

# RC Quad-D

## ***Kran- und Sonderfunktionssteuerung***

Wilhelm Meier

Version 0.1, 15.02.2021: HW\_0.1

# Inhalt

1. Vorwort .....	1
2. Symbolerklärung .....	2
3. Rechtliches .....	2
4. Sicherheitshinweise .....	2
5. Spezieller Sicherheitshinweis: Kleinteile .....	3
6. Einführung .....	4
6.1. Weitere Möglichkeiten .....	4
6.2. Ansicht .....	5
6.3. Grundsätzliches .....	7
6.3.1. Bedienung .....	7
6.3.2. Konfiguration .....	8
7. Software für OpenTX für den Betrieb von <b>RC-Quad-D</b> .....	10
7.1. Installation der Software auf dem Sender .....	11
7.1.1. Unterscheidung der verschiedenen Sendermodelle .....	11
7.2. Konfiguration des Modells (Sender mit <b>480x272</b> -Pixel <i>Farbdisplay</i> , Typ-2) .....	11
7.2.1. Widget für die Bedienung .....	11
7.2.2. Widget für die Konfiguration .....	11
7.2.3. Mixer-Script .....	12
7.2.4. Definition eines Übertragungskanals für das <b>RC-Quad-D</b> am Empfänger .....	12
7.2.5. Definition der Geber (Inputs) .....	13
7.2.6. Spezielle Konfiguration für Sender mit Joystick (FrSky X12s) .....	16
7.2.7. Flugphasen .....	18
7.3. Konfiguration des Modells (Sender mit <b>128x64</b> -Pixel oder <b>212x54</b> -Pixel <i>monochrome Display</i> : Typ-1) .....	19
7.3.1. Telemetrie-Seiten .....	19
7.3.2. Mixer-Script und Übertragungskanal .....	20
7.3.3. Geber und Flugphasen .....	20
7.4. Konfigurationsdateien .....	20
7.4.1. Die Datei <b>MODELS/swstd.lua</b> und weitere Dateien .....	20
7.4.2. Das Widget <b>WMSW/main.lua</b> .....	23
7.4.3. Das Widget <b>WMSWC/main.lua</b> (Konfiguration) .....	24
7.5. Physische Schalter und die Schaltzustände .....	25
7.5.1. ShortCuts / Abkürzungen .....	25
7.5.2. Overlays / Ebenenumschaltung .....	25
8. Einbau .....	26
8.1. Autoerkennung: IBus oder SBus .....	26
8.2. Anschluß an den Akku .....	26
8.3. Schutz .....	26
9. Erste Einrichtung und Anlernen .....	27
9.1. Auto-Erkennung von <b>IBus</b> und <b>SBus</b> .....	27
9.2. Kanal und Adresse .....	27

10. Funktion .....	29
10.1. Bedienung der Schaltfunktionen .....	29
10.2. Konfiguration .....	29
11. Details: Einstellungen im Einzelnen .....	31
11.1. Betrieb <i>ohne</i> Abschaltung .....	31
11.2. Betrieb <i>mit</i> Abschaltung .....	31
11.2.1. Abschaltung durch Endlagenkontakte .....	31
11.2.2. Abschaltung durch Stromüberwachung .....	32
11.3. Proportionaler Betrieb .....	34
11.4. Telemetrie .....	34
11.4.1. Funktionsströme .....	34
11.4.2. Funktionszustände .....	35
12. Betrieb .....	36
13. Fehlersuche und Tests .....	38
13.1. Checkliste <b>IBus</b> .....	38
13.2. Checkliste <b>SBus</b> .....	39
14. Veraltete Informationen .....	41
Appendix A: Einbau .....	42
A.1. IBus oder SBus .....	42
A.2. Anschluß an den Akku .....	42
A.3. Schutz .....	42
15. Kontakt .....	44

# 1. Vorwort



## Lizenz

Dieses Werk ist unter einer Creative Commons Lizenz vom Typ Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International zugänglich. Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie <http://creativecommons.org/licenses/by-nc-nd/4.0/> oder wenden Sie sich brieflich an Creative Commons, Postfach 1866, Mountain View, California, 94042, USA.



Das Modul wie auch diese Doku ist noch unvollständig und *work-in-progress*. Bei jeglichen Unklarheiten in dieser Funktionsbeschreibung und generellem Aufbau und Anschluß, unterlassen Sie den Betrieb und kontaktieren Sie den Bausatzersteller.

## 2. Symbolerklärung



Ein wichtiger allgemeiner Hinweis für den sicheren Aufbau und die sichere Bedienung. Dieser sollte durch den Anwender beachtet werden, um einen sicheren Betrieb zu gewährleisten.



Ein genereller Hinweis, der durch den Anwender beachtet werden sollte.



Ein technischer oder sicherheitstechnischer Hinweis, der unbedingt durch den Anwender beachtet werden muss.



Ein technischer oder sicherheitstechnischer Gefahrenhinweis, der unbedingt durch den Anwender beachtet werden muss. Zur Gefahrenabwendung muss der Anwender unbedingt die gegebenen Anweisungen befolgen und die beschriebenen Maßnahmen ergreifen.

## 3. Rechtliches

Der vorliegende Bausatz wird dem Anwender für eigene Experimente überlassen. Er stellt kein Produkt im Sinne des ProdHaftG oder elektronisches Gerät im Sinne des ElektroG dar und wird als Gerät nicht kommerziell vertrieben.



Die Überlassung gegen Unkostenerstattung erfolgt unter Ausschluss jeglicher Sachmangelhaftung.

Für den vorliegenden Bausatz werden keine Funktionsgarantien gegeben. Für Schäden am Bausatz oder an damit verbundenen Geräten oder Modulen wird keine Haftung übernommen. Gewährleistungen, Garantien und Widerrufsrechte gibt es nicht.

## 4. Sicherheitshinweise

Beim Umgang mit Produkten, die mit elektrischer Spannung in Berührung kommen, müssen die gültigen VDE-Vorschriften beachtet werden, insbesondere VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860.

Werkzeuge dürfen an Geräten, Bauteilen oder Baugruppen nur benutzt werden, wenn sichergestellt ist, dass die Geräte von der Versorgungsspannung getrennt sind und elektrische Ladungen, die in den im Gerät befindlichen Bauteilen gespeichert sind, vorher entladen wurden.

Spannungsführende Kabel oder Leitungen, mit denen das Gerät, das Bauteil oder die Baugruppe verbunden ist, müssen stets auf Isolationsfehler oder Bruchstellen untersucht werden. Bei Feststellen eines Fehlers in der Zuleitung muss das Gerät unverzüglich aus dem Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist. Bei Einsatz von Bauelementen oder Baugruppen muss stets auf die strikte Einhaltung der in der zugehörigen Beschreibung genannten Kenndaten für elektrische Größen hingewiesen werden. Wenn aus einer vorliegenden Beschreibung für den nicht gewerblichen Endverbraucher nicht eindeutig hervorgeht, welche elektrischen Kennwerte für ein Bauteil oder eine

Baugruppe gelten, wie eine externe Beschaltung durchzuführen ist oder welche externen Bauteile oder Zusatzgeräte angeschlossen werden dürfen und welche Anschlusswerte diese externen Komponenten haben dürfen, so muss stets ein Fachmann um Auskunft ersucht werden. Es ist vor der Inbetriebnahme eines Gerätes generell zu prüfen, ob dieses Gerät oder Baugruppe grundsätzlich für den Anwendungsfall, für den es verwendet werden soll, geeignet ist!

Im Zweifelsfalle sind unbedingt Rückfragen bei Fachleuten, Sachverständigen oder den Herstellern der verwendeten Baugruppen notwendig!

Bitte beachten Sie, dass Bedien- und Anschlussfehler außerhalb unseres Einflussbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen. Bei Installationen und beim Umgang mit Netzspannung sind unbedingt die VDE-Vorschriften zu beachten. Geräte, die an einer Spannung über 35 V betrieben werden, dürfen nur vom Fachmann angeschlossen werden. In jedem Fall ist zu prüfen, ob der Bausatz oder die Platine für den jeweiligen Anwendungsfall und Einsatzort geeignet ist bzw. eingesetzt werden kann.

