

Anleitung für Turnigy TGH9x Sender mit 2.4GHz

Vorab, um was geht es:

Stand 0.01 02.08.2012

Es geht um diesen Sender, TH9x, der unter diversen Namen und Marken verkauft wird. Unter anderem bei Hobbyking in Hongkong als Turnigy TH9x für ca 45€ incl 9-Kanal Empfänger, oder bei Jamara als Air Jump3 für 229€. Es ist aber im Grunde immer der selbe Sender nur minimal mit anderer Software und Bedienung.

Er hat trotz aller anderstlautender Meinungen von „Besserwissern, Neidhammeln und Bedenkenträgern“ eine gültige CE-Kennzeichnung mit Konformitätserklärung. Es gibt auch eine Deutsche Kurzanleitung, ausgestellt vom deutschen Hobbyking-Warehouse in Hamburg. Somit steht einem Direktimport und Verzollung auch nichts mehr im Wege. Man kann ihn aber auch direkt bei Hobbyking in Hamburg ordern, dann ist er schon verzollt. Da die Nachfrage weltweit aber sehr sehr groß ist, ist er meist ausverkauft und man muss sich im Backorder eintragen. Wartezeiten 4-8 Wochen sind dann üblich.

Normal ist das ein preiswerter, guter 9 Kanal Sender auf 2,4GHz mit FHSS Frequenz-Hopping-Modul auf 16 Kanälen im 2,4Ghz Band mit Standardsoftware, die ein paar Macken hat aber im großen und ganzen ok ist. Einen Frequenzscann hab ich gemacht und stell ihn noch rein, da sieht man dasss das Ding ganz ok ist..

Der große Vorteil aber ist, dass seine interne Software durch eine andere ersetzt werden kann und damit sehr flexibel und preiswert Funktionen zur Verfügung stehen wie sonst nur in Sendern für 2000€ zu finden sind. Dabei ist die neue Software so flexibel, dass schlichtweg alles möglich was man sich nur vorstellen kann, es gibt faktisch keine Begrenzungen und Einschränkungen! Alles kann allem frei zugeordnet werden.

Auch die Hardware kann relativ einfach erweitert und angepasst werden. So kann das normale 2.4GHz FHSS Sendemodul durch Frsky-Module ersetzt werden und man hat alles an Telemetrie was man sich nur vorstellen kann.

Es gibt verschiedene Entwicklungen von Software für diesen Sender, die alle Open Source sind und ständig angepasst, erweitert und gepflegt werden. Sie bauen alle auf den Grund-Entwicklungen von Thus9x, Er9x, Open9x, Radioclone, Frsky, Gruvin, usw. auf oder sind Mischungen daraus.

Ich habe mich für die Variante **Open9x** entschieden da hier auch eine deutsche Menüführung im LCD Display zur Verfügung steht und das Handling sehr übersichtlich und logisch ist.

Die eigentliche Programierung der Modelle kann statt am Sender auch am PC erfolgen, was sehr praktisch und übersichtlich ist. Dazu gibt es das Programm Companion9x das wiederum auf dem Programm eePe basiert. Am PC können alle Einstellungen für die Modelle und den Sender auch komplett simuliert werden, bevor sie dann in den Sender übertragen werden.

Als gutes Internetforum für diesen Sender, Softwarevarianten, Hardwaremodifikationen ist das 9xforum.com und das FPV-Forum zu nennen, dort findet man alles was man braucht.

Um aber erst mal auf den Sender neue Software aufspielen zu können sind ein paar Hürden zu überwinden. d.h. man muss zuallerst mal eine ISP-Programmierschnittstelle einlöten, damit man auf den Prozessor ATmega64 zugreifen kann. Dann braucht man noch einen USB-Programmieradapter und eine paar Treiber und Programme die das alles bedienen. Die Kosten sind minimal und unter 10€ zu realisieren. Für einen normalen guter Modellbauer und Lötter ist das kein Problem, nur Mut!

Legen wir los.

Nachfolgende Beschreibungen und Erklärungen sind aus meinen Forenbeiträgen entstanden, zusammengefasst, erweitert oder gekürzt. Wenn sie für einen Neuling nicht gleich verstanden werden, so macht das nichts, mehrfach quer lesen und in Ruhe darüber nachdenken bevor man loslegt.

Ein paar Begriffe vorweg:

Flashen heist einen bestimmte Speicherbereich (den Flashspeicher) eines Prozessor mit neuer Software überschreiben, also mit einer neuen Firmware laden. Der Atmega64 hat 64k Flashspeicher 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 und den 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 einer 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 Programmierprogramm das im Hintergrund auf dem PC läuft und den USB-ISP Programmieradater 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.

Vorab:

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 in Weiß, Grün oder Blau mit einem Zwischenstecker für die Platine zum einschleifen.

Meine Hintergrundbeleuchtung hängt aber mit 220 Ohm direkt an 5V, 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 Kurzschuß hat und auch nur ein Vorwiderstand 470-680Ohm für die LED an 9,6-12V Akkuspannung drinnen ist. Und es spart viel Platz.

Unter youtube gibt es dazu ein paar Videos , die das ganz gut darstellen.

