet man in den Foren und unter youtube:

9xforums: <http://9xforums.com/forum/>

Open9x: <http://code.google.com/p/open9x/>

Companion9x: <http://code.google.com/p/companion9x/>

Modifikationen: http://9xforums.com/wiki/index.php/Hardware_Mods_%26_Other_Guides

Ein paar Begriffe vorweg:

Flashen dabei wird ein bestimmter Speicherbereich (den Flashspeicher) eines Prozessor mit neuer Software überschreiben, also mit einer neuen Firmware laden. Der Atmega64 hat 64k Flashsspeicher zur Verfügung. Dort ist die Software enthalten die den ganzen Sender steuert. Diesen Speicherinhalt müssen wir ersetzen, dann haben wir ein ganz anderes Verhalten des Senders.

EEProm ist der Speicherbeich in dem die Einstellungen für die Modelle im Sender hinterlegt sind. Also Dualrate, Expo, Timer, Mischer, Kurven, Modellname, usw.

Fuses sind Einstellungen im Prozessor, von denen lassen wir unter allen Umständen die Finger weg, denn damit können wir sehr viel zerstören, also Finger weg, das ist nur was für Atmega-Spezialisten.

Ram ist der Speicher im Prozessor der für die Berechnungen und Variablen gebraucht wird, da haben wir keine Zugriff, interessiert und auch nicht.

ISP-Programmierschnittstelle ist dazu da um auf den Atmega64 Prozessor zugreifen zu können. Dazu sind in den Sender 6 Leitungen einzulöten und über einen Stecker herauszuführen. Das Verhalten dieser 6 Leitungen wird durch einen USB- ISP- Programmieradapter bestimmt und gesteuert. Diese Schnittstelle gibt es als 6- und als 10-poliger Anschluss.

AVRdude das ist das eigentliche Programmier-Programm das im Hintergrund auf dem PC läuft und den **USB-ISP Programmieradapter** ansteuert, die neue Software in den Prozessor schreibt und überprüft dass alles fehlerfrei abgelaufen ist.

Bevor man loslegt sollte man sich im Internet schlau machen und auch unter Youtube die Videos zum Umbauen ansehen.

Notwendige Umbauten vorab:

Hintergrundbeleuchtung einbauen, **AVR ISP-Programmierstecker** einbauen

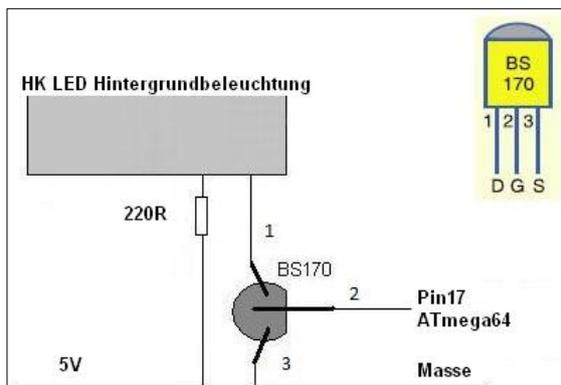
Eventl. Verpolungsschutzdiode überbrücken damit ein Akku (Lipoly, NiMH, LiFe) ganz normal geladen und nicht ausgesteckt werden muss (siehe Schaltplan Stromversorgung Th9x).

Telemetrie für Anzeige am LCD-Display nachrüsten. **Audio** und **Haptic**-Mods

Hintergrundbeleuchtung bei TH9x nachrüsten: (im 9XR schon eingebaut)

Ich habe **zuerst** eine blaue Hintergrundbeleuchtung eingebaut, weil man später da nicht mehr gut ran kommt, da dazu die komplette Platine ausgebaut werden muss.

Es gibt sie für ca. 5€ als fertiges Modul bei HK in Weiß, Grün oder Blau mit einem Zwischenstecker für die Platine zum einschleifen.



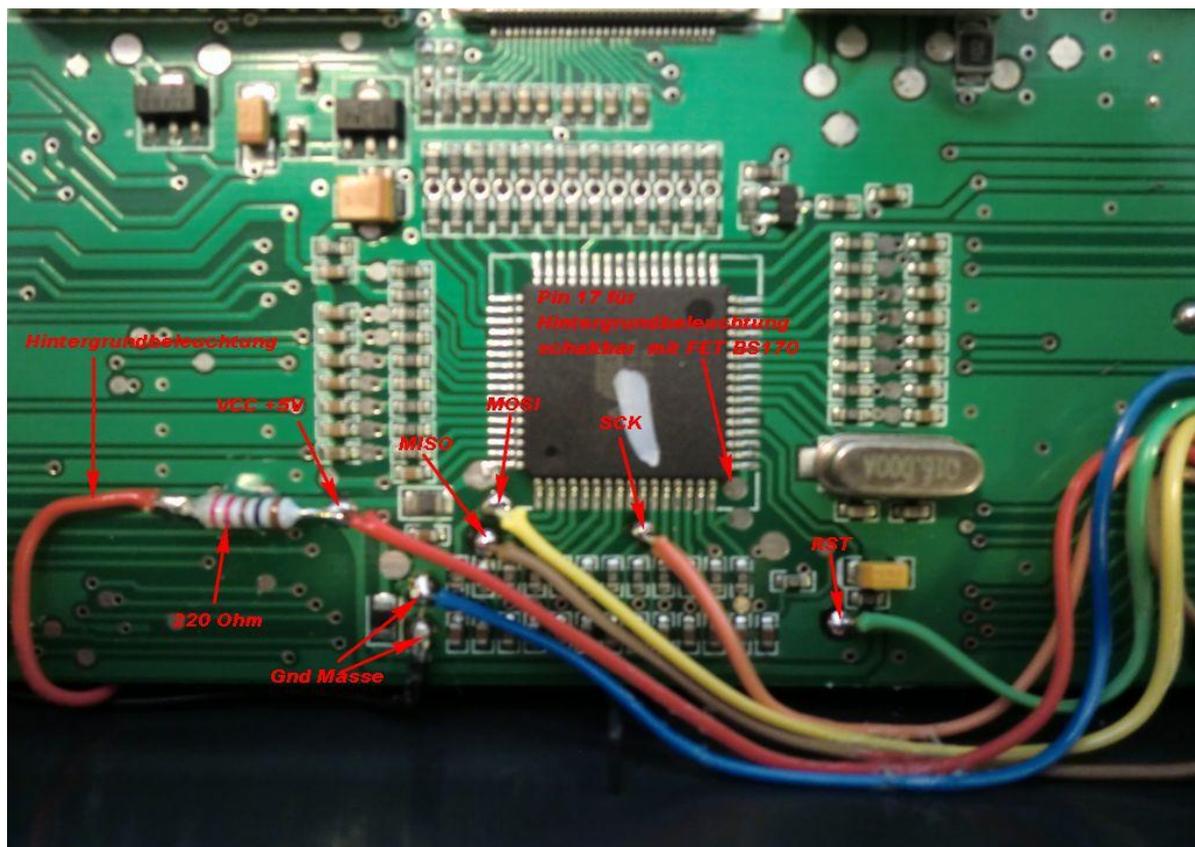
Meine Hintergrundbeleuchtung hängt aber mit 220 Ohm direkt an 5V (rot), macht 10mA, und ist mit den Kabeln links eingebaut. (2 Led 2,65V bei 10mA) Das habe ich gemacht weil dieser blöde Zwischenstecker Murks ist, oft einen Kurzschluss hat und auch nur ein Vorwiderstand 470-680Ohm für die LED an 9,6-12V Akkuspannung drinnen ist. Und es spart viel Platz.