Derjenige, der eine Schaltung oder einen Bausatz aufbaut und fertigstellt oder eine Baugruppe durch Erweiterung bzw. Gehäuseeinbau betriebsbereit macht, gilt nach DIN VDE 0869 als Hersteller und ist verpflichtet, bei der Weitergabe des Gerätes alle Begleitpapiere mitzuliefern und auch seinen Namen und Anschrift anzugeben. Geräte, die aus Bausätzen selbst zusammengestellt werden, sind sicherheitstechnisch wie ein industrielles Produkt zu betrachten.

Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller sondern der Betreiber verantwortlich. Bitte beachten Sie, dass Bedien- und/und Anschlussfehler außerhalb unseres Einflußbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.

Jegliche Vorschriften und Vorsichtsmaßnahmen im Umgang mit elektrischen Komponenten sind vom Anwender einzuhalten.

Beachten Sie ebenfalls die Richtlinien unter [Abschnitt 12](#).

## 5. Spezieller Sicherheitshinweis: Kleinteile



ACHTUNG: Der Bausatz enthält verschluckbare Kleinteile. Von Kindern fernhalten.

## 6. Einführung

Der **RC Quad-D** ist ein digitales Steuermodul (4-Kanal) zum Betrieb mit einem **OpenTX**-Sender, vorzugsweise eine **Radiomaster TX16s** oder anderen sog. *color-LCD* Sendern mit **OpenTX** wie etwa eine **FrSky Horus X12s**. Im Gegensatz zu alten, im analogen Zeitmultiplex-Verfahren arbeitenden Schaltmodulen, wird bei dem **RC Quad-D** die Information für die zu steuernde Funktion vollständig digital übertragen. Daher ist ein störungsfreier Betrieb gewährleistet.

Der **RC Quad-D** hat 4 Ausgänge, an denen 4 Motoren (bis 24V/3A) angeschlossen werden können. Jeder Ausgang kann individuell gesteuert werden. Man kann die Motoren getrennt vorwärts/rückwärts laufen lassen. Die Drehzahl ist wie bei einem Fahrtensteller für vorwärts/rückwärts getrennt einstellbar. Alternativ kann auch jeder Ausgang proportional über einen Übertragungskanal gesteuert werden (dann hat man bis zu 4 Fahrtensteller in einem Modul).

Um Schäden an den Motoren bzw. an den zu steuernden Funktionen zu vermeiden, können die Motoren beim Erreichen ihrer jeweiligen Endlage abgeschaltet werden. Diese Abschaltung erfolgt entweder durch eine Stromüberwachung oder durch Endschalter. Die jeweiligen Abschaltströme können durch das Kranmodul gelernt bzw. vom Anwender eingestellt und angepasst werden. Dies gilt auch für den Proportionalbetrieb.

Die jeweiligen, vom Modul gemessenen Ströme der 4 Ausgänge werden per *Telemetrie* an den Sender übertragen. Auch werden die *Zustände* der einzelnen Ausgänge zur Anzeige auf dem Sender per Telemetrie übertragen. Damit kann man am Sender genau erkennen, ob der Motor bspw. frei läuft, ausgeschaltet ist, sich in einer Endlage befindet oder wegen Überstrom abgeschaltet wurde.

Weiterhin kann ein *Sanftanlauf* eingestellt werden, für jeden Ausgang getrennt.

Die Konfiguration der Parameter erfolgt komplett über den Sender und ein entsprechendes Menu.

Der **RC Quad-D** wird an den **IBus**- oder **SBus**-Ausgang eines Empfängers angeschlossen. Sollen mehrere **RC Quad-D** verwendet werden, so werden diese *alle* (parallel) an den IBus/SBus-Ausgang des Empfängers angeschlossen. Jeder **RC Quad-D** hat eine eindeutige *Adresse* und kann so vom Sender angesprochen werden. Da das **RC Quad-D** auch ein *Telemetrie-Modul* darstellt, muss (sollte) es auch mit dem Telemetrie-Bus (IBus, S.Port) verbunden werden.

Insgesamt können über *einen* Übertragungskanal bis zu 8 **RC-MultiSwitch-D**, **RC-MultiAdapter-DA** oder **RC-ServoSwitch-D** und **RC-Quad-D** angeschlossen werden.

### 6.1. Weitere Möglichkeiten

Nach dem gleichen Funktionsprinzip arbeitet der **RC MultiAdapter-DA**. Dieser ermöglicht den Betrieb alter, analoger Zeitmultiplex-Schaltmodule, die nach dem Verfahren von *Robbe*, oder *Graupner/JR*, oder *CP-Elektronik* oder *Beier NMS* arbeiten. Der **RC MultiAdapter-DA** ermöglicht es, bis zu 5 analoge Zeitmultiplex-Schaltmodule anzuschließen. Dadurch bekommt jedes dieser analogen Schaltmodule über den Adapter ebenfalls eine *eindeutige* Adresse, und kann vom Sender angesprochen werden. Eine PWM-Modulation der Ausgänge eines analogen Zeitmultiplex-Schaltmoduls ist allerdings nicht möglich.

Analoge Zeitmultiplex-Schaltmodule der Fa. *Robbe* besitzen neben den 6 Schaltfunktionen (12 Ausgänge) auch noch 2 proportionale Ausgänge. Mit Hilfe des **RC MultiAdapter-DA** können zwei beliebige **OpenTX**



-Kanäle den beiden proportionalen Ausgängen des *Robbe*-Modules zugewiesen werden. Auch gibt es von der Fa. *Robbe* ein 8-Kanal Multi-Prop Modul, was ebenfalls angesteuert werden kann.

Das Modul **RC-MultiSwitch-D** ist ein digitales 8-Kanal-Schaltmodul. Jeder Ausgang kann individuell **ein** /**aus** geschaltet werden. Für den Zustand **ein** kann eine PWM-Rate festgelegt werden, falls dort etwa ein Motor angeschlossen werden soll oder die Leuchtstärke von Lampen / LEDs reduziert werden soll. Zusätzlich können unterschiedliche Blinkmodi konfiguriert werden.

Das Modul **RC-ServoSwitch-D** ist eine Ansteuerung für bis zu 5 Servos. Diese Servos können bis zu 8 Positionen annehmen. Diese Positionen sind den Schalterstellungen bzw. den Menu-Buttons zugeordnet. Statt Verbraucher ein/aus zu schalten, fahren die Servos bestimmte, vorher erlernte Positionen an. Die Geschwindigkeit, mit der sich die Servos bewegen, kann ebenfalls eingestellt werden. Auch können zwei oder mehrere Servos dieselben Bewegungen ausführen.

## 6.2. Ansicht

In den Bildern [Abbildung 1](#) und [Abbildung 3](#) ist jeweils die Ansicht von oben und unten abgebildet.

Die Maße betragen 46mm x 46mm. Die Platine hat Befestigungslöcher mit 3mm Durchmesser.

Die Stromzuführung und der Anschluss der Motoren erfolgt über Schraubklemmverbinder.