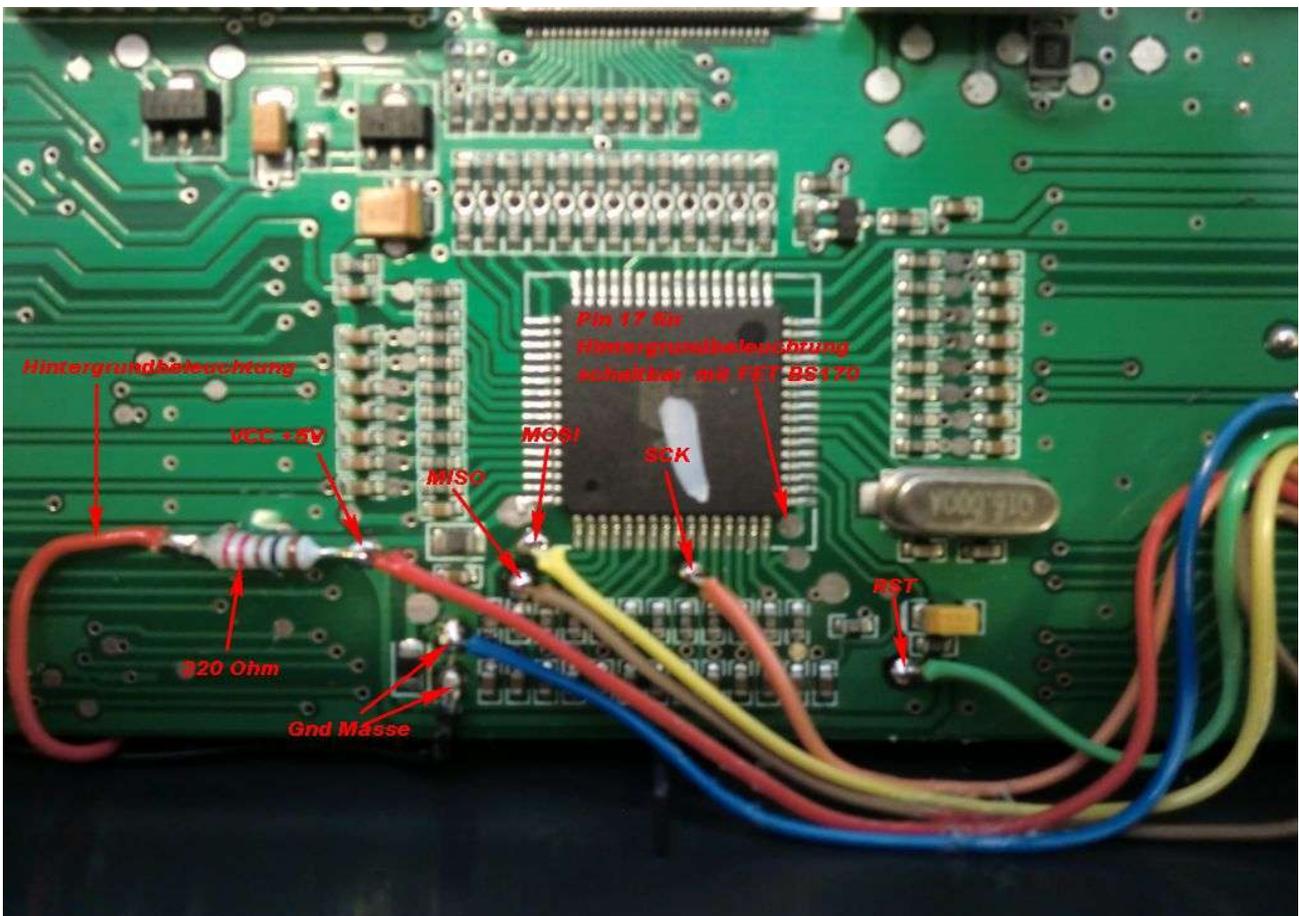
Teil 1 von 4 Einlöten der 6 Leitungen für das Flashen des Senders:

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.

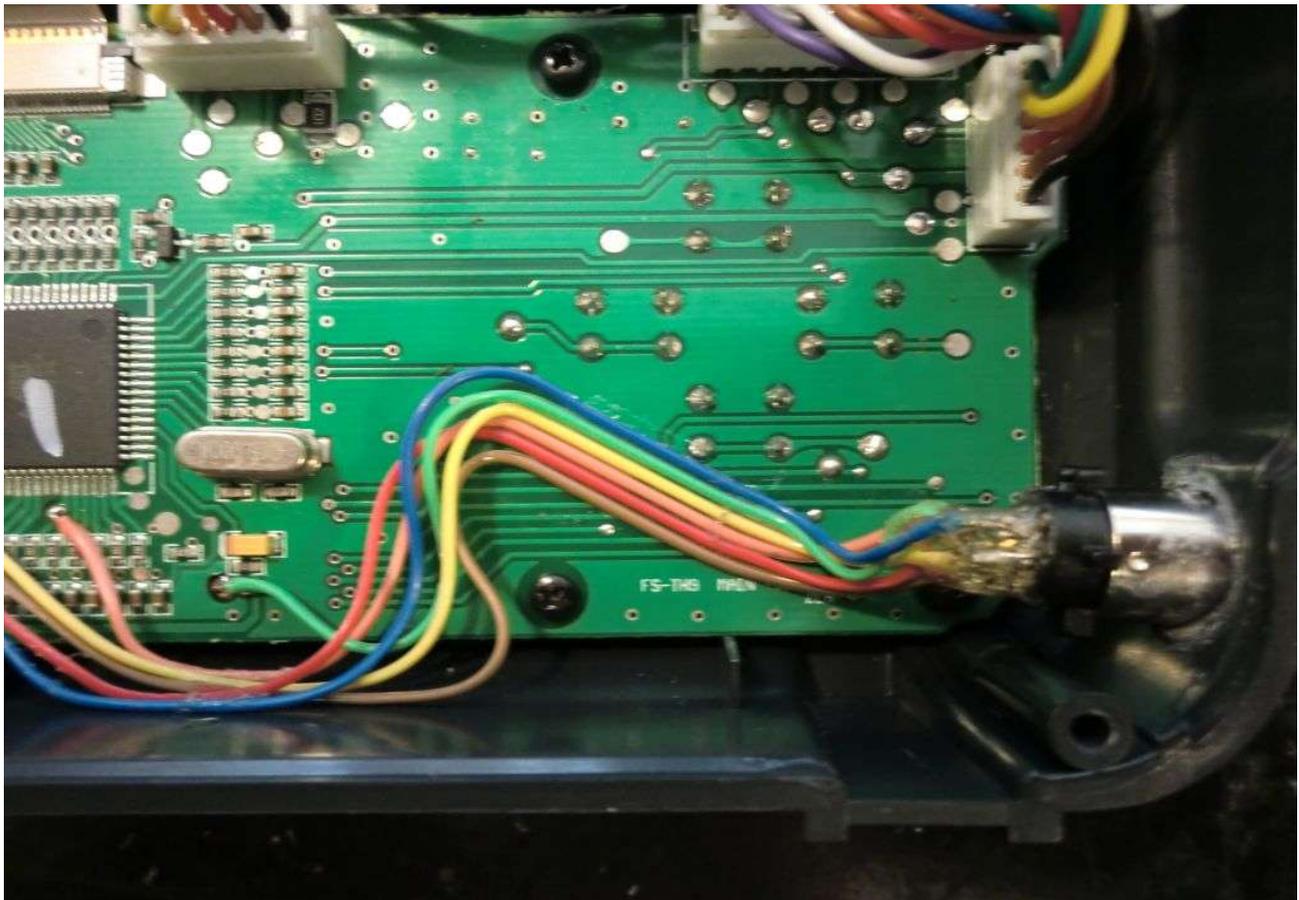
MISO, MOSI, SCK, RST, GND/Masse, VCC +5V

Ich habe das ganze auf eine 6-polige Mini-Din Buchse gelegt und mit einem 10mm Loch seitlich am Gehäuse rausgeführt.