Flashen Teil 1 von 4 Einlöten der 6 Leitungen für das Flashen des Senders

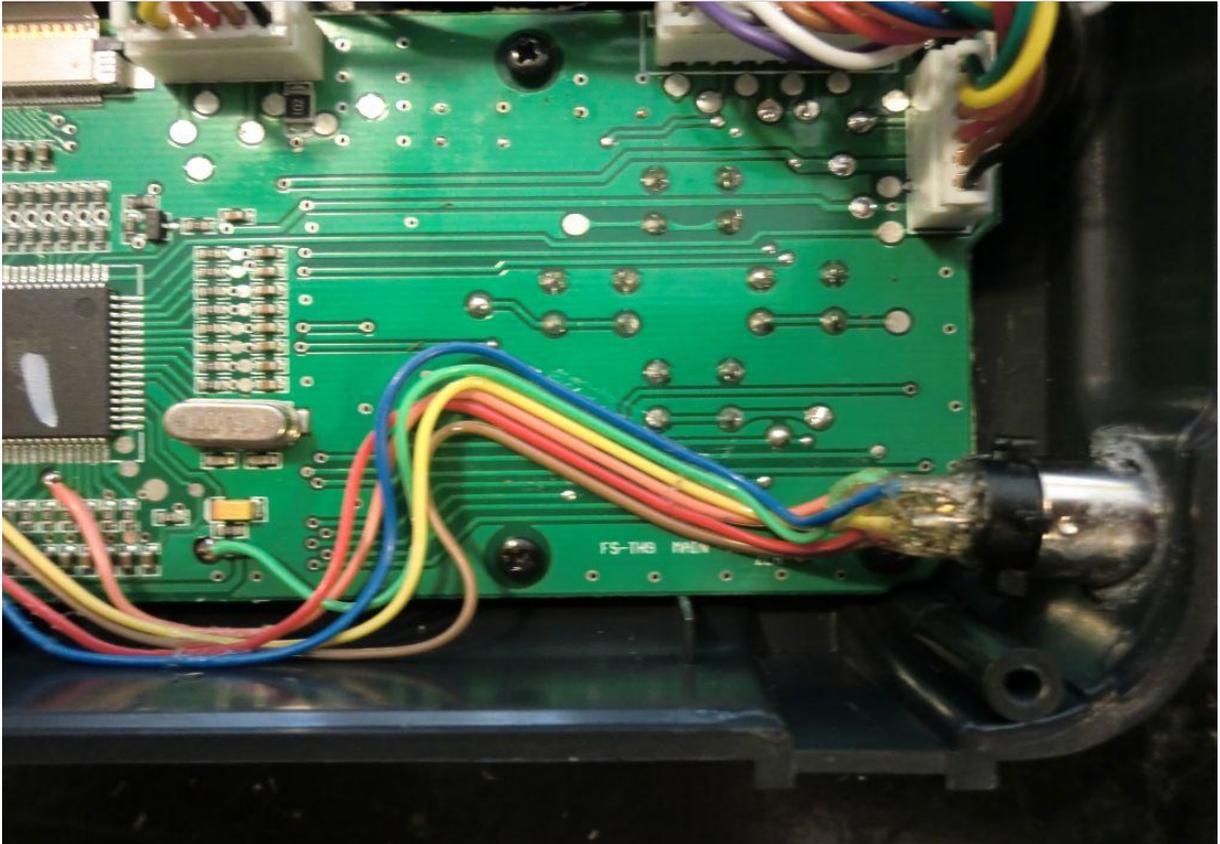
Der Sender wird geöffnet und der 12-polige Stecker zum Gehäusehinterteil vorsichtig abgezogen. Um den Prozessor herum sind die 6 Lötpins zu erkennen die wir brauchen um die Leitungen anzulöten.

MISO, MOSI, SCK, RST, GND/Masse, VCC +5V

Ich habe das Ganze auf eine 6-polige Mini-Din Buchse gelegt und mit einem 10mm Loch seitlich am Gehäuse rausgeführt.



ISP-Anschluss und Hintergrundbeleuchtung einlöten

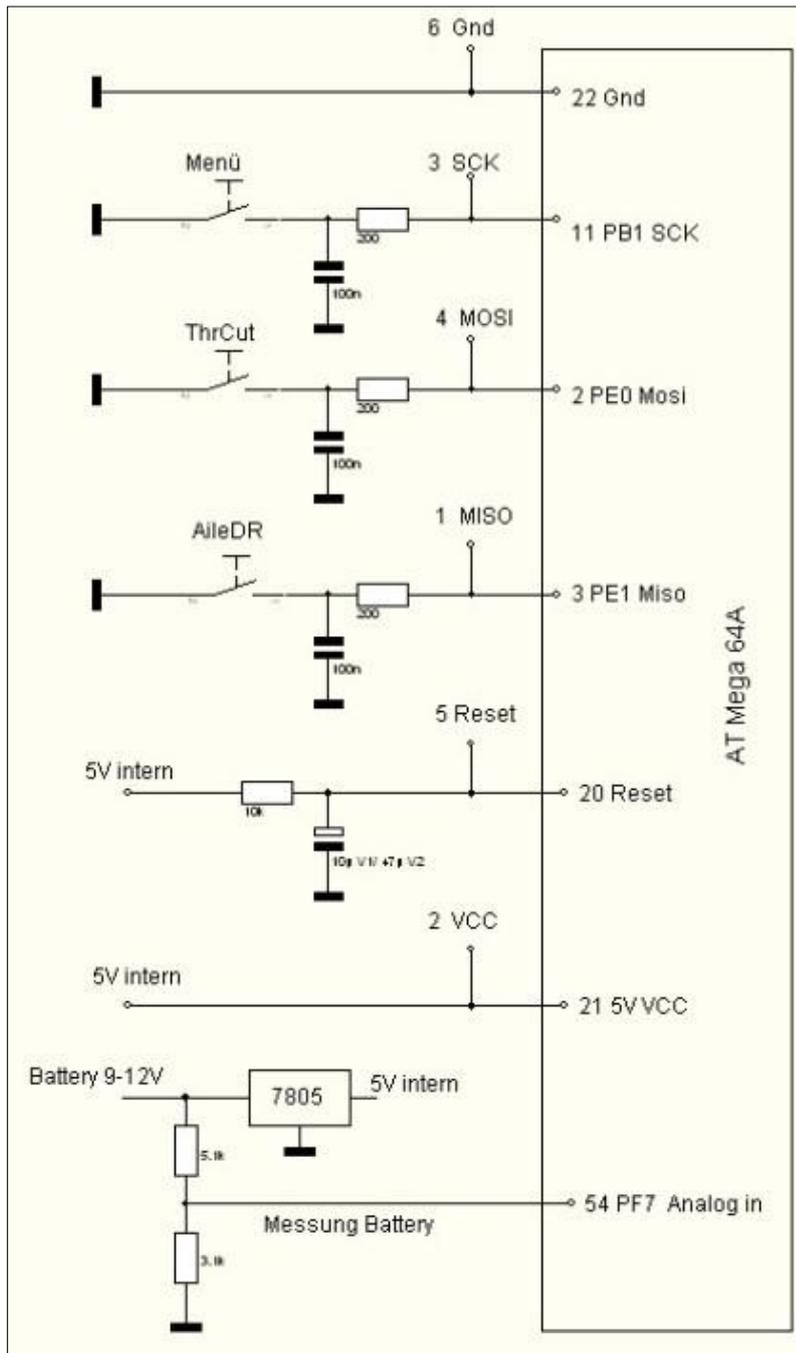


Mini-Din Buche im Gehäuse



Belegung von Stecker, Kabel und Farben

Im Sender 9XR ist der 6-polige Wannenstecker als ISP Programmer schon fertig eingebaut!



Auszug aus dem Schaltplan des Senders für das Verständnis

Man beachte die beiden Schalter **ThrCut** und **AileDR**.