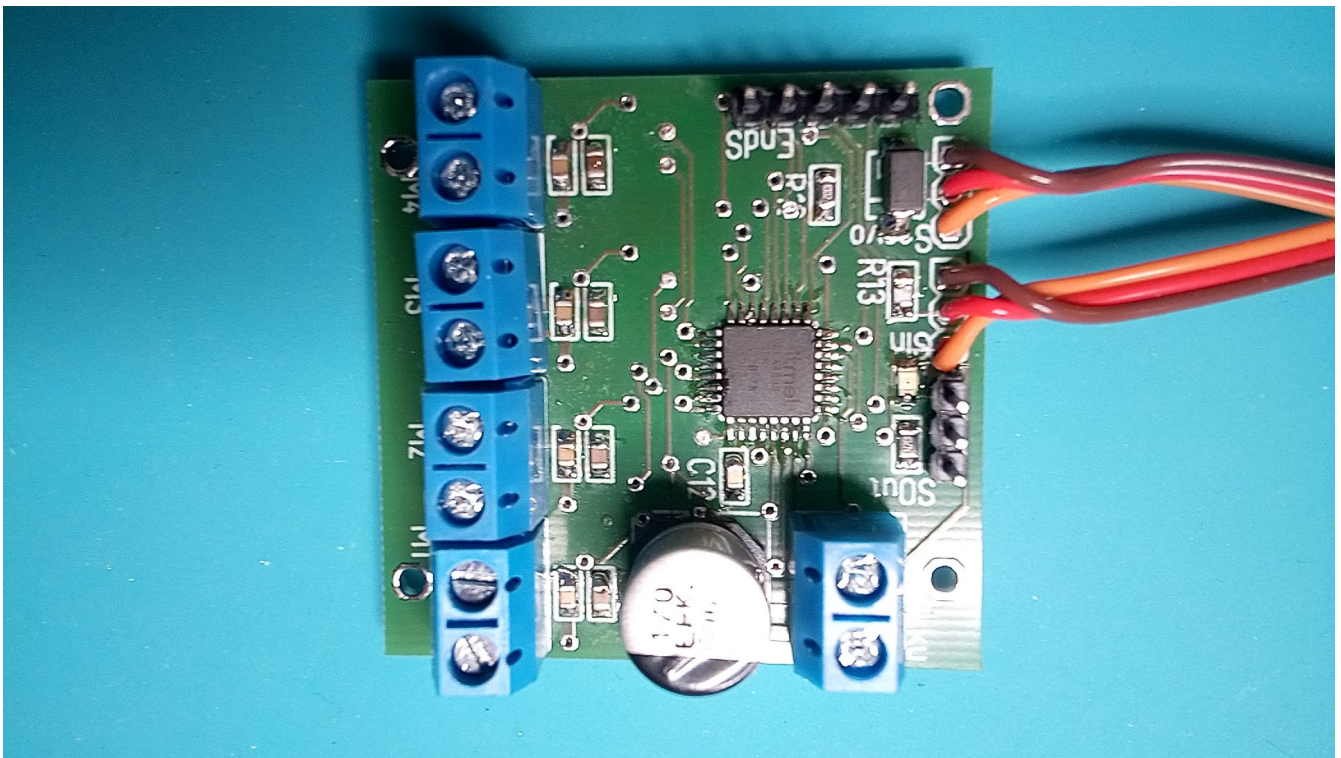


Abbildung 1. Ansicht der Kran- und Sonderfunktionssteuerung von oben (Version 1)

Achtung: die neue **Version 4** hat einige Besonderheiten. Zum Erkennen hier die geänderte Ansicht von oben:



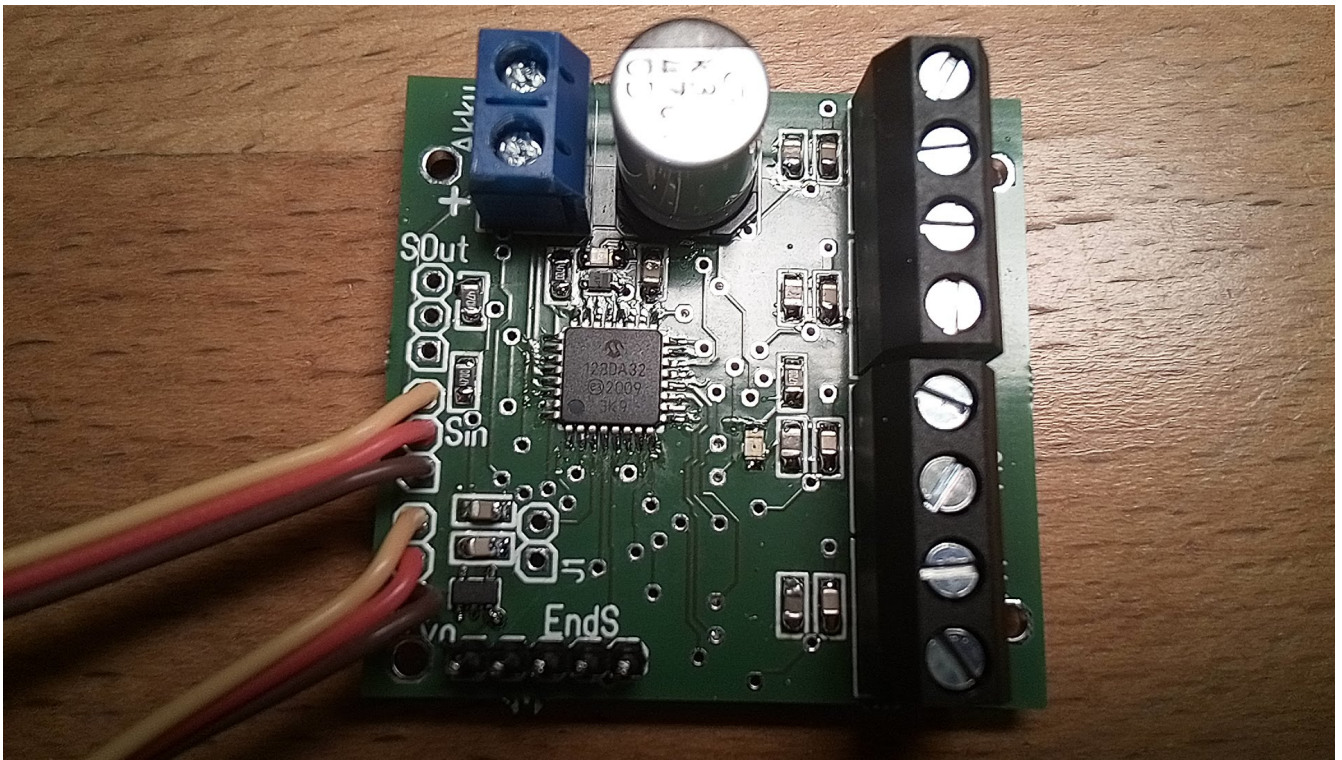


Abbildung 2. Ansicht der Kran- und Sonderfunktionssteuerung von oben (Version 4)

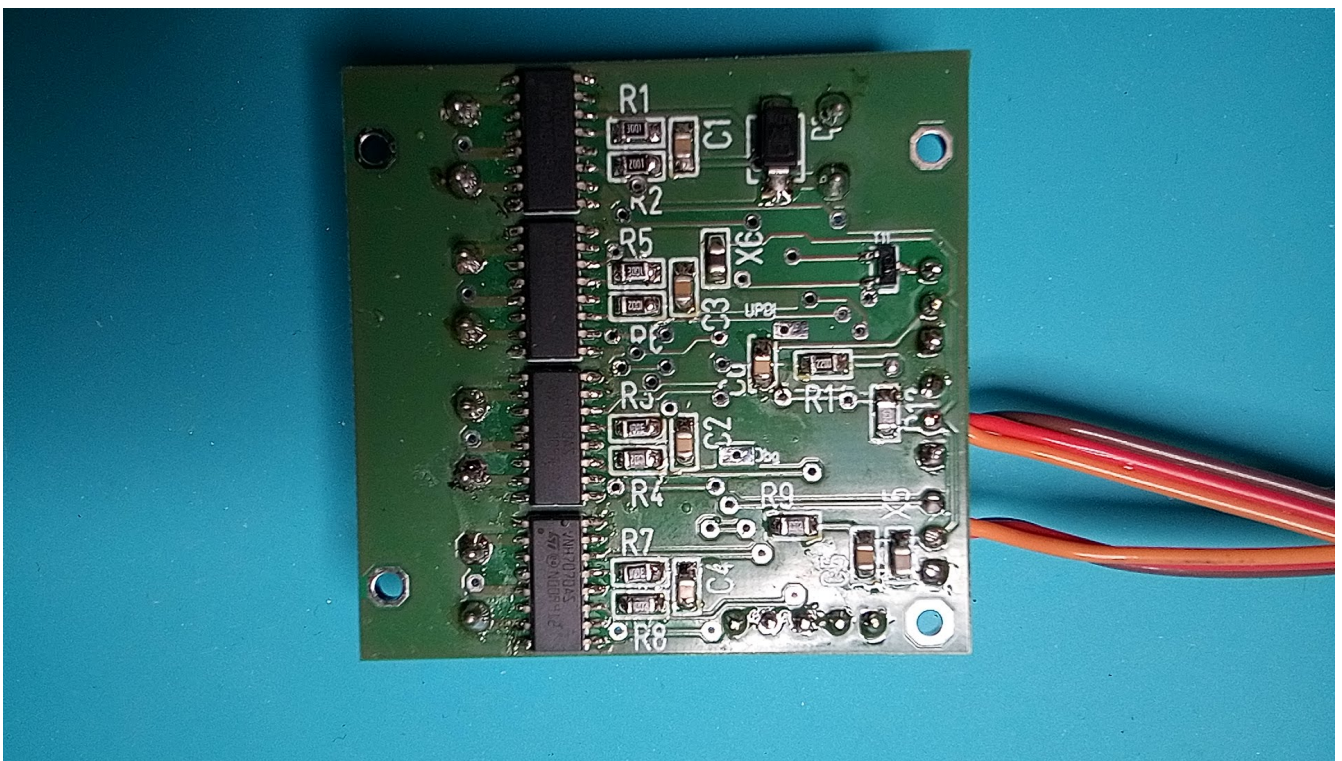


Abbildung 3. Ansicht der Kran- und Sonderfunktionssteuerung von unten

Das Modul wird vom Empfänger mit Spannung versorgt.