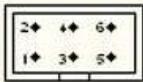
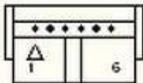
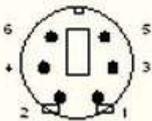
ISP-Anschluss und Hintergrundbeleuchtung einlöten



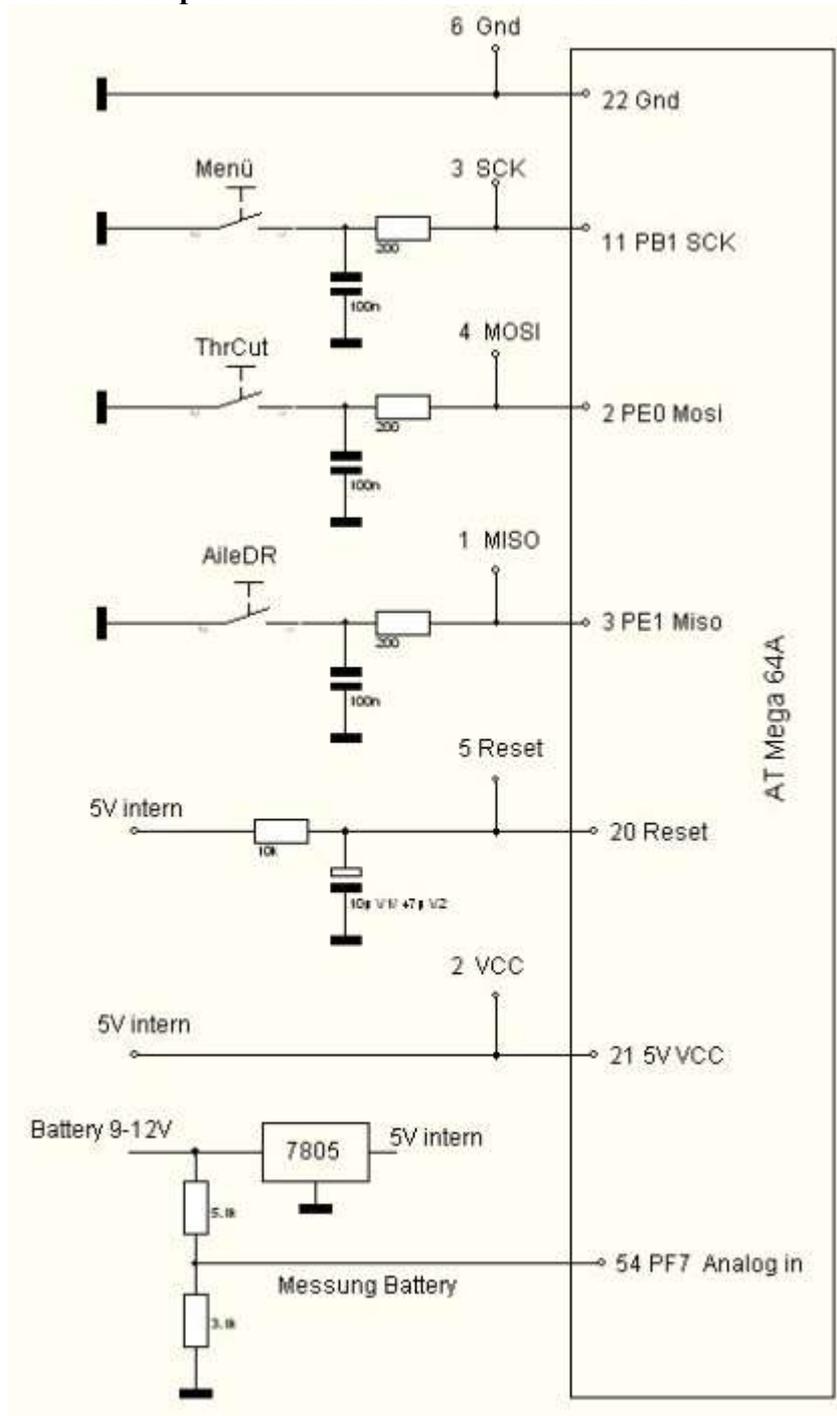
Mini-Din Buche im Gehäuse



Belegung von Stecker, Kabel und Farben

<p>6 Polig Wannenstecker</p>  	<p>MiniDin 6 Polig Buchse</p> 	<p>Sicht auf Buchsenseite = Lötseite des Steckers MiniDin 6 Polig Buchse im Sender einbauen</p> <p>Adapter ca 10cm anfertigen MiniDin 6 Polig Stecker auf 6 Polig Schneidebuchse</p> <p>Farbkodiertes Flachbandkabel verwenden</p> <p>1 braun = MISO 2 rot = VCC 5V 3 orange = SCK 4 gelb = MOSI 5 grün = Reset 6 blau = Gnd</p>						
<p>6 Polig Schneidbuchse</p> <table border="0"> <tr> <td>2 VCC</td> <td>4 MOSI</td> <td>6 Gnd</td> </tr> <tr> <td>1 MISO</td> <td>3 SCK</td> <td>5 Reset</td> </tr> </table>		2 VCC	4 MOSI	6 Gnd	1 MISO	3 SCK	5 Reset	
2 VCC	4 MOSI	6 Gnd						
1 MISO	3 SCK	5 Reset						

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, das sie sonst die Signale MISO und MOSI zu stark belasten.

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.

Rst 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.

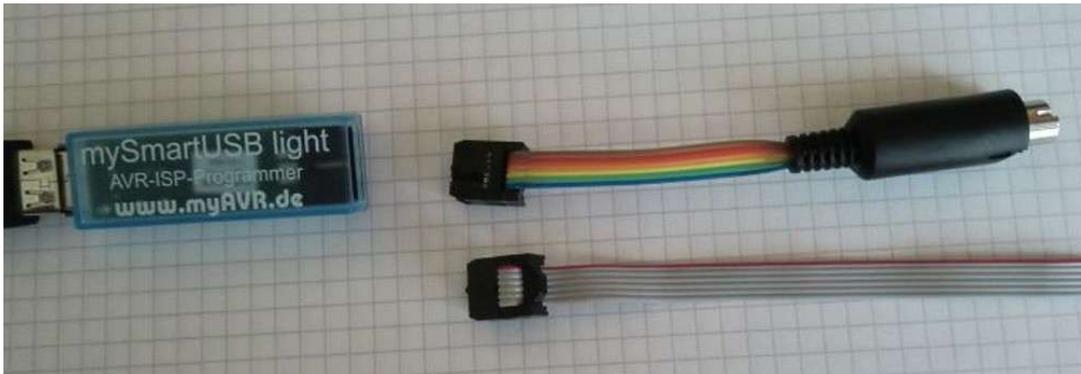
Flashen Teil 2 von 4

Im Sender ist jetzt der ISP-Anschluß eingebaut und auf eine 6 Polige Mini-Din Buchse gelegt.