Zum Flashen müssen sie betätigt sein, d.h. der Sender muss beim Einschalten piepsen! Sie dürfen also nicht in Grundstellung stehen, da sie sonst die Signale MISO und MOSI zu stark belasten.

Die Menü-Taste ist nicht betätigt und beeinflusst SCK nicht.

Das hängt auch stark vom verwendeten USB AVR-ISP Programmierer aber, je nach eingebauten Schutzwiderständen kann das passieren, muss aber nicht.

Funktionen der Schalter und Taster am Sender

Die Schalter am Sender ThrCut und AileDR sind in Grundstellung geschlossen
(Es kommt beim Einschalten des Senders keine Fehlermeldung Switch Error am Display)

Zum Flashen müssen die Schalter aber geöffnet sein, damit die Ausgänge
Im Programmiergerät für Mosi und Miso nicht zu stark belastet werden.

(200 Ohm an Masse wenn die Schalter geschlossen sind!)

Die Menütaste am Sender ist nicht betätigt, offen und kein Problem für SCK

Es kommt dann beim Einschalten die Meldung Switch Error, das muss so sein!

Stromversorgung 5V intern oder extern

Sender einschalten: 5V vom Sender,

dann darf vom Progammer keine 5V kommen!

oder:

Sender ausgeschaltet lassen,

dann kommen die 5V vom Programmiergerät.

Dann meldet aber der Sender Unterspannung und Piepst.

Das muss so sein, denn die Batterieüberwachung

meldet Unterspannung des Akkus da keine Spannung gemessen wird.

Reset muss an Pin 20 das Signal auf Masse ziehen.

Dort ist aber 10uF oder 47 uF verbaut.

Das kann ein paar ms dauern, vorher darf mit dem flashen nicht begonnen werden.

Im Programmiergerät sitzt hier meist ein NPN -Transistor der das Resetsignal auf Masse
zieht und hält.

Jetzt kann man flashen.

Flashen Teil 2 von 4

Im Sender ist jetzt der ISP-Anschluss eingebaut und auf eine 6 Polige Mini-Din Buchse gelegt.

Jetzt brauchen wir noch einen Adapter:
6 Polig Mini-Din Stecker auf 6 Polig Schneide-Buchse.

Das sieht dann so aus:

Bitte immer wieder mit dem Ohmmeter messen, dass kein Kurzschluss eingelötet wurde und die Nummer zu den Drahtfarben passen!



Einen guten AVR-ISP-Programmer besorgen unter www.myAVR.Com

Als USB-ISP Programmer verwende ich einen **mySmartUSB light**, einfach darum, weil ich den für alle anderen Atmel Projekte auch habe und immer ohne Ärger funktioniert.

Für den Erstanwender eine wichtige Besonderheit, es muss zuerst der Silab-Treiber installiert sein bevor der USB-Programmer das erste Mal angesteckt wird, sonst gibt es Ärger und muss umständlich angepasst werden.
(Gilt aber prinzipiell für alle USB-Geräte, zuerst den Treiber laden, dann erst anstecken!)

Es gibt auch andere USB ISP-Programmer für ca. 4-10€, dann aber entsprechend anpassen.

Hier heist es aufpassen, den am Markt gibt es sehr viel Schrott!

Die einen gehen sofort, die anderen nicht. Keine aktuellen Treiber, zu große Schutzwiderstände, Timing nicht einstellbar, SCK kommt nicht, läuft nicht mit AVRdude, nicht als STK500 usw. Da hilft nur sich vorab in den einschlägigen Foren schlau machen bevor man sich Ärger einkauft.

Diesen Scheiß/Ärger muss ich mir nicht antun, das Ding hat einfach zu funktionieren, da spart man am falschen Ende.

Siehe auch unter 9xforums, ISP-Programmer

Der AVR ISP-Programmer: mySmartUSB Light

Für diesen AVR ISP USB Programmer gibt es 2 wichtige Programme:

Supportprogramm: **SupportBox_MSUL.exe**

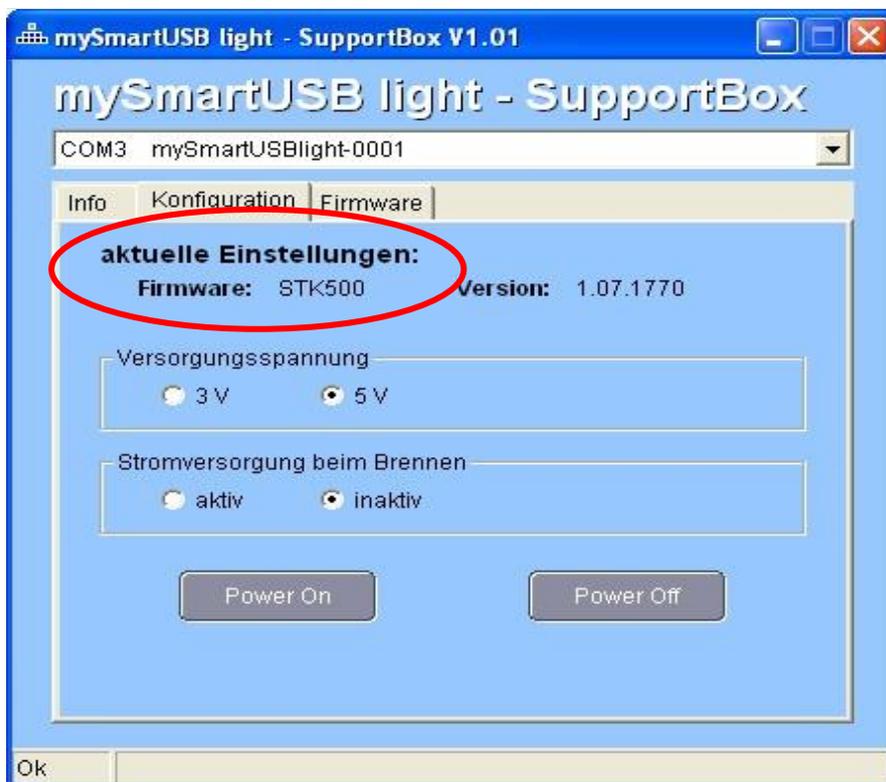
Programmiertool: **myAVR_ProgTool.exe**

SupportBox_MSUL.exe

Dort sieht man den COM-Port (hier COM3) unter dem er sich am PC angemeldet hat und die Einstellungen wie er sich verhalten soll:

wie ein **Atmel STK500**,

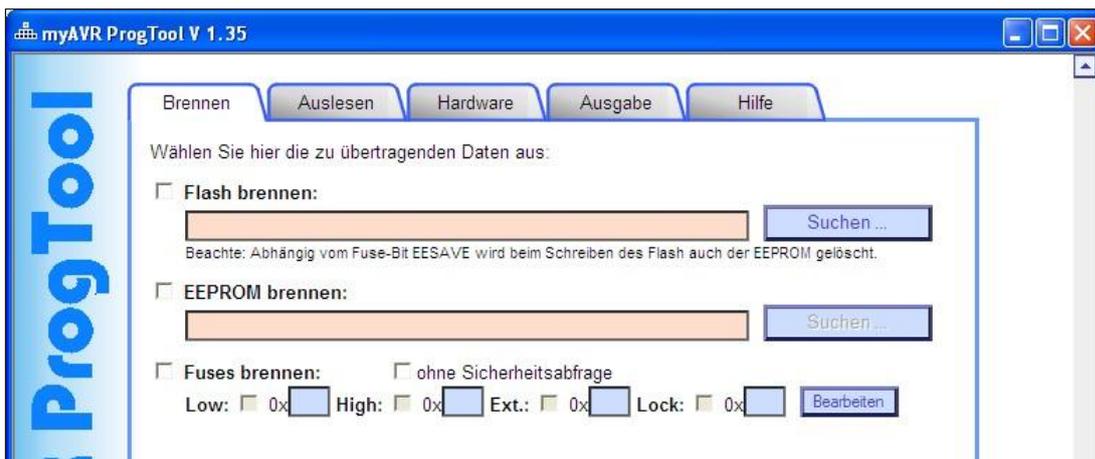
und ob er eine Spannung 3.3V oder 5V ausgeben soll oder nicht.



