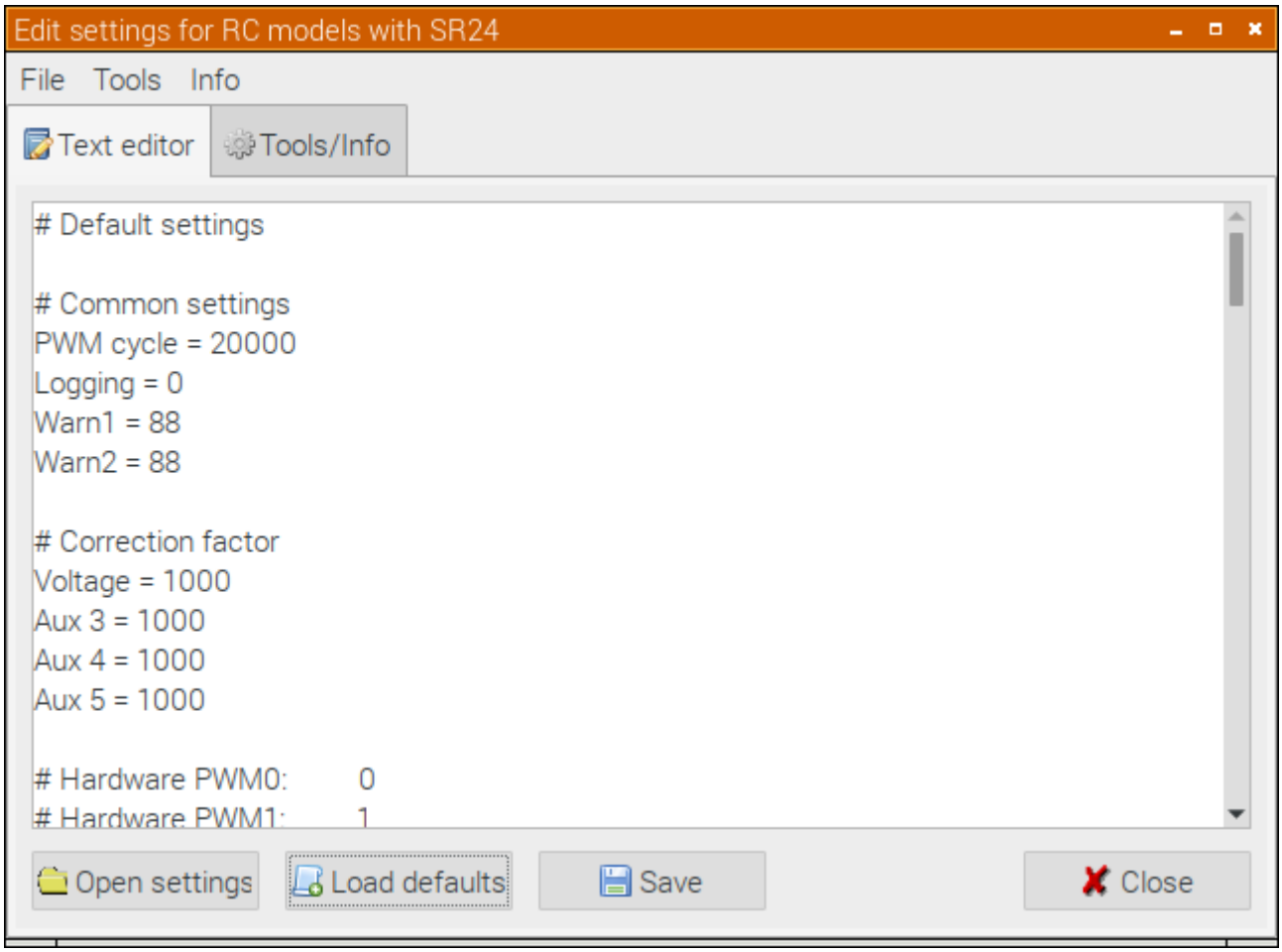
Yuneec ZigBee Receiver SR24 an Raspberry Pi

SR24wizard

Die Settings dienen als Mittler zwischen den empfangenen Daten aus den Kanälen der Funkfernsteuerung ST16 (oder ST24, ST10) und den Ausgängen des Raspberry Pi. Per Settings kann man die Steuerung relativ frei konfigurieren. Es handelt sich um eine einfache Textdatei, die man natürlich auch mit einem normalen Texteditor bearbeiten kann.

Das Wizard-Programm dient dazu, eine Settings-Textdatei zu erzeugen und die Settings zu verwalten.



Auf der zweiten Seite "Tools/Info" sind der Binde-Button und ein paar wichtige Informationen über die Settings zu finden.