Achtung: Hardware Version 1

Die **maximale Spannung** (Empfängerspannung) darf 5,5V **nicht** überschreiten.



Achtung: Hardware Version 4

Die **maximale Spannung** (Empfängerspannung) darf 16V **nicht** überschreiten.

## 6.3. Grundsätzliches

### 6.3.1. Bedienung

Die Bedienung der Schaltmodule erfolgt grundsätzlich über ein *Menu*. Dieses ist als *Widget* in **OpenTx** realisiert (Telemetrie-Skript für kleinere Anlagen).

In **Abbildung 4** sieht man eine Seite (oben rechts: **Page: 1/3**) des Menus. Diese Seite ist für die Bedienung eines **RC-Quad-D** konfiguriert: in der ersten Spalte stehen die *Namen* der Funktionen (hier: **HubObn**, **Auslgr**, **HubUnt**, **Drehen** stellvertretend für *Hubarm-oben*, *Ausleger*, *Hubarm-unten* und *Drehteller*) für die 4 Ausgänge des Steuermoduls. Diese Namen können in der Konfigurationsdatei (s.a. **Abschnitt 7.4.1**) frei gewählt werden. Damit *entfällt* das Problem einer *Schalterbeschriftung*, was man von früher her kennt.

In den weiteren Spalten stehen die verschiedenen *Zustände*, die eine Schaltfunktion annehmen kann. Jeder Zustand kann durch eine *Selektion* aktiviert werden.

Auch die Namen dieser Zustände können in der Konfigurationsdatei frei geändert werden.

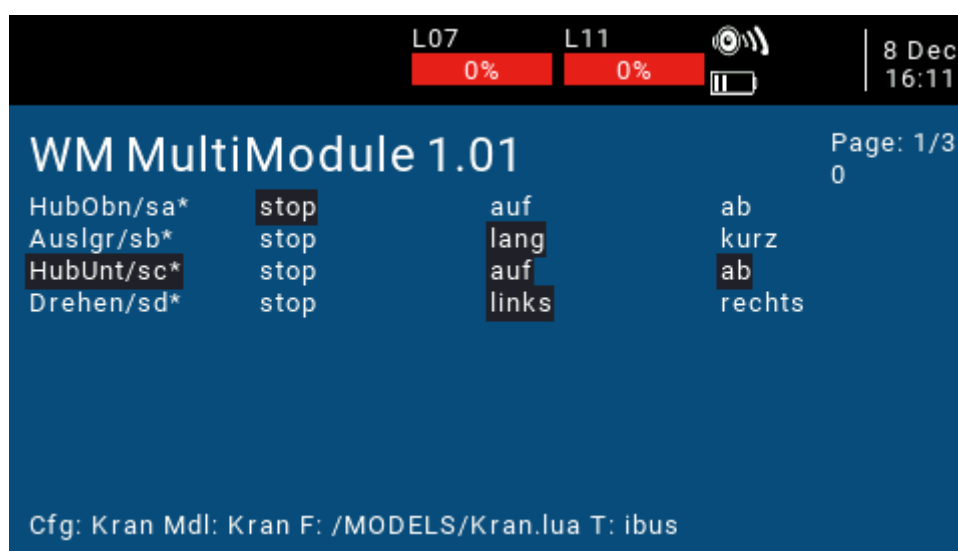


Abbildung 4. Die Bedienoberfläche als Widget (Seite 1)

Sollte eine Schaltfunktion auch ein *ShortCut* in Form eines physischen Schalters (3-Positionenschalter) zugeordnet haben, so erscheint der *Name* des Schalters zusätzlich hinter dem Namen der Funktion. In der ersten Zeile bedeutet **HubObn/sa\***, dass der Funktion **HubObn** zusätzlich der *ShortCut* Schalter **sa** zugeordnet wurde. Dies geschieht auch in der Konfigurationsdatei.

Weil die Anzahl der phys. Schalter am Sender naturgemäß begrenzt ist (etwa: **sa**, ... **sh**), kann man nicht für alle Funktionen *ShortCuts* über Schalter definieren. Eine Abhilfe bieten die sog. *Overlays*. Dies ist bei den analogen Schaltmodulen als *Ebenenumschaltung* bekannt. Wird ein phys. Schalter für *mehr* als eine Funktion definiert, so wird er *automatisch* zu einem *Overlay*. Dies bedeutet, dass die Zuordnung des Schalters zu einer Funktion von der gerade sichtbaren Menu-Seite abhängig ist (Analogie: Menu-Seite = Ebene).

Zusammen mit dem schnellen Umschalten zwischen Menu-Seite mit Hilfe des 6-Positionen-Tasters **6pos** bei manchen Sendern (etwa: **Radiomaster TX16s** oder **FrSky X12s**) ergibt sich eine sehr komfortable Bedienung.



### 6.3.2. Konfiguration

Die Konfiguration wichtiger *Parameter* der Schaltfunktionen erfolgt ebenfalls über ein Menu-System. Dies ist als eigenes Widget realisiert.

Im Bild [Abbildung 5](#) sind zu jeder Funktion die konfigurierbaren *Parameter* aufgelistet. Zudem besteht die Möglichkeit, ein *Reset* durchzuführen. Ein *Reset* setzt für *alle* Parameter wieder die Standardwerte ein.

Soll ein Parameter konfiguriert werden, so muss er *selektiert* werden und sein *Wert* wird dann anschließend über das Potentiometer **S1** eingestellt. Der Wert wird *oben rechts* im Menu in % und auf einer Skala von **0 ... 31** angezeigt (bei **SBus**: **0 ... 15**).

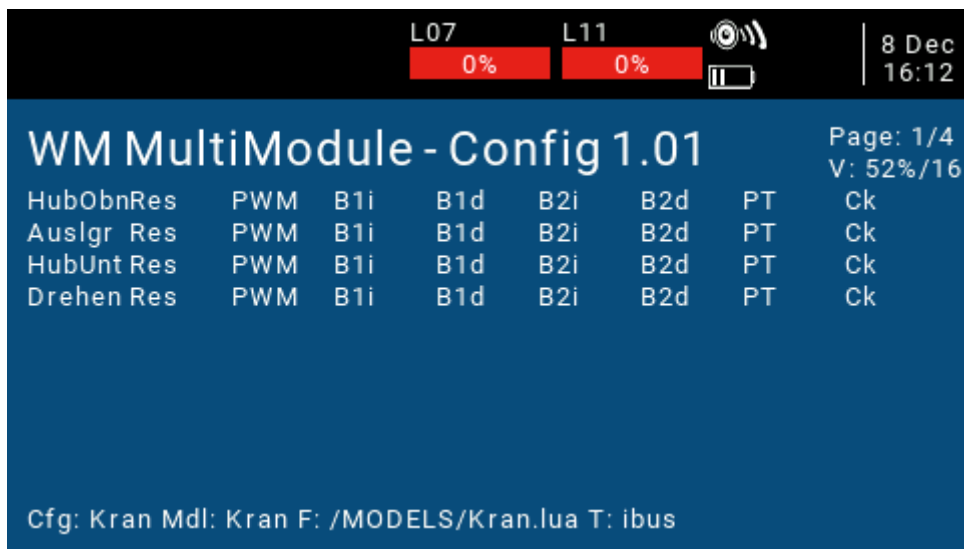


Abbildung 5. Die Konfigurationsoberfläche als Widget (Funktionsspezifische Parameter)

Zusätzlich zu den *funktionsspezifischen* Parametern für jedes Schaltmodul (Achtung: nicht jedes Schaltmodul kann alle Parameter ausführen), existiert noch eine *weitere* Menu-Seite für modul-spezifische (für ein Modul als Ganzes) Parameter (s.a. [Abbildung 6](#)).