Jetzt brauchen wir noch eine 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!



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-Programer für ca 4-10€, dann aber entsprechend anpassen.

Für diesen USB Programmer gibt es 2 wichtige Programme:

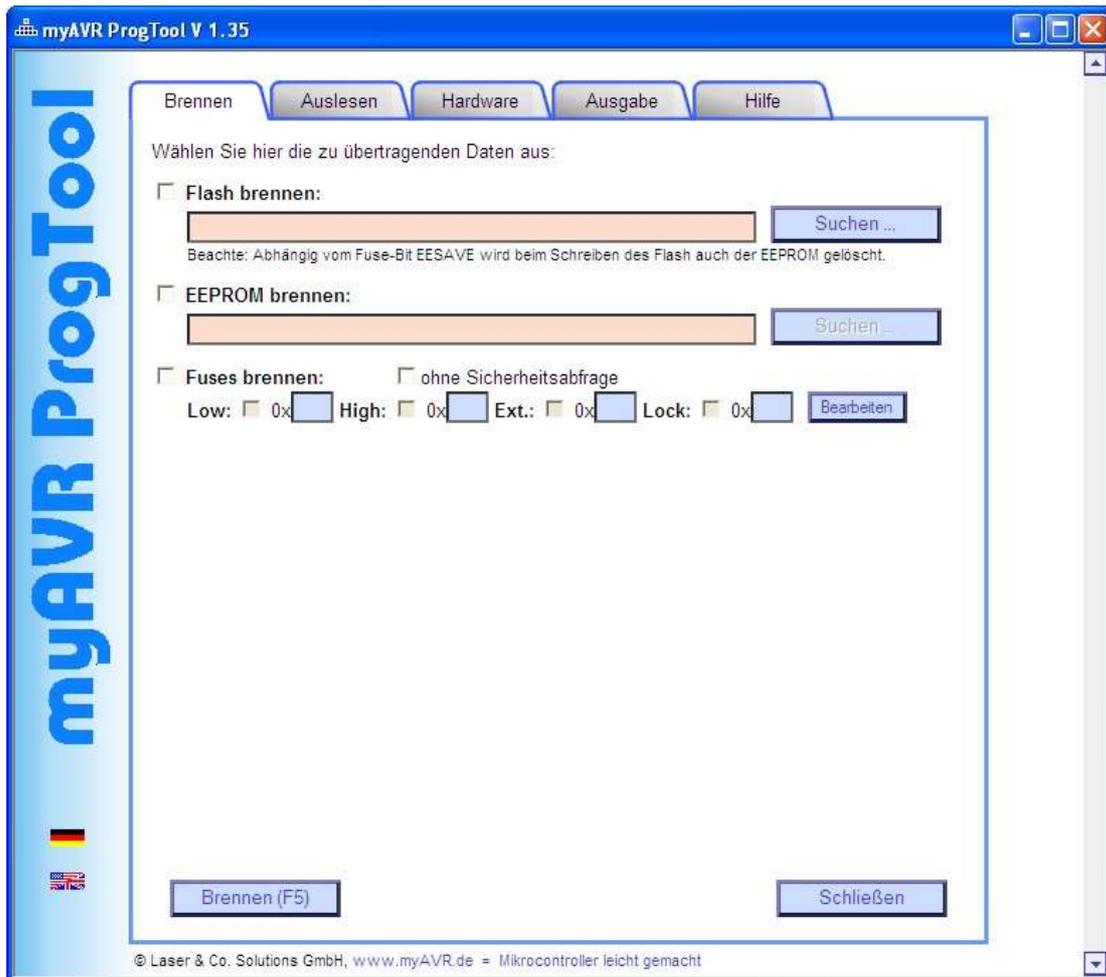
Die SupportBox_MSUL.exe

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 Programmier-Tool
myAVR_ProgTool.exe
um den Flashspeicher mit der Firmware für den Sender zu programmieren
(Open9x oder TH9x oder ER9X)
und das EEPROM für die Einstellungen und Modelle zu programmieren.

Damit könnte man jetzt den Sender schon einfach programmieren indem man die
.Hex oder .Bin Files überträgt, fertig, das wars!



Aber:

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 simmuliert erst mal die Funktionen am PC

Dazu wird das Programm Companion9x benutzt.

Also suchen, downloaden, installieren, deutsch einstellen

Companion9x starten:

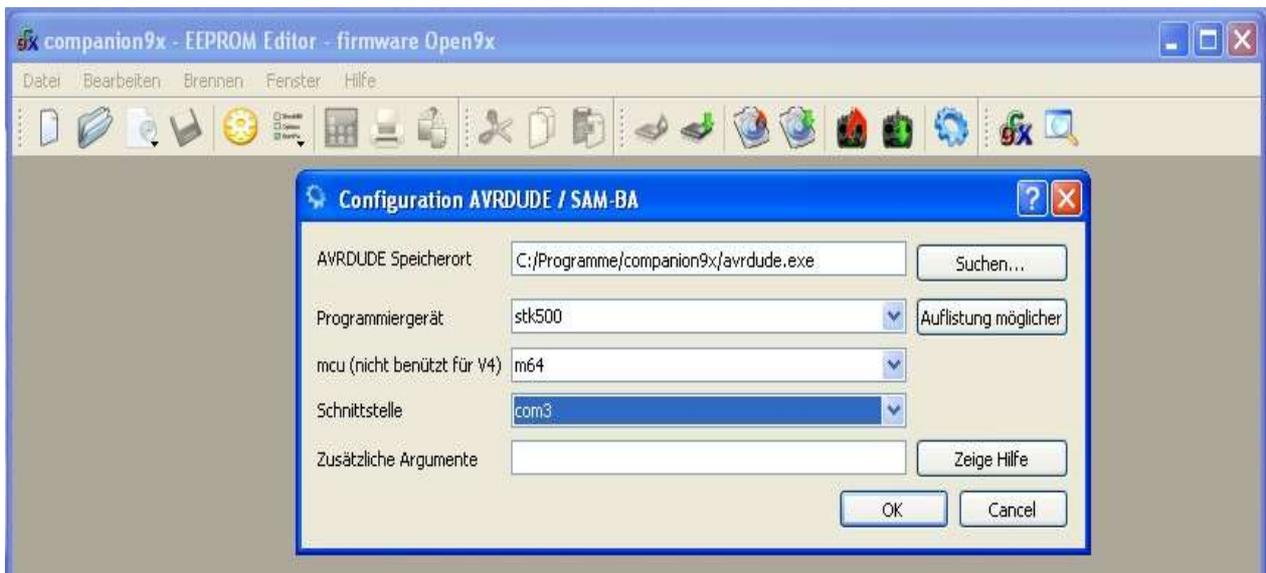
Jetzt muss man darin erst mal 2 wichtige Einstellungen machen

a: für das Brennprogramm und den verwendeten Brenner (z.B. an Com3 als STK500!) damit weiß AVRdude (das eigentliche Brenn-Programm) was er wie wo machen soll und arbeitet mit der BrennHardware mySmartUSB light zusammen!

b: für die Grundeinstellungen der Firmware die erzeugt werden soll.