Das andere Programm ist das eigentliche Programmierwerkzeug **myAVR_ProgTool.exe** um den Flashspeicher mit der Firmware für den Sender zu programmieren (OpenTx für TH9x oder 9XR) und das EEPROM für die Einstellungen und Modelle zu programmieren.

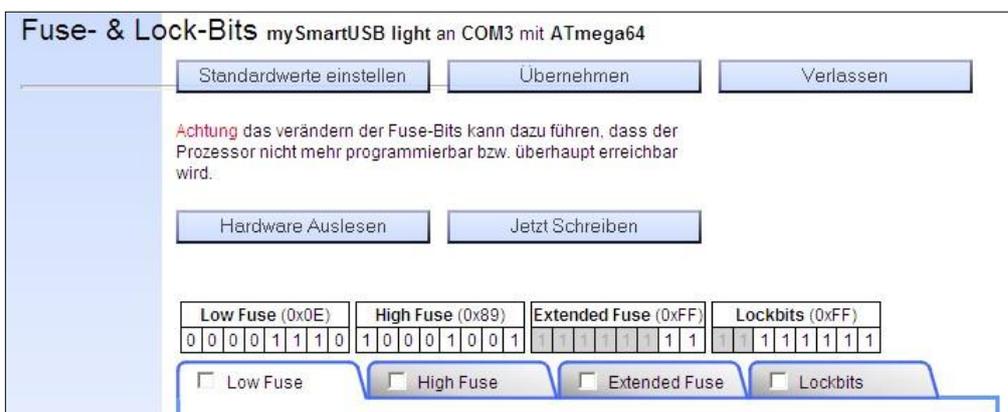
Damit kann man jetzt den Sender schon flashen indem man die .Hex oder .Bin Files überträgt ohne dass man CompanionTx verwendet.

Auch die Modelle könnte man so übertragen, ohne dass man Companion braucht.



Die Fuses kann man mit Bearbeiten und dann Hardware Auslesen ansehen. Ich hab sie einfach mal an der Th9x für den Atmega 64 ausgelesen.
LowFuses=0E HighFuses=89 ExtendedFuses=FF Lockbits=FF

Von den Fuses einfach die Finger lassen!



Arbeiten mit CompanionTx

Da die Funktionen und Optionen für den Sender sehr umfangreich sind, stellt man sich zuerst mal

a: die Firmware für den Sender zusammen mit div. Modulen und Funktionen

b: Programmiert am PC ein Modell und simuliert erst mal die Funktionen am PC

Dazu wird das Programm Companion9x benutzt.

Also suchen, downloaden, installieren, deutsch einstellen

CompanionTx starten:

Jetzt muss man darin erst mal 2 wichtige Einstellungen machen

a: Für das Brennprogramm und den verwendeten Brenner (z.B. an Com3 als **STK500!**)

Damit weis AVRdude (das eigentliche Brenn-Programm) was es wie wo machen soll und arbeitet mit der Brenn-Hardware mySmartUSB light zusammen!

b: Für die Grundeinstellungen der Firmware die erzeugt werden soll.

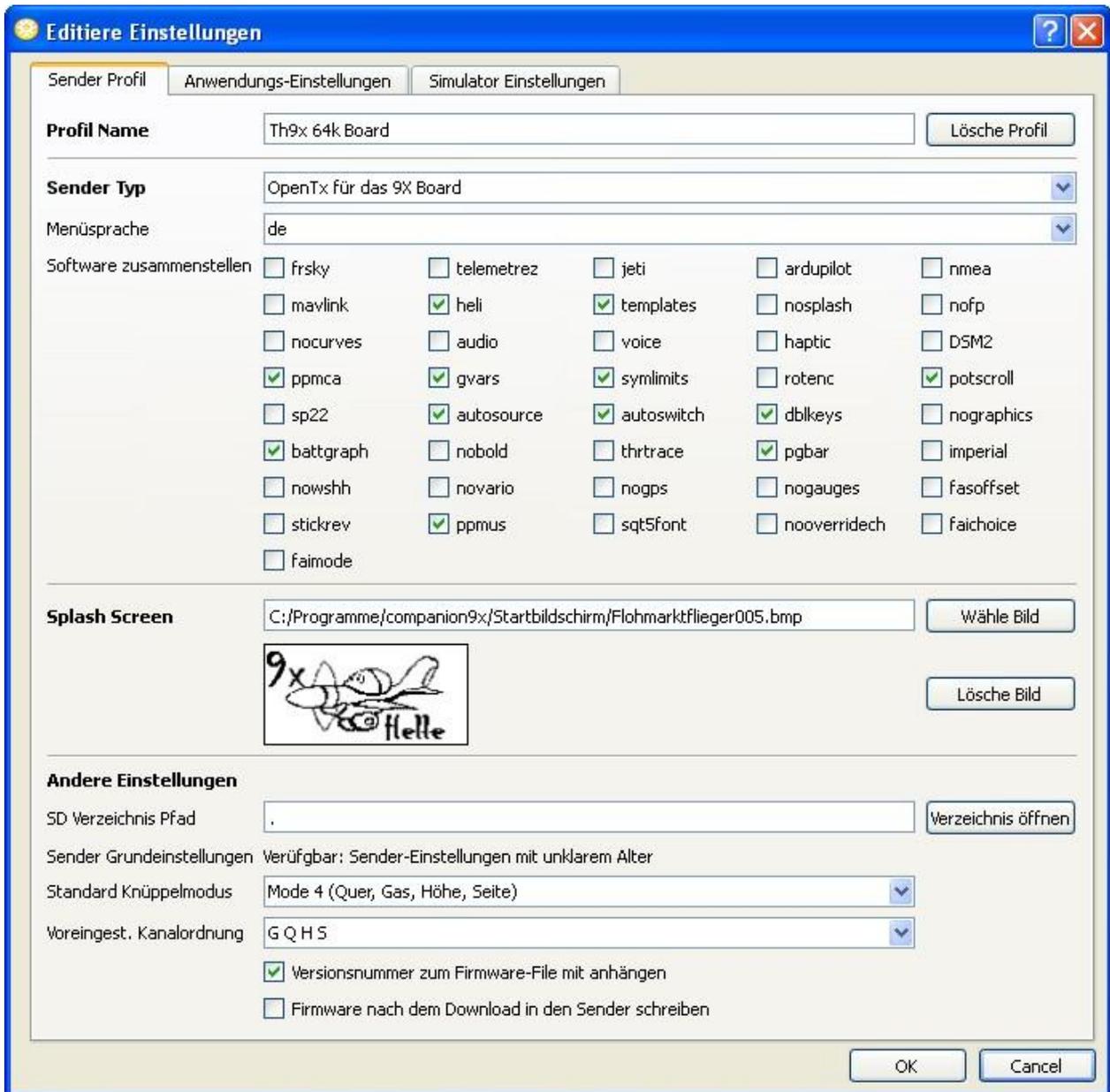
Firmware wählen, hier opentx, die Funktions-Module zusammenstellen, den Startbildschirm, usw

Einstellungen für das Brennprogramm



Wer einen anderen Programmierer hat kann unter Programmiergerät eine Liste aufmachen und seinen AVR ISP Programmer suchen.