Dies betrifft vor allem das Anlernen der Modul-Adresse, wobei dies aus Sicherheitsgründen nur beim *Einschalten* des Moduls möglich ist (s.a. [Abschnitt 9.2](#)).

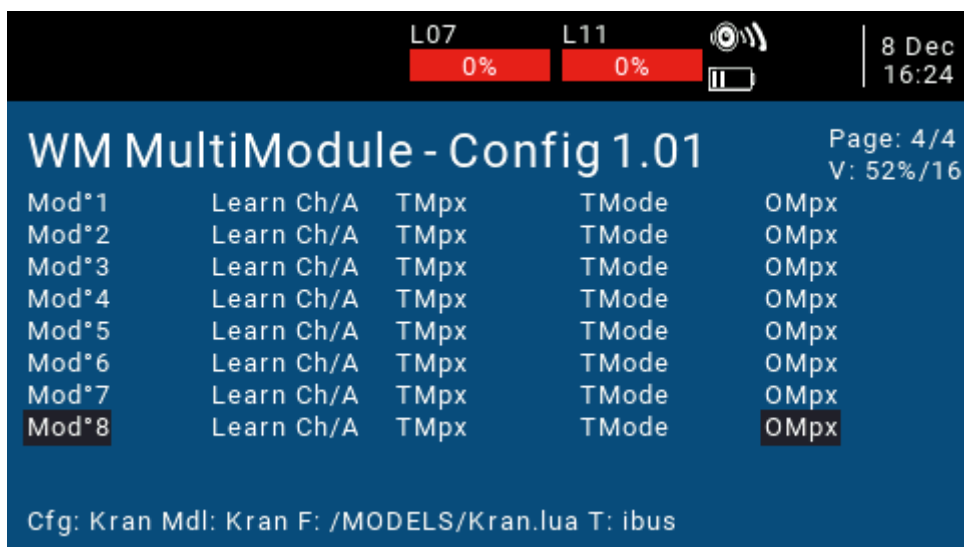


Abbildung 6. Die Konfigurationsoberfläche als Widget (modulglobale Parameter)

Da dieses Widget zum eigentlichen Betrieb *nicht* erforderlich ist, kann es *nach* der Konfiguration deaktiviert werden bzw. durch andere Widgets ersetzt werden (dies gilt auch für das Telemetrie-Skript zu Konfiguration).

## 7. Software für OpenTX für den Betrieb von RC-Quad-D

Für den Betrieb und zur Konfiguration des RC-Quad-D sind einige Scripte und Konfigurationsdateien notwendig. Diese findet man in der aktuellsten Version jeweils auf [GitHub](#).

*Lizenz der Software, des Übertragungsverfahrens und der Dokumentation*



Bitte beachten Sie, dass neben dieser Dokumentation auch die [Software](#) und das Übertragungsverfahren als Konzept der einer Lizenz (Nutzungsvereinbarung) unterliegt. Die genauen Bedingungen dieser Lizenz finden Sie [hier](#).

Für den RC-Quad-D sind die folgenden Dateien relevant (wie auch bei allen anderen *digitalen* Modulen):

- Widgets
  - `Widgets/WMSW/main.lua` (Bedienungs-Widget)
  - `Widgets/WMSWC/main.lua` (Konfigurations-Widget)
- Mixer-Scripte
  - `MIXES/encib.lua`
  - `MIXES/encsbm.lua` (für Anlagen mit MPM-HF-Modul)
  - `MIXES/encxjt.lua` (für Anlagen mit internem XJT-HF-Modul)
  - `MIXES/encsb.lua` (nur zu Testzwecken)
- Telemetrie-Scripte (für monochrome Sender)
  - `SCRIPTS/TELEMETRY/wmsw.lua` (Bedienungs-Widget)
  - `SCRIPTS/TELEMETRY/wmswc.lua` (Konfigurations-Widget)
- Bibliotheken
  - `SCRIPTS/CONFIG/wmcfg.lua`
  - `SCRIPTS/WM/wmlib.lua`
- Konfiguration
  - `MODELS/swstd.lua` (Standardkonfiguration für Anlagen mit großem Display)
  - `MODELS/swstdm.lua` (Standardkonfiguration für Anlagen mit mittlerem Display)
  - `MODELS/swstds.lua` (Standardkonfiguration für Anlagen mit kleinem Display)
  - `MODELS/swstdx.lua` (Standardkonfiguration für Anlagen mit kleinem Display und wenigen Modulen)
  - `MODELS/<name>.lua` (modellspezifische Konfigurationen)(z.B. `Abc.lua` für das Model mit dem Namen `Abc`)

## 7.1. Installation der Software auf dem Sender

Kopieren Sie die o.g. Dateien in *korrespondierende* (gleichlautende) Verzeichnisse auf der SD-Karte des Senders.



Die Datei `MODELS/swstd.lua` sollte unangetastet bleiben (als *fallback*-Konfiguration). Stattdessen kopieren Sie diese Datei in für jedes Modell jeweils eigene Dateien. Dann ist für das Modell mit dem Namen `ABC` der *Dateiname* `ABC.lua` zuständig. Verwenden Sie *keine* Leerzeichen im Namen und nur max. 6 Zeichen vor dem `.lua`.

Kommen weitere Modelle hinzu, so wiederholen Sie diesen Schritt für *jedes* Modell mit angepasstem Namen.

Anschließend kann der *Datei-Inhalt* mit einem normalen Texteditor (etwa `NotePad`) editiert werden.

Bitte beachten Sie auch die Unterscheidung zwischen `Sbus` und `IBus`.

### 7.1.1. Unterscheidung der verschiedenen Sendermodelle

Grundsätzlich unterscheiden sich die Sender mit den `480x272`-Pixel großen *Farbdisplay* von den Sendern mit den `128x64`-Pixel oder `212x64`-Pixel großen *monochromen* Displays von einander:

- Typ-1: `128x64`-Pixel oder `212x64`-Pixel großes *monochromes* Display
- Typ-2: `480x272`-Pixel großen *Farbdisplay* (auch *Touchdisplay*)

Bei *Typ-1* Sendern existieren im Menu-System keine sog. *Widgets* sondern nur *Telemetrie*-Seiten. Auf der anderen Seite existieren bei den *Typ-2* Sendern keine *Telemetrie*-Seiten sondern wiederum nur *Widgets*. Dies macht eine Unterscheidung bei den Skripten für die Schaltmodule notwendig.

Zudem sind die *Typ-1* Sender im verfügbaren `RAM` sehr *eingeschränkt*. Daher kann hier oft das Telemetrie-Skript zur *Bedienung* und das Telemetrie-Skript zur *Konfiguration* nicht gleichzeitig geladen werden. Es kommt in diesem Fall zu einer *Fehlermeldung* des Senders. Man kann sich in so einem Fall behelfen, indem man die (nicht so häufige) Konfiguration und die (häufigere) Bedienung abwechselnd auf *eine* Telemetrie-Seite legt.

## 7.2. Konfiguration des Modells (Sender mit `480x272`-Pixel *Farbdisplay*, Typ-2)

Die folgenden Einstellungen gelten *nur* für Sender mit `480x272`-Pixel *Farbdisplay*.

### 7.2.1. Widget für die Bedienung

Richten Sie eine Telemetrie-Seite (volle Größe) ein mit dem Widget `Widgets/WMSW/main.lua`. Dieses stellt sich als `Wm MultiModule x.y` dar.

### 7.2.2. Widget für die Konfiguration

Richten Sie eine Telemetrie-Seite (volle Größe) ein mit dem Widget `Widgets/WMSWC/main.lua`. Dieses stellt

sich als `Wm MultiModule Config x.y` dar.

### 7.2.3. Mixer-Script

Richten Sie *eines* der **LUA**-Scripte als *Mixer-Script* ein. Hier wählen Sie *nur eines* der Auswahl aus:

- `encib` für **IBus**-Empfänger
- `encsbm` für **SBus**-Empfänger und Sender mit MPM-Multiprotokoll-HF-Modul
- `encxjt` für **SBus**-Empfänger und Sender mit XJT-HF-Modul

### 7.2.4. Definition eines Übertragungskanals für das **RC-Quad-D** am Empfänger

Das Schaltmodul wird über eine serielle Schnittstelle an den Empfänger angeschlossen. Dies ist je nach Empfänger **IBus** oder **SBus**.