(Firmware wählen, hier open9x, die Funktions-Module zusammenstellen, den Startbildschirm, usw)

Einstellungen für das Brennprogramm



Grundeinstellungen für Companion9x

Einstellungen [?] [X]

Sprache (Benötigt Neustart) System default language [v]

Überprüfe auf Aktualisierung beim Start Prüfe ausgewählte Fw Updates

Startbild anzeigen HG Beleuchtung blau [v]

Anzahl zuletzt geöffnete Dateien 10 [v] Verhalten Startbild Bibliothek 'companion' Startbilder einschliessen [v] Pilot: 

Benutzer Startbild Bibliothek C:/Programme/companion9x/Bilder [v] Verzeichnis öffnen Pixel invertieren [X]

Benutzer Startbild C:/Programme/companion9x/Bilder/splashscreen-joystick.bmp [v] Bild öffnen

eEprom backup folder C:/Programme/companion9x/Eeprom Daten [v] Verzeichnis öffnen auto backup before write

Firmware open9x für Standard Board [v] Language de [v] Download

<input type="checkbox"/> frsky	<input type="checkbox"/> jeti	<input type="checkbox"/> ardupilot	<input type="checkbox"/> nmea
<input checked="" type="checkbox"/> heli	<input checked="" type="checkbox"/> templates	<input type="checkbox"/> nosplash	<input type="checkbox"/> nofp
<input type="checkbox"/> audio	<input type="checkbox"/> haptic	<input type="checkbox"/> PXX	<input type="checkbox"/> DSM2
<input type="checkbox"/> ppmca	<input type="checkbox"/> potsroll	<input checked="" type="checkbox"/> autoswitch	<input type="checkbox"/> nographics
<input type="checkbox"/> pgbarr	<input type="checkbox"/> imperial		

Letzte geladene Version: 1143 [v] Prüfe für Updates

Ask for flashing after Download

Standard Knüppelmodus Mode 4 (AIL THR ELE RUD) [v] Standardeinstellung Kanalzuordnung R E T A [v]

Profile slot 1 [v] Profile Name Test001 21072012 [v] Save Profile

Joystick aktivieren kein Joystick gefunden [v] kalibrieren

OK Cancel

Simulation in Companion9x

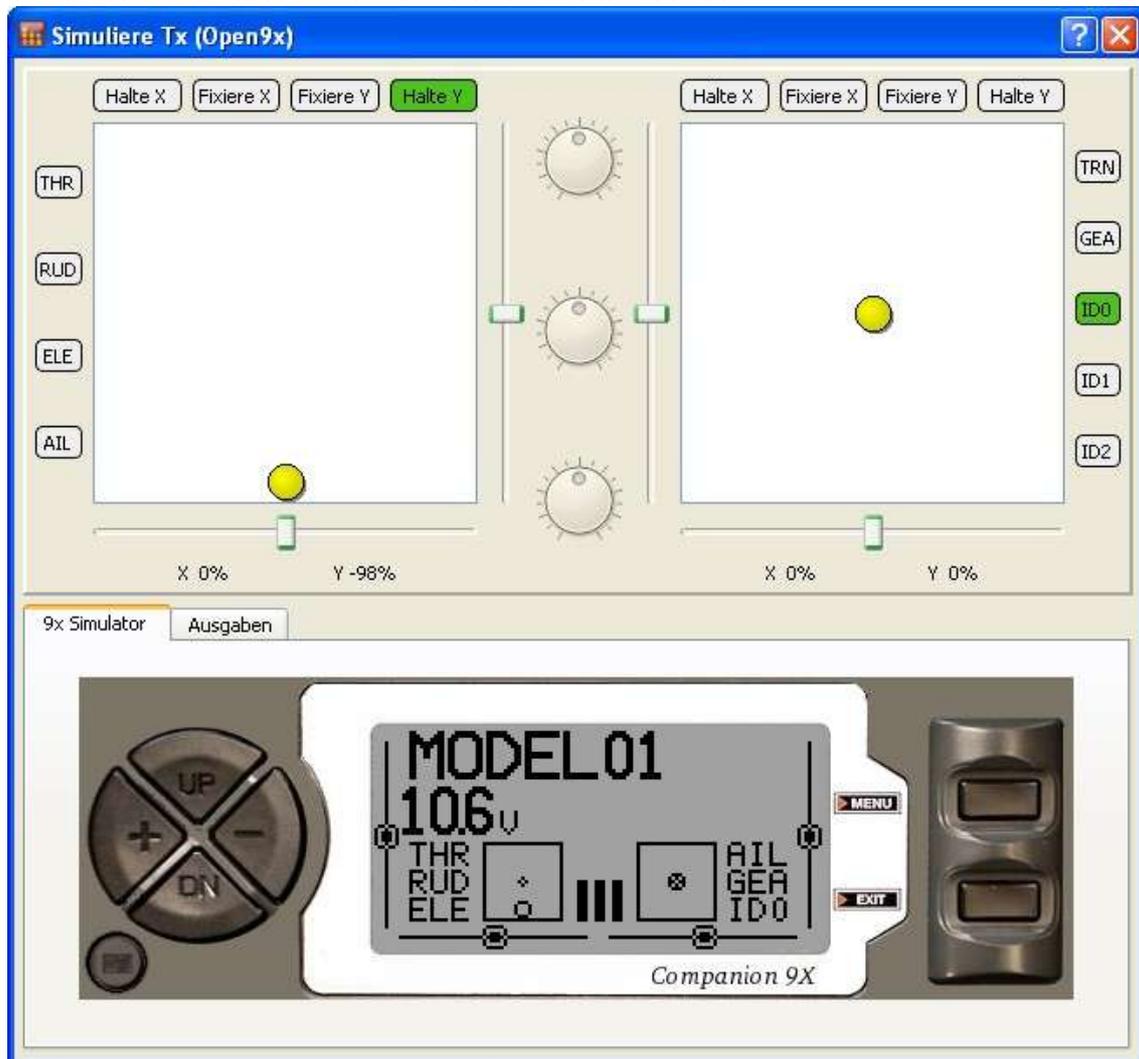
Im Companion9x kann man auch Modelle komplett programmieren und simulieren bevor man die Modell-Daten dann in das EEPROM des Sender flasht.