Dieses Programm verwendet folgende units :
SR24_chsets: Diese Unit verarbeitet die Konfiguration und vermittelt zwischen Eingang (Kanäle) und Ausgang des Systems. Es wertet die Settings-Datei aus.
SR24_dec: Diese Unit stellt alle Funktionen zum Kontrollieren der Seriellen Schnittstelle und zur Auswertung der empfangenen Daten zur Verfügung. Hier werden nur die Funktionen für das Binden genutzt. Das das Senden einer BIND-Message, welche den Empfänger SR24 in den Bindemodus versetzt, kann natürlich auch im Steuerprogramm implementiert werden.

Die Settings

Um den verschiedenen Servos und Schaltausgängen Kanäle und Ausgabewerte zuzuordnen, werden alle relevanten Daten in einer Textdatei gespeichert. Damit ist es möglich, sein Modell zu konfigurieren.

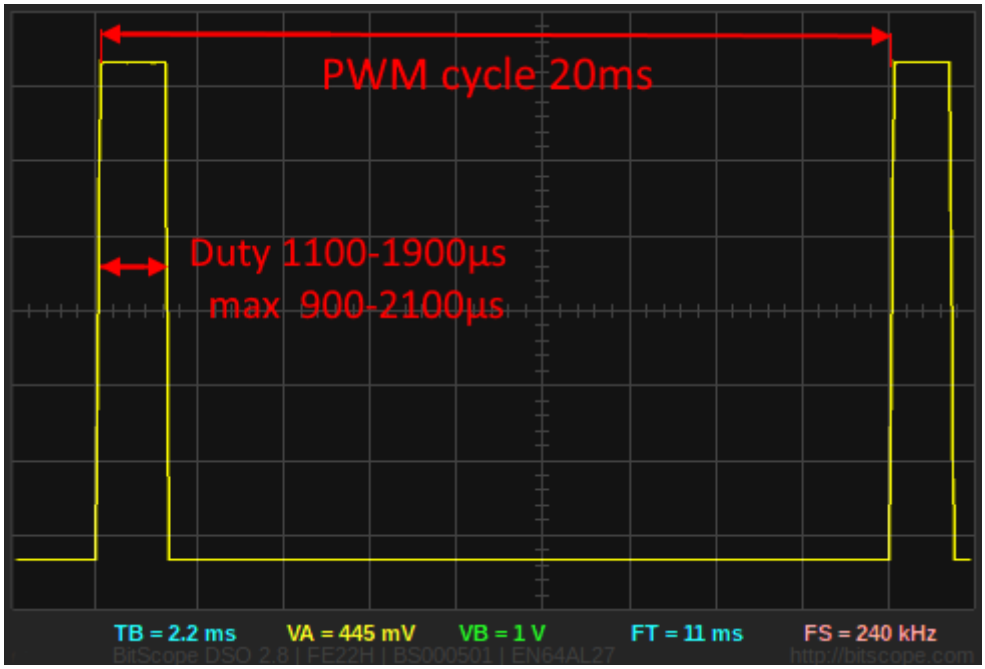
Standard-Kanäle der ST16: Durch Channel Settings und Mode-Wechsel in der ST16 könnten diese Zuordnungen aber auch anders sein. Die Nummerierung der Kanäle startet hier immer mit Eins (nicht wie in der Telemetrie mit Null).

Channel 1 - Throttle THR (Default: 683=-100% bis 3412=+100%, Neutral ist 2048=0%)
Channel 2 - Roll AIL
Channel 3 - Pitch ELE
Channel 4 - Yaw RUD
Channel 5 - Flight mode
Channel 6 - RTH
Channel 7 - Camera tilt
Channel 8 - Camera pan*
Channel 9 - Gimbal tilt mode*
Channel 10 - Gimbal pan mode*
Channel 11 - Landing gear* (Werte für Schalter können zwischen 0 und 4095 liegen)
Channel 12 - Aux button*
Start/stop - Mixed mit Throttle = 0

* nicht bei ST10

Diese Kanäle kann man nun den Ausgängen des Raspberry Pi zuordnen.
Es stehen zwei Hardware-PWM Kanäle für Servos oder Motorsteuerung zur Verfügung:
GPIO18 (pin12) PWM0
GPIO13 (pin33) PWM1

Servo output – PWM wird üblicherweise zwischen 1100-1900µs genutzt. Maximal ist es bei Analogservos 900-2100µs. Die PWM Impulse werden alle 20ms gesendet.



Von der ST16 kommen folgende Werte:
untere Stellung der Sticks: 683=-100%
Mittelstellung: 2048=0%
obere Stellung der Sticks: 3412=+100%

Für Schalter gibt es verschiedene Werte zwischen 0 und 4095 (entspricht -150% bis +150%). Was wie bei der ST10/16 zugeordnet ist, kann man im HW-Monitor oder im Output-Monitor auf der Fernsteuerung sehen.