Das Schaltmodul ist standardmäßig auf Kanal **10** eingestellt. Es kann jedoch auch ein anderer Kanal angelernt werden (s.a. [Abschnitt 9.2](#)).

Im Sender muss daher für diesen Kanal als Quelle das entsprechende Mixer-Script eingetragen werden.

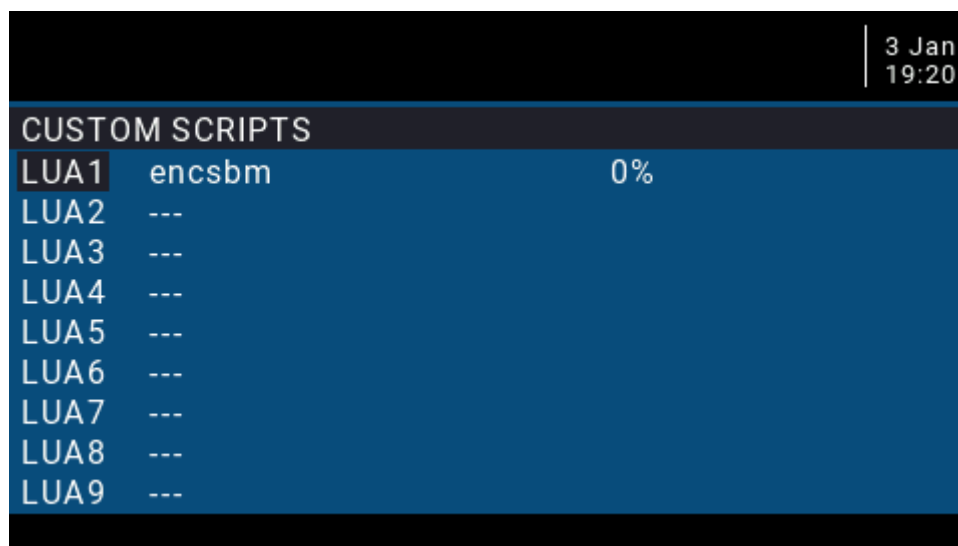


Abbildung 7. Das Mischer-Skript `encsbm.lua`



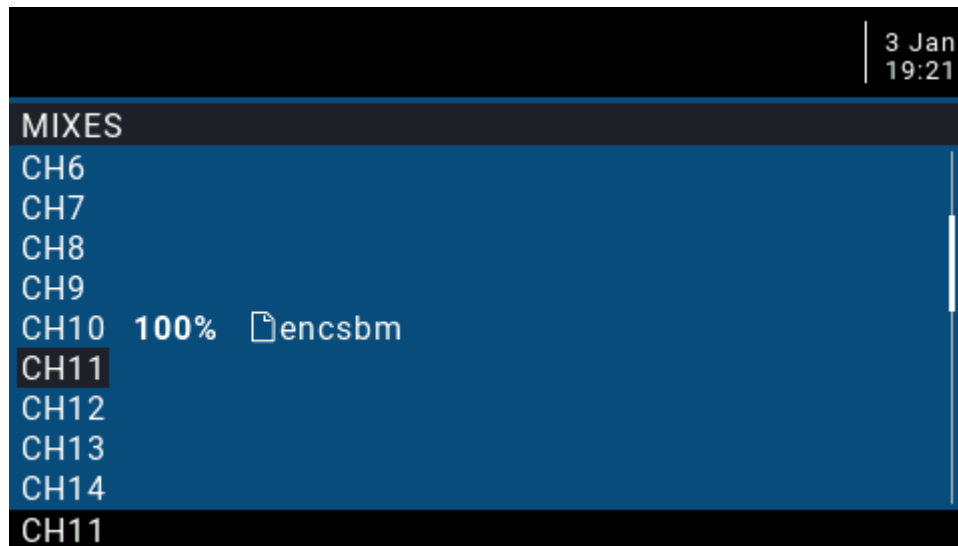


Abbildung 8. Hier wird der Übertragungskanal 10 wird für die Schaltmodule verwendet.

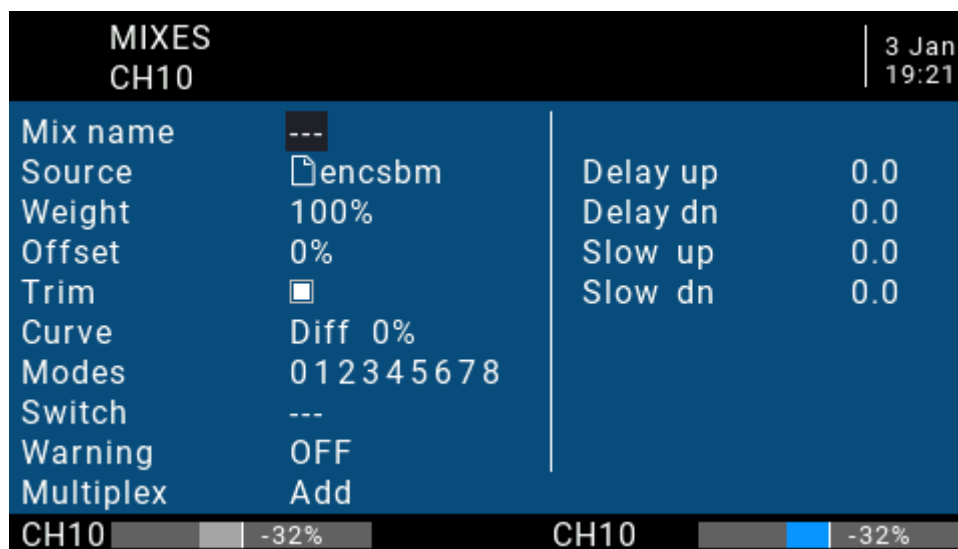


Abbildung 9. Als Quelle des Mischers des Übertragungskanals wird das Mischer-Skript `encsbm.lua` eingetragen



#### Mögliche Fehlerquelle im späteren Betrieb

Bei *jedem* Einschalten scannen die Schaltmodule *alle* Übertragungskanäle auf ein spezielles Konfigurationssignal, um den Übertragungskanal zu finden und ggf. die Adresse des Moduls neu zu lernen. Dieses Konfigurationssignal kann auch durch zufällige Geberposition auf anderen Kanälen unbeabsichtigt entstehen. Um dieses Problem zu umgehen, sollten die Übertragungskanäle mit Nummern *oberhalb* des gewählten Kanals für die digitalen Schaltmodule *unbenutzt* sein. Um auch bei späteren Erweiterung von Modellen hier vor Fehlern geschützt zu sein, sollte sich der Übertragungskanal für die digitalen Schaltmodule auf der *höchsten* Kanalnummer befinden. Dies ist bei **SBus** der Kanal **16** und bei **IBus** der Kanal **14** oder **16** (je nach verwendeter Version der Firmware im *4in1-MPM-HF-Modul*).

### 7.2.5. Definition der Geber (Inputs)



### Bedienelemente für die Widgets

Die Widgets müssen mit Bedienelementen des Senders bedient werden können (Scroll, Select).

In **OpenTX**-Version < **2.4.0** kann dies nur durch *normale* Bedienelemente (Geber) erfolgen. Dazu sind einige Definitionen nötig.

Definieren Sie die folgenden, in den Widgets voreingestellten *Inputs* nach dem Muster aus [Abbildung 10](#):

- I8 : Navigation von links nach rechts (und vom Zeilenende in den Zeilenanfang der nächsten Zeile)
- I9 : Navigation von rechts nach linkes (und vom Zeilenanfang in das Zeilenende der vorherigen Zeile)
- I10 : Selektion
- (I11) : Navigation nach oben (in Standardeinstellung unbenutzt)
- (I12) : Navigation nach unten (in Standardeinstellung unbenutzt)

(Lassen die die *unbenutzten* Inputs in ihrem Modell frei. Andernfalls sind Fehlfunktionen nicht ausgeschlossen.)

Möchten Sie andere Geber-Nummern verwenden, so ist dies auch möglich. Die Gebernummern müssen dann in der entsprechenden *Widget*-Konfiguration definiert werden. Dies ist dann für beide Widgets gleichermaßen notwendig bzw. empfehlenswert.