Wenn das dann alles passt werden 2 Dateien in den Sender übertragen

a: zuerst die eigentliche Firmware (Open9x, thus9x, er9X) in den Flashspeicher des Senders

b: dann die Einstellungen und die Modellparameter in den EEPROMspeicher des Senders

Simulator mit Tasten und Funktionen wie am Sender



Simulation mit den Werten und Kanälen

Ausgaben

SW1	SW2	SW3	SW4	SW5	SW6	Piepston	SW7	SW8	SW9	SW10	SW11	SW12
CH1		●				0.0	0.0			●		CH9
CH2		●				0.0	0.0			●		CH10
CH3		●				0.0	0.0			●		CH11
CH4		●				-14.1	0.0			●		CH12
CH5		●				0.0	0.0			●		CH13
CH6		●				0.0	0.0			●		CH14
CH7		●				0.0	0.0			●		CH15
CH8		●				0.0	0.0			●		CH16

Dualrate und Expo Konfiguration für ein Modell

Konfiguration	Hubschrauber Setup	Phasen	Expos/DR	Mischer	Grenzen	Kurven
Rud	Weg (100%)	Expo (+35%)	Schalter (RUD)			
Ele	Weg (100%)	Expo (+35%)	Schalter (RUD)			
Thr						
Ail	Weg (100%)	Expo (+35%)	Schalter (RUD)			

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>

Companion9x Schreibe Firmware, Einstellungen für Splash Screen



Dann hab ich auch noch 2 Bedienungsanleitungen für den Sender mit drangehängt, Deutsch und Englisch, die recht gut sind

 [ParkeFlyer_Flyer9x_The_Manual_v1.2.pdf](#) (2,61 MB)

 [th9x_Deutsch_Anleitung.pdf](#) (3,26 MB)

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.

Auslesen können wir mit dem Brennprogramm von mySmarLightUSB oder aber auch aus Companion9x, das ist mal egal.

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)

mySmartlight USB an den PC stecken und mit dem Supportbox-Programm so konfigurieren, dass keine Spannung rausgeschaltet wird. Keine 5V und keine 3,3V! Siehe Bilder oben

An den Fuses machen wir gar nichts, nie!!, da lassen wir immer schön die Finger weg!!

Erst jetzt an den Sender anstöpelrn und es passiert erst mal gar nichts!



Dann 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. open9x mit ausgewählten Modulen und Funktionen.

Siehe Companion9x

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, Standardwert 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!

Menühandling von Open9x

Wie man am Sender zu welchen Menüs kommt ist eigentlich ganz einfach und recht logisch aufgebaut. (eigentlich müsste man die Tasten [+] und [-] tauschen, geht aber nicht so einfach!)

Es gibt einen **kurzen** Tastendruck oder einen **langer** Tastendruck!

Menü Grundeinstellungen:

[+] **lang** drücken und man kommt in die Menüs der Grundeinstellungen für den Sender
mit [+] [-] **kurz** werden darin die Seiten 1-6 durchgeblättert
mit [Up] [Dn] **kurz** werden in den Seiten die Positionen/Zeilen angesprungen
in den Positionen/Zeilen mit [+] [-] **kurz** die Werte geändert

mit [Exit] **kurz** Rücksprung zur aktuellen Seite
oder
mit [Exit] **lang** Rücksprung zum Grundbildschirm

Menü Modelleinstellungen:

mit [-] **lang** kommt man in die Modelleinstellungen, aktiv ist immer das Modell mit einem [*]
mit [+] [-] **kurz** werden die Seiten 1-10, 11 oder 12 durchgeblättert
mit [Up] [Dn] **kurz** werden in den Seiten die Zeilen/Positionen angesprungen
in den Positionen mit [+] [-] **kurz** die Werte gändert

oder:

wenn in einer Zeile **mehrere** Werte einzugeben sind,
Positionen mit [+] [-] **kurz** anfahren, mit [Menu] **kurz** anwählen, **Position blinkt** jetzt,
dann mit [+] [-] **kurz** Werte eingeben, mit [Menu] **kurz** bestätigen.

mit [Exit] **kurz** Rücksprung zur aktuellen Seite
oder
mit [Exit] **lang** Rücksprung zum Grundbildschirm

Alles ganz einfach und immer gleicher Ablauf!

Modelle auswählen, kopieren, löschen, verschieben

(Verfahren gilt auch um die Zeilen in den Mischern oder DR/Expo zu bearbeiten!)

Aktiv ist immer das Modell mit dem [*] davor

Modell Auswählen oder neues Modell anlegen:

mit [Up] [Dn] in die Zeilen gehen, mit [Menu] **lang** das Modell auswählen, Stern [*] aktiviert
Wählt man eine leere Zeile und [Menu] **lang** wird ein neues Modell angelegt.

Modell Kopieren:

mit [Up] [Dn] in die Zeilen gehen, mit [Menu] **das** Modell auswählen, Zeile wird invers dargestellt,
mit [Up] [Dn] wird Modell in erste freie Zeile verschoben, mit [Menu] bestätigt oder mit [Exit]
abgebrochen.

Modell Verschieben:

mit **[Up]** **[Dn]** in die Zeilen gehen, mit **[Menu]** das Modell auswählen, Zeile wird invers dargestellt, nochmal mit **[Menu]** Zeile wird umrandet, mit **[Up]** **[Dn]** Zeile verschieben, mit **[Menu]** bestätigen oder mit **[Exit]** abbrechen.

Modell Löschen:

mit **[Up]** **[Dn]** in die Zeilen gehen, mit **[Menu]** das Modell auswählen, Zeile wird invers dargestellt, dann mit **[Exit]** lang kommt Auswahlfenster für Löschen, mit **[Menu]** bestätigen oder mit **[Exit]** abbrechen

Man kann sich merken:

mit **[+]** **[-]** **[Exit]** gibt es jeweils als **kurz** oder **lang**

mit **[Menu]** lang kommt man **in ein Untermenü** der aktuellen Seite (falls vorhanden)
z.B. bei den Mischern, DR/Expo- Grafiken, Kurven einstellen usw.