Die Schnittstelle findet man unter Windows, Start, Systemeinstellungen, System, Hardware, Geräte manager, Anschlüsse (Je nach Windowsversion geringfügig anders)

Grundeinstellungen für CompanionTx

Das sind gute Einstellungen von Software-Optionen die mit einem normalen Flsky-HF-Modul in Sender zusammenpassen.

FrSky, Jeti, DSM2 sind andere HF-Module, die andere, weitere Optionen brauchen.

Von FrSky gibt es HF-Module mit und ohne Telemetrie. Die kann man beide problemlos benutzen. Aber **nur** wenn man auch die Hardware-Modifikation für die Telemetrie gemacht hat darf man FrSky anwählen, sonst nicht.

Das wird oft missverstanden.

Der Grund ist eigentlich ganz einfach: Die Softwareoption „frsky“ erwartet 2 Schalter an anderen Pins und erwartet auch dass die Telemetrie-Modifikationen gemacht wurden.

Man kann FrSky-Module mit Telemetrie immer einbauen, aber: Wenn keine Modifikationen gemacht wurden dann nicht FrSky anwählen sondern „normal“ lassen.

Flashen Teil 3 von 4

Den Startbildschirm (**Splash screen**) an der Funke kann man selber erstellen oder fertige Bildchen anpassen. Jedes einfache Bildbearbeitungsprogramm ist dazu geeignet.

Ich arbeite gern mit Irfanview

Entscheidend dabei:
Bildauflösung reduzieren auf 2 Farben schwarz/weiß
Bildgröße reduzieren auf 128x64 Pixel

Der Splash screen wird in die Firmware integriert und muss eingebunden werden, somit steht er in Flashspeicher und nicht im EEPROM!

Splash screen, siehe auch unter:

<http://9xforums.com/forum/viewforum....5415241c13362a>

CompanionTx OpenTx Flashen, Einstellungen für Splash Screen



Flashen Teil 4 von 4

Statt gleich zu brennen, lesen wir zuerst die Original-Software aus und speichern sie ab. Da passiert noch nichts Großartiges und wir machen nichts kaputt.

Flashspeicher und EEPROM auslesen und Brennen können wir mit dem Programm **myAVR_ProgTool.exe**, das beim **mySmartUSB light** ISP-Programmer dabei ist.

(oder aber auch aus Companion9x, das ist mal egal).

Ablauf genau beachten:

Sender einschalten.

Dann die beiden Schalter ThrCut und AileDR betätigen, das heist zu einem heran schalten, damit die 200Ohm nicht gegen Masse geschaltet werden und die Programmier-Signale unzulässig belasten. (siehe Schaltbild oben)

Den ISP-Programmer **mySmartUSB light** an den PC stecken und mit dem Supportbox-Programm **SupportBox_MSUL.exe** so konfigurieren, dass keine Spannung rausgeschaltet wird. Keine 5V und keine 3,3V! Siehe Bilder oben

Dann Programm **myAVR_ProgTool.exe** starten

An den Fuses machen wir gar nichts, nie! Da lassen wir immer schön die Finger weg!!

Erst jetzt an den Sender anstöpseln und es passiert erst mal gar nichts!



Auslesen des Flash und in Datei schreiben,

Dann Auslesen des EEPROM und in Datei schreiben, Fertig!

Wenn das schon mal problemlos geklappt hat funktioniert unsere Schnittstelle prima!

Jetzt können wir den Sender umprogrammieren und mit einer eigenen Firmware laden z.B. openTx mit ausgewählten Modulen und Funktionen.

Ablauf:

Zuerst die Firmware in den Flashspeicher brennen,
EEProm formatieren per Tastendruck
dann erst das EEPROM brennen!

Sender ausschalten
Programmierstecker abziehen, die beiden Schalter wieder in Grundstellung schalten!

Sender einschalten, er startet mit der neuen Software, gewonnen!

Jetzt sind noch 2 Dinge zu tun:

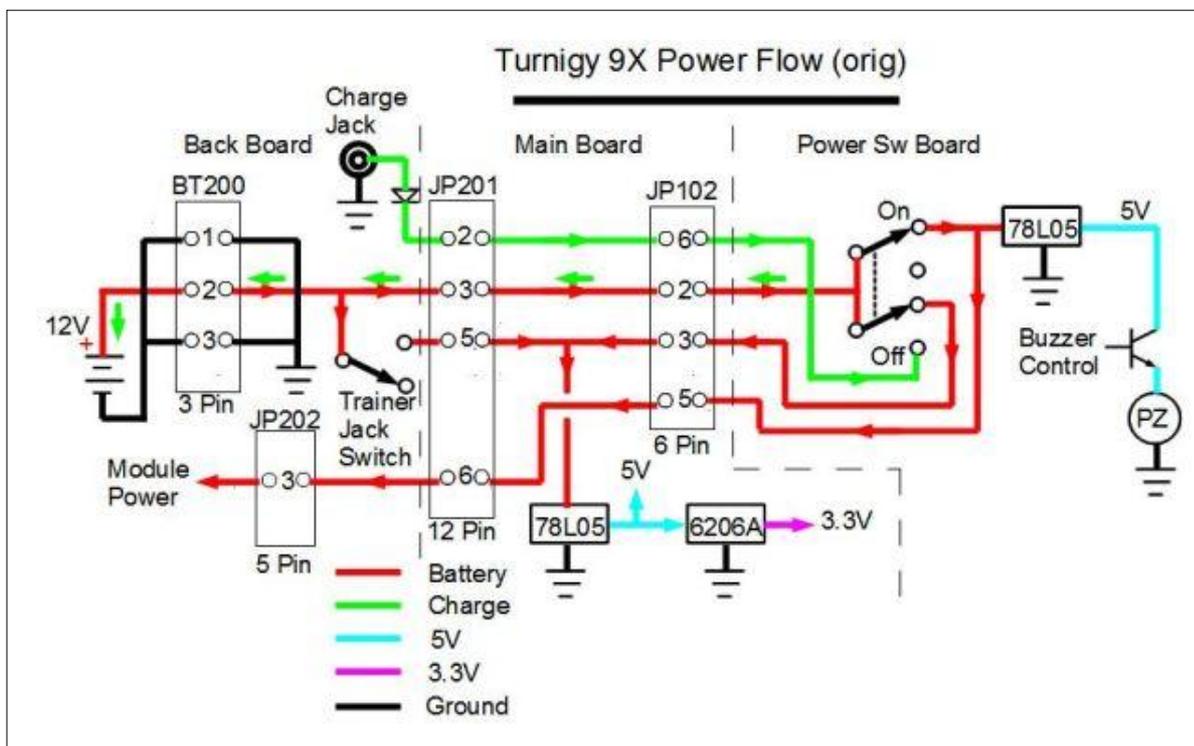
Menu im Sender für Kontrast suchen und einstellen, Werte so um die 30,
Standartwert von 25 ist etwas wenig

Menu im Sender Stick kalibrieren suchen und alle 7 Analogwerte Mitte, Min und Max
bewegen jeweils mit [Menu]

Fertig ist die Kiste!

Th9x Schaltplan der Stromversorgung für Akku laden

Ladestecker: Hohlstecker 5,5x2,1mm Belegung: Innen-Stift = Plus, Aussen-Ring = Minus



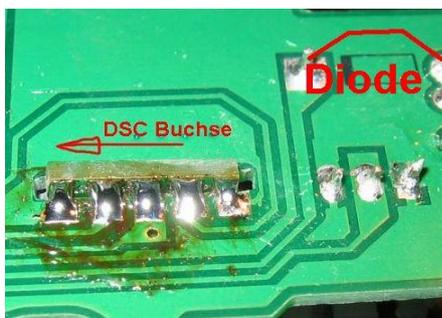
Laden eines Akkusatzes in der Th9x mit modernem Ladegerät:

Moderne Ladegeräte messen zyklisch die Akkuspannung um einen vollen Akku zu erkennen.