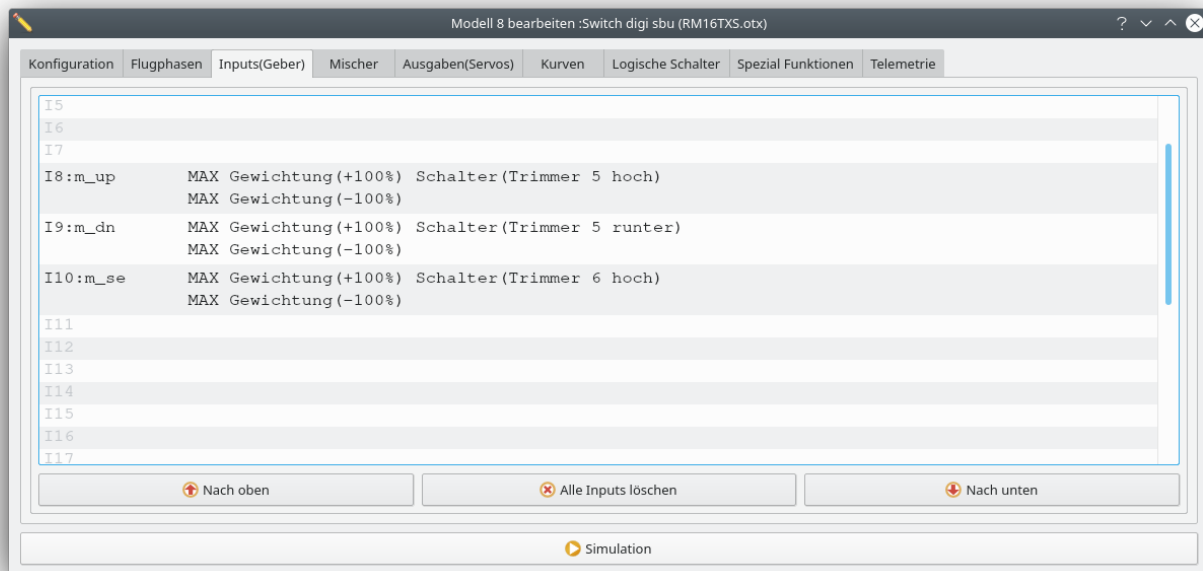


Abbildung 10. Benötigte Geber zur Bedienung der Widgets (Darstellung im Companion)

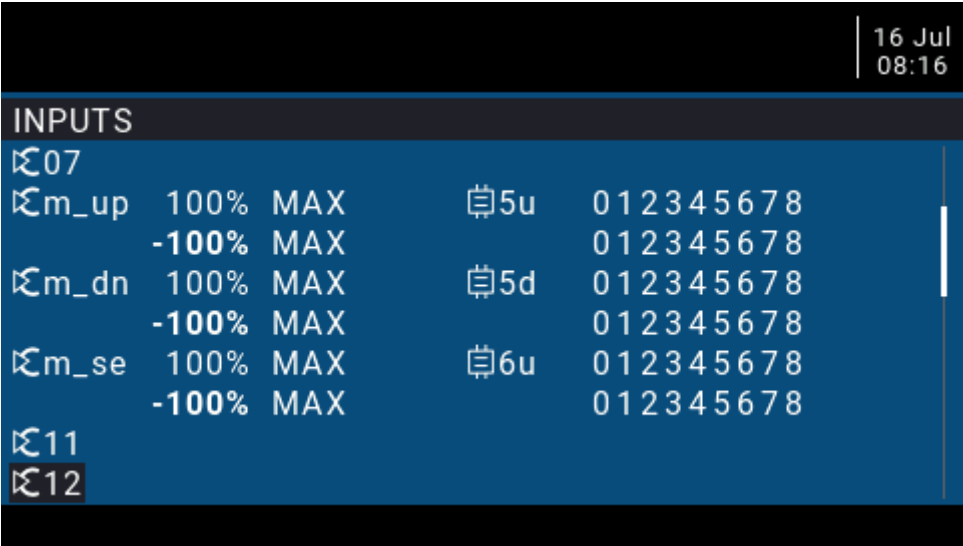


Abbildung 11. Benötigte Geber (Darstellung im Sender)

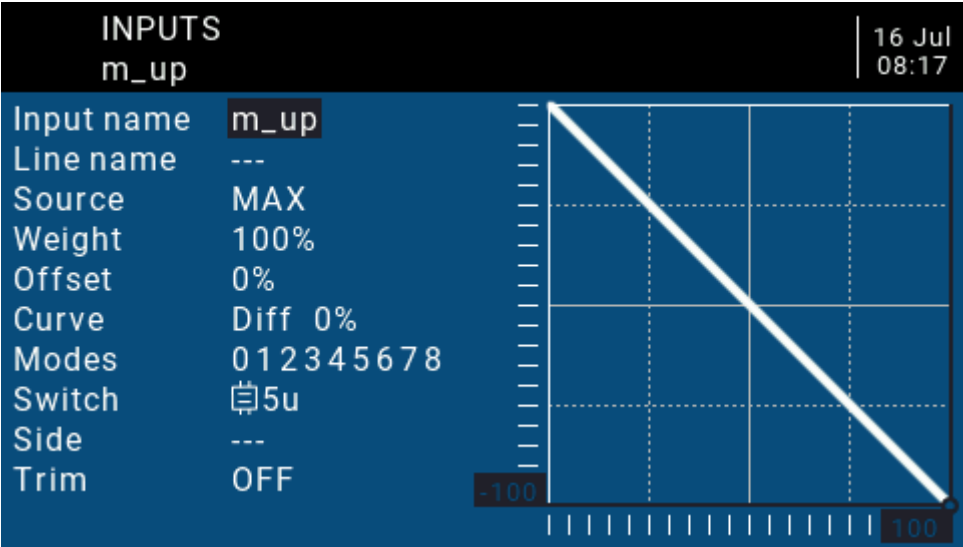


Abbildung 12. Benötigte Geber (Darstellung im Sender)

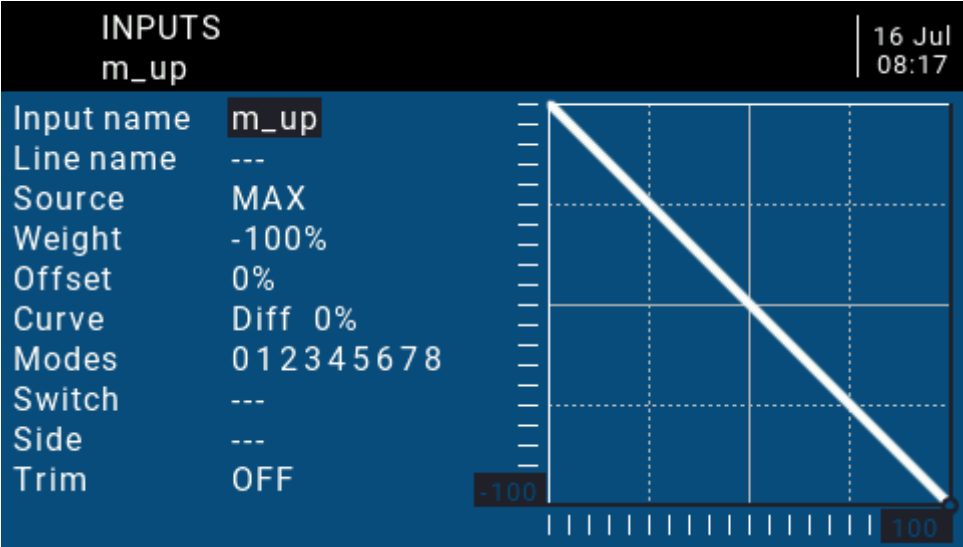


Abbildung 13. Benötigte Geber (Darstellung im Sender)

Um die Bedienung weiter zu vereinfachen, werden standardmäßig weitere Bedienelemente für die Widgets verwendet. Diese sind voreingestellt. Sie können aber in der Datei `MODELS/swstd.lua` bzw. der modellspezifischen Datein `MODELS/<name>.lua` geändert werden.



#### *SpeedDial mit den Gebern LS und RS*

Bei Sendern mit den beiden Schiebern **LS** und **RS** sind diese für die direkte Anwahl von Menu-Punkten im Normalfall eingerichtet.