[Up] **[Down]** laufen nur die Zeilen rauf oder runter egal ob kurz oder lang oder zeigen verschiedene Grundbildschirme an.

Wichtig:

In den Menüs, bei den Auswahlwerten unterscheidet man:

Nur in **Großbuchstaben**, das sind Schalter, Funktionen, Kurven THR RUD ELE AIL GEA ID0

Groß- und Kleinbuchstaben, das sind die 4Sticks und 3Potis Thr, Rud, Ele, Ail, P1, P2, P3

Dann gibt es noch Zusätze wie %, s, m, t und Vorsätze wie !

% prozentuler Wert in Abhängigkeit von etwas z.B. TH%

s bezieht sich auf einen Stick z.B THs

m bezieht sich auf einen Momenten-Schalter z.B. THm

t bezieht sich auf einen Trigger/Auslöser z.B. THt

! bedeutet invers, Not, nicht betätigter Schalter, d.h. in Grundstellung z.B. !RUD !ALE

CH1 bis CH16 das sind die 16 Servoausgänge, die kann man auch frei zuordnen

So wäre z.B. eine Zuordnung für Graupner Belegung

CH1 Thr Gas auf Kanal 1

CH2 Ail Querruder auf Kanal 2

CH3 Ele Höhenruder auf Kanal 3

CH4 Rud Seiteruder auf Kanal 4

CV1-CV8 sind die Kurven, jeweils beliebig als 3, 5, 9, 17-Punkte mit festen X-Werten

z.B. 5-Punkt mit -100 -50 0 +50 +100 , oder jeweils frei definierbaren X/Y-Werten

SW1-SW9, SWA,SWB, SWC das sind 12 Software-Schalter die mit Bedingungen oder Ereignissen oder Verknüpfungen belegt werden können.
z.B. SW6 v1>v2 Rud Ele, SW7 v>ofs Ele 55

FS1-FS16 sind 16 Funktions-Schalter, also eine freie Zuordnung von Hardware- und Software-Schaltern mit Aufgaben z.B. THR SafetyCH16 66 ok TRNm Reset Timer1 SW2 SafetyCH1 35 ok

Bezeichner und Bedeutungen (leider etwas verwirrend, was nun gemeint ist)

Sticks , Potis und Servo (mit Kleinbuchstaben)

Thr= Throttle= Gas

Ail= Aileron=Querruder

Ele Elevator= Höhenruder

Rud= Rudder= Seitenruder

P1, P2, P3 = Potis am Sender

Ch0-Ch16= 16 Servoausgänge

Echte Schalter am Sender (nur Großbuchstaben)

Man darf sich an den Bezeichnungen der Schalter nicht stören, die sind nun mal so angebracht, aber alle kann man komplett frei zuordnen.

TRN= Trainer Schalter/Taster

GEA=Gear=Fahrwerkschalter

AIL= AilDR= Querruder Dualrate Schalter

ELE=EleDR= Höhenruder Dualrate Schalter

RUD=RudDR= Höhenruder Dualrate Schalter

IDO,ID1 ID2= 3POS= CYC1, CYC2,CYC3= 3 Stufenschalter

THR=THR Cut off= Gas Abschaltug Schalter

Virtuelle Schalter die über Verknüpfungen entstehen (nur Großbuchstaben)

CS1-CSC= SW1-SWC=12 Softwareschalter

FW1-FW16=16 Funktionsschalter

Kurven

CV1-CV8 jeweils als 3, 5, 9,17 Punkte-Kurven mit festen oder variablen X-Werten

Flugphasen

FP0-FP4= 5 Flugphasen

Servoeinstellungen

Offset= Mittelstellung = Neutralstellung=Subtrim

Min= nach unten begrenzen

Max=nach oben begrenzen

Timer

TM1,TIMER1, TM2, TIMER2

Trimmwerte von den einzelnen Sticks

TrmT, TrmA, TrmE, TrmR

ABS, Max, Full,

Vorsätze

! = NOT = nicht betätigt = nicht erfüllt = in Grundstellung

m = Momentenschalter (einmal betätigen Ein, nochmal betätigen Aus)

Nachsätze

% = in prozentuale Abhängigkeit von der Stellung eines Wertes

s = Stickwert

t = Triggerereignis,Start

ofs= of static= mit Festwert

v= variabel=Ausgangswert, z.B. Stellung eines Sticks

x= Positionswert eines Sticks/Potis

Vergleichsfunktionen bei Mischern und DR/Expo

Vergleich Variable mit Festwerte (ofs = of static = mit Festwert)

v>ofs größer als ein Festwert Beispiel: v>of Thr 55 = Gasstellung größer 55%

v<ofs kleiner als ein Festwert

|v|>ofs Betrag der Variable größer als Festwert

|v|<ofs Betrag der Variablen kleiner als Festwert

Vergleich mit 2 Variablen

v1>v2 größer Beispiel: Thr>Ale Gasstellung größer als Querruder

v1<v2 kleiner

v1==v2 genau gleich

v1!=v2 nicht gleich

v1>=v2 größer gleich

v1<=v2 kleiner gleich

AND= logisches UND

OR= logisches ODER

XOR= logisches EXOR

+= Werte werden dazuaddiert

*= Werte werden dazumultipliziert

:= Werte werden dazuidividiert

Sonstige weitere Bezeichner (nur ein Auszug!)

Saftey CH, Trianer, Audio, Instant trim, Reset, Vario, Play, Start, Backlight, Haptic

Das sind nur mal die Wichtigsten, dazu gibt es viele weitere Bezeichner und Variablen je nach geladenen Modulen, Funktionen, Telemetrie usw.

Verzögerungszeiten und Servogeschwindigkeiten jeweils getrennt einstellbar!

Es können für jeden Mischer und jeden Kanal getrennte Verzögerungszeiten und getrennte Servogeschwindigkeiten für jeweils Auf und Ab eingestellt werden (jeweils bis 7,5s)

PPM PPM16 PPMSim CH4-16

Der Sender kann so konfiguriert werden dass er 4, 6, 8, 10, 12, 14, 16 Servo-Kanäle ausgibt.

Das Timing für den Impulsrahmen kann beliebig eingestellt werden, normal sind 22,5ms und 300us Pause pro Kanal. Das PPM Signal kann als positive oder negative Signal generiert werden.