Das geht aber nur wenn die Verpolungsschutz-Diode gebrückt wird. (Diode im Plan links von JP201) Die Diode selbst sitzt auf der Rückseitenplatine rechts neben der DSC-Buchse. Auf der Platine ist ein Diodensymbol zu sehen. **Also eine Drahtbrücke einlöten und die Diode überbrücken!**

Falls man mal den Ladestecker verkehrt herum angeschlossen hat passiert gar nichts, denn jedes **moderne** Ladegerät misst zuerst die Akkuspannung bevor es mit dem Laden beginnt und meldet dann, dass der Anschluss verpolt ist. Der Ladevorgang kann auch nur mit ausgeschaltetem Hauptschalter erfolgen (Power OFF). Siehe die grüne Linie und Pfeile.

Verpolungsschutzdioden sind Relikte aus alten Zeiten als man mit primitiven Ladegeräten und kleinen Strömen eine Akkusatz in 14 Std geladen und gehofft hat, dass er vollständig geladen ist.



openTx für Taranis Anleitung Deutsch

Das ist alles ok wenn man einen modernen NiMH Akkusatz verwendet, z.B. einen Sanyo Eneloop oder andere Zellen mit sehr geringer Selbstentladung (der Begriff Eneloop ist geschützt, darum verwendet jeder Akkuhersteller eine andere Bezeichnung für diese Art der Zellen). Sender mit 2,4GHz haben nur noch einen sehr geringen Stromverbrauch, ca 100mA, so dass auch ein Akkusatz mit 2100mAh sehr lange reicht.

Außerdem reicht eine Akku-Spannung von ca 7,5V völlig aus, da der normale 5V Spannungsregler ca 7V Eingangsspannung braucht. Deshalb reichen 7 Zellen NiMH oder 2 Zellen LiFe. Wer den normalen 5V Spannungsregler gegen einen 5V Low Drop Typ ersetzt kommt mit ca 5,5V Akkuspannung aus. Das sind 5 Zellen NiMH oder 2 Zellen LiPo

LiPo oder LiFe haben aber auch noch einen Balanceranschluss. Entweder verwendet man dann eine Zellenanzahl mit eingebautem Balancer und Überlade- und Entladeschutz (Stichwort Rx & Tx Akkusatz), oder baut einen kleinen Balancer mit ins Batteriefach ein, oder lädt den Akku nur gelegentlich extern mit Balancer. Solange der Ladestrom und Entladestrom klein ist driften die einzelnen Zellen nicht weit auseinander. Da passiert nicht viel.

NiMH Akkus sind vollgeladen mit ca 1,27V/Zelle und leer mit ca 1,1V/Zelle

Bei einem 8 Zellen NiMH Akkusatz stellt man die Warnschelle am Sender deshalb auf ca 9,0V ein. LiPo Zellen sind fast leer bei ca 3,3V-3,4V/Zelle, bei 3 Zellen die Warnschelle auf ca 10,5V einstellen, dann hat man noch genügend Zeit. Sender Systemeinstellungen 1/6, Akku leer wenn:

Aufpassen muss man am Anschluss des Akkus am Sender. Wenn man den Original-Stecker vom Batteriehalter verwendet und an den Akku anlötet kann nichts passieren. Allerdings benutzen mache nur einen kleinen 2-poligen Stecker der am Akkupack fertig montiert ist.

Wenn man da nicht höllisch aufpasst geht der Sender beim Einschalten in Rauch auf!

Also markieren, dreimal überlegen und kontrollieren damit der Stecker richtig eingesteckt wird.

Buchsenbelegung am Sender:

Rot = Plus = Mitte Buchse

Schwarz = Minus = links und rechts an der Buchse



Link-Sammlung der Modifikationen

Fertige Modelle und Einstellungen für Taranis

<http://rcsettings.com/>

OpenTX University, die beste Englische Lernseite!!

<http://open-txu.org/>

Die beste Seite über Frsky Baugruppen:

http://www.eflightwiki.com/eflightwiki/index.php?title=FrSky_Telemetry

Immer das aktuellste Handbuch openTX für Taranis als PDF

Suche Dateiname mit aktuellem Datum immer am Ende

<http://openrcforums.com/forum/viewtopic.php?f=92&t=3563&sid=0e96387d744e3cd47282f3a885fa78de>

Opentx, Companion und alle Infos findet man hier:

<http://www.open-tx.org/>

Companion zum Download hier:

<http://downloads-20.open-tx.org/companion/>

OpenTx Firmware als fertige *.bin dateien für alle Sender gibt es hier:

<http://downloads-20.open-tx.org/firmware/binaries/>

Das wichtigste Forum zu openTx und companion

<http://openrcforums.com/forum/index.php?sid=6b9eb378314ef3a5e3cb0f7e811099c2>

LUA Scripte und Informationen

<http://www.open-tx.org/luascripts.html>

<http://www.open-tx.org/2014/06/02/luascripts/>

Immer das aktuellste Handbuch openTX für Taranis als PDF

Suche Dateiname mit aktuellem Datum immer am Ende

<http://openrcforums.com/forum/viewtopic.php?f=92&t=3563&sid=0e96387d744e3cd47282f3a885fa78de>

Viele Infos über OpenTx, Taranis, Programmierung bei FPV community:

<http://fpv-community.de/showthread.php?24783-FrSky-TARANIS-FrSky-neuster-Geniestreich-16-Kanaele-2-4Ghz-openTX-8-Sprachen>

Bei rcgroups gibt es viele Taranis und FrSky Seiten mit vielen Videos:

<http://www.rcgroups.com/forums/showthread.php?t=1914834>

Dort gibt es auch die Frsky-Taranis Seiten:

<http://www.rcgroups.com/forums/showthread.php?t=1866206>

Splashscreens Library für er9x Th9x gibt es hier, kann man anpassen an Taranis:

<http://openrcforums.com/forum/viewforum.php?f=43>

openTx für Taranis Anleitung Deutsch

Für Segelflieger F3F F3J die Spezialseite aus England, fertige Programme, Tips+Tricks

<http://rc-soar.com/opentx/setups/f3f/index.htm>

FPV-Community-Seiten mit eigenen Frsky-Seiten

<http://fpv-community.de/forumdisplay.php?79-FrSky>

Sonstiges: für Th9x 9XR

http://9xforums.com/wiki/index.php/Hardware_Mods_%26_Other_Guides

http://9xforums.com/wiki/index.php/How_to_do_a_full_mod_on_your_9x

http://9xforums.com/wiki/index.php/9x_Full_Mod_Telemetry

http://9xforums.com/wiki/index.php/9x_Full_Mod_FrSky

<http://9xforums.com/forum/viewforum.php?f=9>

<http://9xforums.com/forum/viewforum.php?f=23>

Die Programmierer und das Team von openTx



Wir bedanken uns bei den vielen freiwilligen Helfer und Idealisten der open-source Gemeinde. Hunderte haben mitgewirkt.

Für die aktive, konstruktive Mitarbeit, die Tests, Kontrollen, Korrekturen, Übersetzungen und Anpassungen, für die Vorschläge, Verbesserungen und Erweiterungen, die vielen Ideen die aktiv im 9xforums diskutiert und umgesetzt wurden.

FrSky hat zusammen mit diesem Team Taranis mit openTx entstehen lassen. So etwas gab es bisher noch nicht!

Instructions for building and programming

You'll want to modify the code to your own needs, it is very easy if you know the C language. First to program the microcontroller following the instructions:
[Flashing the 9x](#) by Jon Lowe.

Building from source

Of course you need a cross-compiler to be able to compile the sources. You can use WinAVR for this reason. Just do a search on the internet for WinAVR, it is free.