Für Schalter stehen die folgenden GPIO-Ports zur Verfügung:

GPIO 5	Pin 29	
GPIO 6	Pin 31	
GPIO 12	Pin 32	
GPIO 16	Pin 36	
GPIO 17	Pin 11	
GPIO 22	Pin 15	
GPIO 23	Pin 16	Reserviert für „Voltage warning 1“ als Input
GPIO 24	Pin 18	Reserviert für „Voltage warning 2“ als Input
GPIO 25	Pin 22	
GPIO 26	Pin 37	
GPIO 27	Pin 13	

Die Ports können Schaltern oder auch Sticks zugeordnent werden. Werden Sticks digitalen Ausgängen zugeordnet werden, dann schalten sie bei etwa einem Drittel Servoweg um.
Die anderen GPIO ports sollten für spezielle Anwendungen freibleiben.

Gemeinsam genutzte Settings: Hier gibt es zur Zeit nur die Impulsfrequenz für die beiden Hardware-PWM Kanäle. Diese beträgt 20000µs, das sind 20ms und das (noch nicht realisierte) Logging. Logging bedeutet, dass auf der SD-Karte des Raspberry Pi Logdaten im gleichen Format wie bein den Yuneec-Koptern geschrieben werden. Diese können dann mit den Flightlog auf der St16 verglichen werden.
0 … Default – kein Logging
1 … Nur Telemetry. Die Spalten sind mit Default-Werten aufgefüllt. GPS-Daten stammen von der ST16.
2 … Telemetry und Remote. Hier sehen wir, was auf den Kanälen angekommen ist.
3 … Telemetry, Remote und RemoteGPS. In RemoteGPS werden die GPS-daten der ST16 gespiegelt.

Warn1 und Warn2 sind frei zuordenbare GPIO-Eingänge. Hier sollten Schmitt-Trigger mit hardwaremäßigen Schwellwerten für Voltage Warning 1 und 2 angeschlossen sein. 88 heißt: Nicht genutzt. Standardmäßig sind GPIO23 und 24 dafür vorgesehen.

```
PWM cycle = 20000
Logging = 0
Warn1 = 88
Warn2 = 88
```

Korrekturfaktoren: Korrokturfaktoren sind Ganzzahlen zum Umrechnen von analogen Werten aus Analog-Digital-Wandlern (ADC). Hier kann man Spannung, Ströme. Temperaturen usw. messen und übergeben. ADC-Auslesen ist noch nicht realisiert. Die Werte sind Platzhalter.

```
Voltage = 1234
Aux 3 = 1000
Aux 4 = 1000
Aux 5 = 1000
```

Servos: Dann kommt in der Textdatei die Zuordnung der Servos und Schalter zu den Kanälen der ST16 auf der Eingangsseite und den Raspberry Pi Ports auf der Ausgangsseite. Wir haben zu vergeben:

Eingänge:
Channel Nummer der ST16: 1 … 12

Servo 1 .. 6 sind die vier Sticks und die Tilt und Pan Steller.

Servowege in µs: Min- Neutral – Max. Die Werte von der ST16 werden proportional umgerechnet. Neutral muss nicht in der Mitte sein, es können auch 'Verzerrungen' eingegeben werden (z.B. spezielle Mittelstellungen bei Motorsteuerung).

Ausgänge:
Hardware PWM0: 0
Hardware PWM1: 1
GPIO Portnummer (BCMxy): xy (siehe Liste oben)
Keine Zuordnung (Platzhalter): 88

PWM Zyklen können invertiert werden. Dies scheint aber keine Funktion bei meinen Analogservos zu haben.
Wenn die Servos nicht den beiden Hardware-PWM-Kanälen zugeordnet sind, können sie als Schwellwertschalter genutzt werden. Dann sollte an dieser Stelle GPIOnr 2 stehen.

Beispiel für Throttle-Stick and HW-PWM Kanal 0:
```
[Servo 1]
Channel = 1
Minimum = 900
Neutral = 1500
Maximum = 2100
GPIOnum = 0
PWM reverse = False
```

Beispiel für Yaw-Stick als 3-Wege-Schalter:
```
[Servo 6]
Channel = 4
Minimum = 1100
Neutral = 1520
Maximum = 1900
GPIOnum = 17
GPIOnum2 = 27
```

Werden die Servos digitalen GPIO Ports zugeordnet erfolgt die Umschaltung bei etwa einem Drittel des Servowegs.

Switches: In der nächsten Sektion erfolgt die Zuordnung der Schalter. GPIOnum 2 ist ein zweiter Port für 3-Wegeschalter. Er ist der unteren Position des Schalters zugeornet.

Beispiel Flightmode-Schalter:

```
[Switch 1]
Channel = 5
Upper  position = 3412
Middle position = 2048
Lower  position = 683
GPIOnum = 17
GPIOnum 2 = 27
```

Beispiel Fahrwerk-Schalter:

```
[Switch 4]
Channel = 11
Upper  position = 4095
Middle position = 4095
Lower  position = 0
GPIOnum = 25
GPIOnum 2 = 88
```