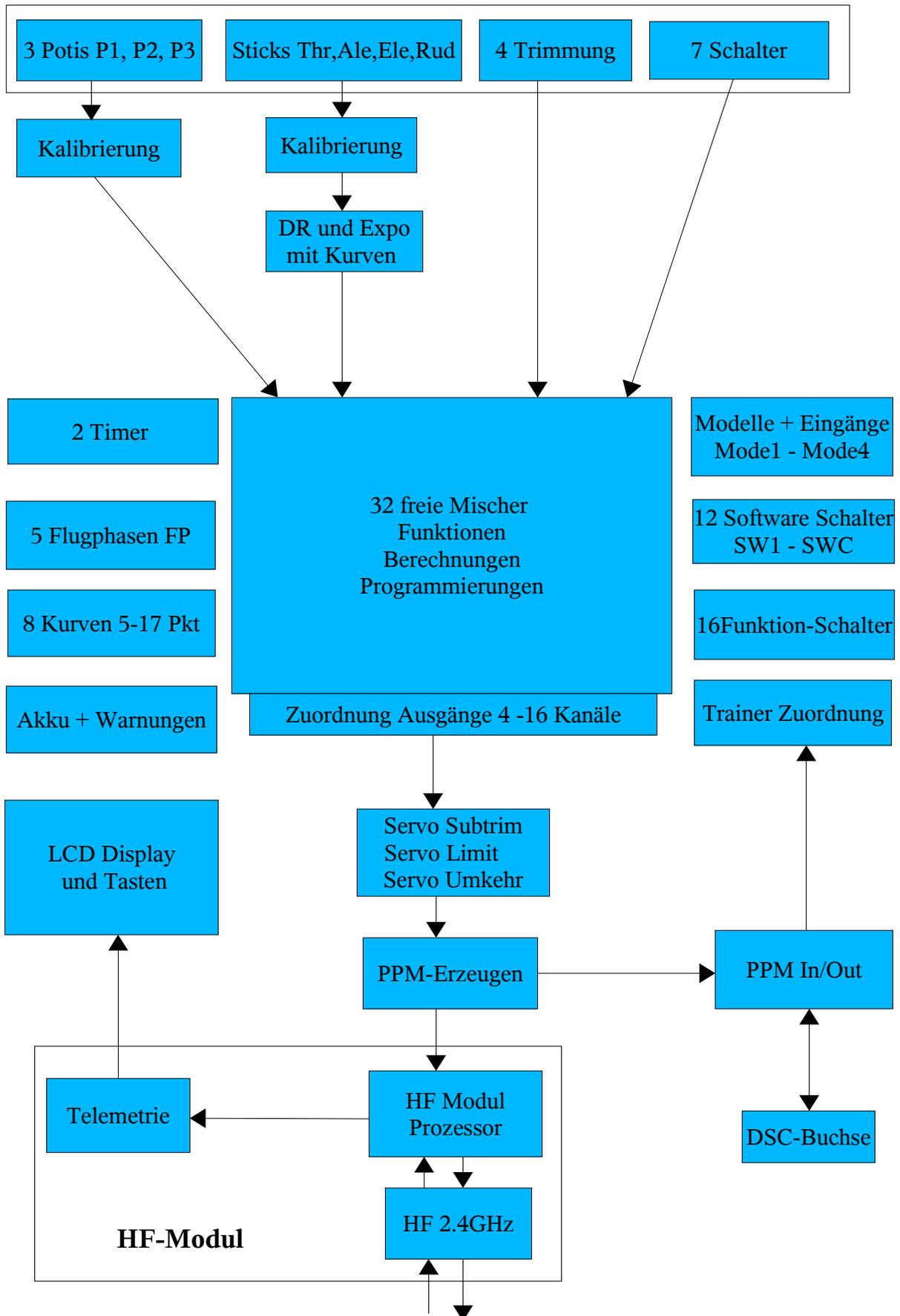
Die Servo-Impulsmitten können frei eingestellt werden, je nach Servo und Hersteller (Graupner, Futaba, Multiplex) sind das 1500, 1520, 1550ms für Mittelstellung des Servos.

Der Sender kann als Schülersender und als Lehrersender verwendet werden, dabei können Kanäle Einstellungen und Übergabewerte frei angepasst werden, selbst wenn der Schüler mit der Belegung Mode 1 und der Lehrer mit Mode 4 fliegt. Die zum Lehrer übertragenen Werte können im Lehrersender auch noch frei skaliert werden, bedingt freigeschaltet oder gesperrt werden.

Der Sender kann auch als Flug-Simulator-Sender (PPMSIM) verwendet werden, auch da ist wieder eine freie Zuordnung von Kanalzahl (CH4-CH16) und Art des PPM positiv oder negativ möglich.

Soweit mal Stand 14.08.2012

Übersicht und Ablaufdiagramm TH9x mit Open9x



Open9X Software r1195 Stand 01.08.2012

Am LCD-Display des Senders kann man alles einstellen.

Bedienung und Handling wie oben erklärt.

Die Menüs sind sehr umfangreich teilweise auch mehr als 7 Zeilen pro Seite, aber man kann sie in 5 Kategorien einteilen.

Einschaltbildschirm

Einschalten und Startbildschirm (Splash Screen) erscheint, dann kommen Warungen vor falschen Stickpositionen und Schalterstellungen

Grundbildschirm

Aufgeteilt in 2 Hälften, obere und unterer Hälfte, mit den 4 Cursorstasten durchschaltbar

2x8 ServoAusgänge als Balken oder als Zahlenwerte

Alle Potis, Sticks, Schalter, alle Softwareschalter, Timerwerte, Statistik des Throttelerwertes, Akkuspannung, Flugphasen, Warnungen, usw darstellbar

Setup der Fernsteuerung [+] lang

6 Seiten mit allen möglichen Grundeinstellungen des Senders, PPM, Kanäle, Signalerzeugung, Trainer, Zuordnungen, Mode, Beleuchtung, Peeper, Debug, Kalibrierung, Trimmreaktionen, AD-Wandler, usw. Teilweise mehr als 7 Zeile pro Seite

Modellauswahl [-] lang

16 Modelle zur Auswahl, Anwählen, Löschen, Kopieren, Verschieben

Modelleinstellungen

11-12 Seiten pro Modell

Modellgrundeinstellungen, Kanäle, Servos, DR/Expo, Grafiken, Limits, Schalter, Funktionen, Kurven, Mischer, Berechnungen, Zeiten, Telemetrie

Teilweise mehr als 7 Zeile pro Seite

und weitere Untermenüs bei DR/Expo, Kurven, Mischern, als Grafiken

Am PC ist das alles viel übersichtlicher mit dem Programm Companion9x

Alles kann dort eingestellt und simuliert werden, bevor es in den Sender übertragen wird.