Use SVN to get sources: `svn checkout http://open9x.googlecode.com/svn/trunk/ open9x`

Put yourself in the src

To compile the standard version: `make`

For version FrSky enter: `make EXT=FRSKY`

From author of the software:

I hope you enjoy the openTx FW!

This is an Open Source project, which means I do not ask for money in return, and you are free to view, download, edit and re-distribute the code under GNU v2 license.

If you have any questions, improvements, or to submit compliments, I would be happy to read either on the official project page: <http://code.google.com/p/open9x/>

Either on the forums 9xgroups: <http://9xforums.com/forum/viewforum.php?f=45>

Specifically for bugs / enhancements: <http://code.google.com/p/open9x/issues/list> openTx

and companion9x are free to use under the GNU License v2.0. I spent (and continue) much time to make this software as good as possible.

OpenTx is free to use under the GNU GPL v2.0 License. Feel free to use, copy and modify it as you wish! If you feel that this software has been beneficial you can show your support by donating to MSF. Please tell-us that you did it and you'll be added to the "donators" list.



http://www.msf.org.uk/support_our_work.aspx

Advertise for this cheap radio to your friends, I bet he will look at you with big eyes exclaiming that you'd better go buy a "real" radio. It does not matter, you will have more money for "real" aircraft!

Yours Bertrand Songis

Wird fortgesetzt.

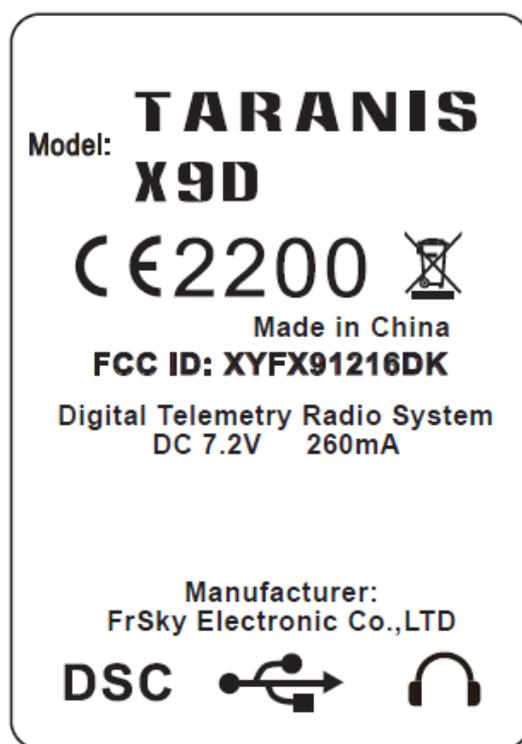
EU-Konformität, CE-Kennzeichnung, FCC-Prüfprotokolle,

**Sender und Empfänger wurden geprüft und entsprechen den EU-Normen.
Die FCC-Prüfprotokolle können von der Homepage von FrSky geladen werden.**

Eine gültige CE-Kennzeichnung ist auf Sender und Empfänger angebracht.

Die Konformitätserklärung stellt der General-Importeur aus und liegt bei.

**Damit ist der Sender Frsky Taranis und die entsprechenden Empfänger
in der EU ohne Einschränkungen zugelassen.**



Konformitätserklärungen für EN 300328 V1.7.1 FrSky-Baugruppen bis 31.12. 2014

DECLARATION OF CONFORMITY

DoC Number: FE20130621

Certifying Organization: FrSky Electronic Co., Ltd.

No. 100 Jinxi Road, Wuxi, Jiangsu, China

Product Description: Digital Telemetry Radio System

Product Model Name: Taranis X9D

We, FrSky Electronic Co., Ltd., declare under our sole responsibility that the above named product(s) conform(s) to all of essential requirements of the European Union Directive **1999/5/EC Radio & Telecommunications Terminal Equipment (R & TTE)**. The conformity assessment procedure referred to in Article 10 and detailed in Annex IV of Directive 1999/5/EC has been followed.

The following harmonized standards and normative documents are those to which the product's conformance is declared, and by specific reference to the essential requirements of Article 3 of the Directive 1999/5/EC.

Article 3.1a

EN 60950-1: 2006 + A11: 2009+A1: 2010+A12: 2011
EN 62311: 2008

Article 3.1b

EN 301 489 -1 V 1.9.2
EN 301 489 -17 V2.1.1

Article 3.2

EN 300 328 V1.7.1

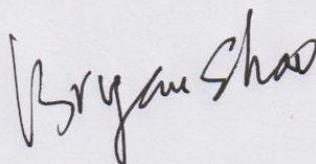
The Technical Construction File (TCF), relevant to the product described above and which support this DoC, are kept at the Certifying Organization stated above.

Name:

Signature:

Title:

Bryan Shao



President

Issue Date:

Jun. 21st, 2013

CE für Sender X9D X9DPlus : Seit 01.01.2015 gelten neue Normen für die HF-Baugruppen

DECLARATION OF CONFORMITY

DoC Number: FF20150129C

Certifying Organization: FrSky Electronic Co., Ltd.

F-4, Building C, Zhongxiu Technology Park, No.3 Yuanxi Road, Wuxi, 214125, Jiangsu, China

Product Description: Digital Telemetry Radio System

Product Model Name: Taranis X9D/ X9D Plus

We, FrSky Electronic Co., Ltd., declare under our sole responsibility that the above named product(s) conform(s) to all of essential requirements of the European Union Directive 1999/5/EC Radio & Telecommunications Terminal Equipment (R & TTE). The conformity assessment procedure referred to in Article 10 and detailed in Annex IV of Directive 1999/5/EC has been followed.

The following harmonized standards and normative documents are those to which the product's conformance is declared, and by specific reference to the essential requirements of Article 3 of the Directive 1999/5/EC.

Article 3.1a EN 60950-1: 2006/A11: 2009/A1:2010/A12:2011
EN 62311:2008

Article 3.1b EN 301 489 -1 V 1.9.2
EN 301 489 -17 V2.1.1

Article 3.2 EN 300 328 V1.8.1

The Technical Construction File (TCF), relevant to the product described above and which support this DoC, are kept at the Certifying Organization stated above.

Name:

Signature:

Title:

Bryan Shao



President

Issue Date:

Jan, 29th, 2015

DECLARATION OF CONFORMITY

DoC Number: FF20150129B

Certifying Organization: FrSky Electronic Co., Ltd.

F-4, Building C, Zhongxiu Technology Park, No.3 Yuanxi Road, Wuxi, 214125, Jiangsu, China

Product Description: 2.4GHz Radio System

Product Model Name: XJT , XFT , XHT , FSD(T) , SXT

We, FrSky Electronic Co., Ltd., declare under our sole responsibility that the above named product(s) conform(s) to all of essential requirements of the European Union Directive 1999/5/EC Radio & Telecommunications Terminal Equipment (R & TTE). The conformity assessment procedure referred to in Article 10 and detailed in Annex IV of Directive 1999/5/EC has been followed.

The following harmonized standards and normative documents are those to which the product's conformance is declared, and by specific reference to the essential requirements of Article 3 of the Directive 1999/5/EC.

Article 3.1a

EN 60950-1: 2006/A11: 2009/A1:2010/A12:2011
EN 62311:2008

Article 3.1b

EN 301 489 -1 V 1.9.2
EN 301 489 -17 V2.1.1

Article 3.2

EN 300 328 V1.8.1

The Technical Construction File (TCF), relevant to the product described above and which support this DoC, are kept at the Certifying Organization stated above.

Name:

Bryan Shao

Signature:



Title:

President

Issue Date:

Jan. 29th, 2015

CE für die Empfänger

DECLARATION OF CONFORMITY

DoC Number: FF20150129A

Certifying Organization: FrSky Electronic Co., Ltd.

F-4, Building C, Zhongxiu Technology Park, No.3 Yuanxi Road, Wuxi, 214125, Jiangsu, China

Product Description: 2.4GHz Receiver

Product Model Name: X8R, XSR, X4R, X6R, X12R, L9R, L12R, S3R, S4R, S6R

We, FrSky Electronic Co., Ltd., declare under our sole responsibility that the above named product(s) conform(s) to all of essential requirements of the European Union Directive 1999/5/EC Radio & Telecommunications Terminal Equipment (R & TTE).

The conformity assessment procedure referred to in Article 10 and detailed in Annex IV of Directive 1999/5/EC has been followed.

The following harmonized standards and normative documents are those to which the product's conformance is declared, and by specific reference to the essential requirements of Article 3 of the Directive 1999/5/EC.

Article 3.1a

EN 60950-1: 2006/A11: 2009/A1:2010/A12:2011
EN 62311:2008

Article 3.1b

EN 301 489 -1 V 1.9.2
EN 301 489 -17 V2.1.1

Article 3.2

EN 300 328 V1.8.1

The Technical Construction File (TCF), relevant to the product described above and which support this DoC, are kept at the Certifying Organization stated above.

Name:

Signature:

Title:

Bryan Shao



President

Issue Date:

Jan. 29th, 2015

CE für Sender X9D, X9D Plus Pultsender 9XE ab 2015 ETSI V1.8.1

DECLARATION OF CONFORMITY

DoC Number: FE20150713

Certifying Organization: FrSky Electronic Co., Ltd.

F-4, Building C, Zhongxiu Technology Park, No.3 Yuanxi Road, Wuxi, 214125, Jiangsu, China

Product Description: Digital Telemetry Radio System

Product Model Name: Taranis X9D, Taranis Plus, Taranis E

We, FrSky Electronic Co., Ltd., declare under our sole responsibility that the above named product(s) conform(s) to all of essential requirements of the European Union Directive 1999/5/EC Radio & Telecommunications Terminal Equipment (R & TTE). The conformity assessment procedure referred to in Article 10 and detailed in Annex IV of Directive 1999/5/EC has been followed.

The following harmonized standards and normative documents are those to which the product's conformance is declared, and by specific reference to the essential requirements of Article 3 of the Directive 1999/5/EC.

Applied / Complied Harmonized Standards	
R&TTE Directive 1999/5/EC, Article 3(1)(a) Health & Safety	EN 60950-1:2006+A11:2009+A1:2010+A12:2011+A2:2013 EN 62479:2010
R&TTE Directive 1999/5/EC, Article 3(1)(b) EMC	EN 301 489-1 V1.9.2, EN 301 489-17 V2.2.1
R&TTE Directive 1999/5/EC, Article 3(2) Radio	EN 300 328 V1.8.1

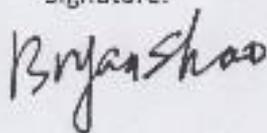
The Technical Construction File (TCF), relevant to the product described above and which support this DoC, are kept at the Certifying Organization stated above.

Name:

Signature:

Title:

Bryan Shao



President

Issue Date:

Jul. 13th, 2015

DECLARATION OF CONFORMITY

DoC Number: FF20150129A

Certifying Organization: FrSky Electronic Co., Ltd.

F-4, Building C, Zhongxiu Technology Park, No.3 Yuanxi Road, Wuxi, 214125, Jiangsu, China

Product Description: 2.4GHz Receiver

Product Model Name: X8R, XSR, X4R, X6R, X12R, L9R, L12R, S3R, S4R, S6R

We, FrSky Electronic Co., Ltd., declare under our sole responsibility that the above named product(s) conform(s) to all of essential requirements of the European Union Directive 1999/5/EC Radio & Telecommunications Terminal Equipment (R & TTE). The conformity assessment procedure referred to in Article 10 and detailed in Annex IV of Directive 1999/5/EC has been followed.

The following harmonized standards and normative documents are those to which the product's conformance is declared, and by specific reference to the essential requirements of Article 3 of the Directive 1999/5/EC.

Article 3.1a

EN 60950-1: 2006/A11: 2009/A1:2010/A12:2011
EN 62311:2008

Article 3.1b

EN 301 489 -1 V 1.9.2
EN 301 489 -17 V2.1.1

Article 3.2

EN 300 328 V1.8.1

The Technical Construction File (TCF), relevant to the product described above and which support this DoC, are kept at the Certifying Organization stated above.

Name:

Signature:

Title:

Bryan Shao



President

Issue Date:

Jan. 29th, 2015

Hinweis zu der neuen EU Sendenorm ETSI V1.8.1

ETSI V1.7.1 (bis 31.12.2014) und ETSI V1.8.1 (ab 01.01.2015)

Seit 01.01.2015 dürfen RC-Sender in der EU nur noch nach ETSI V1.8.1 senden.
Weltweit wird aber weiterhin nach ETSI V1.7.1 konformem Verfahren gesendet.

Das heist: Die HF-Firmware der Sende-Module im Sender und in den Empfängern mussten speziell für die EU geändert werden (wegen $MU \leq 10\%$ und /oder LBT-Verfahren).

Mit V1.7.1 kann Frsky alle X-Empfänger, alle D-Empfänger, alle V-II-Empfänger bedienen
Mit V1.8.1 kann Frsky nur noch alle X-Empfänger bedienen, keine D und V-II Empfänger mehr!

Bestandschutz ist gegeben, niemand muss umrüsten, aber das hilft ja nicht wirklich weiter
Alle Neugeräte die in der EU verkauft werden haben eine HF-Firmware nach V1.8.1 drauf.

V1.7.1 und V1.8.1 passen nicht zusammen!

Sender und Empfänger müssen die gleichen Versionen haben sonst kann man nicht binden.

Wer schon länger mit Frsky arbeitet und dann neue Geräte zukauf, hat dann Sender und Empfänger mit verschiedenen HF-Softwareständen. Alt V1.7.1 und Neu V1.8.1

Was kann man machen wenn man alte und neue X- und D-Empfänger hat:

Die D-Empfänger ersetzen durch X-Empfänger (nah ja?)

Entweder muss man alle Sender und Empfänger auf V1.8.1 umflashen, (geht bei D-Empf nicht)
oder Neugeräte auf V1.7.1 zurückflashen, was eigentlich nicht erlaubt ist, aber technisch geht.

Man kann auch ein externes XJT-Modul mit V1.7.1 verwenden und den Sender auf V1.8.1 lassen.
Damit kann man weiterhin die bisherigen D- und V-II-Empfänger nutzen.

Das externes XJT-Modul überträgt die Telemetriewerte ganz normal auf den Sender,
ohne extra Kabel oder Display (das interne XJT-Modul muss dann abgeschaltet sein)

Software zum umflashen gibt es auf der FrSky Homepage, Download, Firmware

<http://www.frsky-rc.com/download/>

Von Frsky gibt es 3 Varianten für die HF-Software ETSI V1.7.1 oder ETSI V1.8.1

Weltweit als Non EU Versionen, mit ETSI V1.7.1 für alle X-Empfänger, alle D-Empfänger

2 EU-Versionen: Nur am Versions-Datum zu unterscheiden 23.01.2015 und 13.04.2015

EU-Version V1.8.1 X-Empfänger mit V1.8.1, D-Empfänger mit V1.7.1 Version 150123

EU-Version V1.8.1 X-Empfänger, keine D-Empfänger mehr Version 150413

Wer seine D-Empfänger weiterhin nutzen will, kann die passende *.frk Software auf
das Sender HF-Modul und die Empfänger flashen.

Das ist die Software für die HF-Übertragung, die in den Sendemodulen selbst gespeichert ist.

Diese FrSky HF-Software *.frk hat nichts mit der Bedienoberfläche openTx zu tun!