#### *Menu-Schnellanwahl mit dem 6-Positionentaster 6pos (Krähenfüße)*

Der 6-Positionentaster **6pos** ist standardmäßig als *Schnellwahl* für die ersten sechs Menuseiten (Schaltmodule) konfiguriert. Dies ist besonders interessant mit der Möglichkeit der *Overlays* (Ebenenumschaltung).



#### *Parameterwerte einstellen mit Potentiometer S1*

Im Widget für die *Konfiguration* des **RC-Quad-D** kann/muss man einige Parameter einstellen wie etwa die **PWM**-Rate für die Motoren oder die Abschaltströme. Bei anderen Modulen sind dies auch andere Parameter. Wenn dieser Parameter im Menu angewählt ist, kann man den Wert direkt über das Poti **S1** einstellen. Dies wird dann auch im Menu als Wert (oben rechts) angezeigt. Die aktive Selektion wird durch eine *Invers*-Darstellung der Werte oben rechts im Menu kenntlich gemacht.

### 7.2.6. Spezielle Konfiguration für Sender mit Joystick (FrSky X12s)

Der Sender *FrSky Horus X12s* besitzt ein besonderes Bedienelement: den *Joystick* links unten. Dieser kann statt der o.g. Trimm-Buttons ebenfalls für die Navigation im Menu-System der Schaltmodule verwendet werden. Da es sich dabei um proportionale Geber handelt, muss mit Hilfe eine Kurve (Sprungfunktion) aus der proportionalen Informations des Gebers eine digitale gemacht werden.



Wird die hier beschriebene Konfiguration ausgeführt, so sollten die Geber **Jsx** und **Jsy** nicht mehr als *normale* Geber verwendet werden, da sie in jedem Fall den *Cursor* der Widgets beeinflussen.

Es muss dazu *eine* Kurve nach dem Muster [Abbildung 14](#) erstellt werden. Diese Kurve kann für beide Richtungen (positive und negative Richtung) jeder Achse des Joysticks verwendet werden. Die *Spiegelung* der Kurve geschieht durch eine *Negation* (das Ausrufezeichen **!** in der *Input-Konfiguration*). *Getrue nach der Regel, dass \_Unzulänglichkeiten* der Geber in der Input-Konfiguration *korrigiert* werden, geschieht dies durch vier Input-Definitionen aus den zwei Gebern **Jsx** und **Jsy** für die Navigation **rechts**, **links**, **oben** und **unten**.

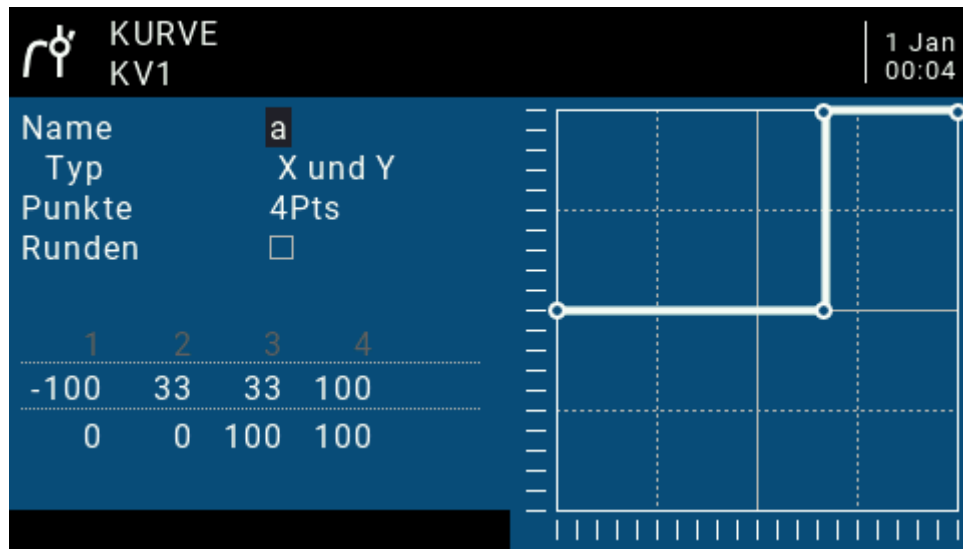


Abbildung 14. Kurve zur Umwandlung einer JoyStick-Bewegung in eine Navigation

In [Abbildung 15](#) sieht man die Verwendung der Kurve in einem Input.

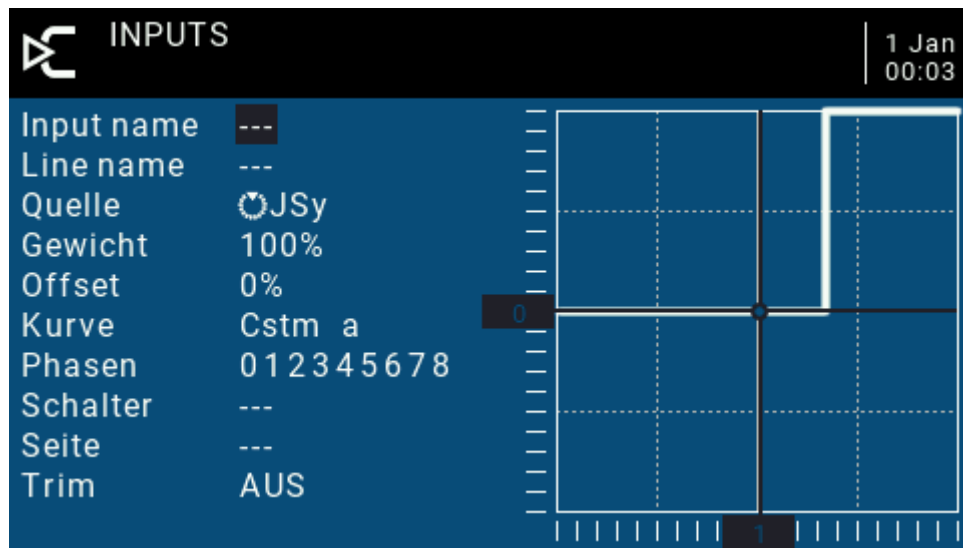


Abbildung 15. Verwendung der Kurve aus [Abbildung 14](#) für den Joystick in y-Richtung

In [Abbildung 16](#) sieht man den Überblick über die gesamte Konfiguration aller vier Inputs. Man achte auf die Spiegelung der Kurve durch !.

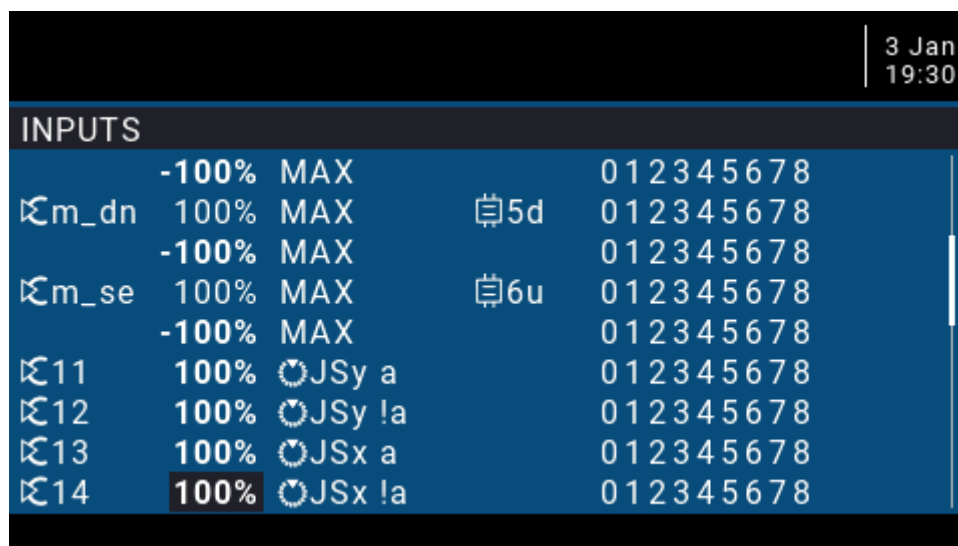


Abbildung 16. Konfiguration der alternativen Inputs 11...14 für die Bedienung

Der letzte Schritt besteht im *Einbau* der Inputs in die Widget-Konfiguration:

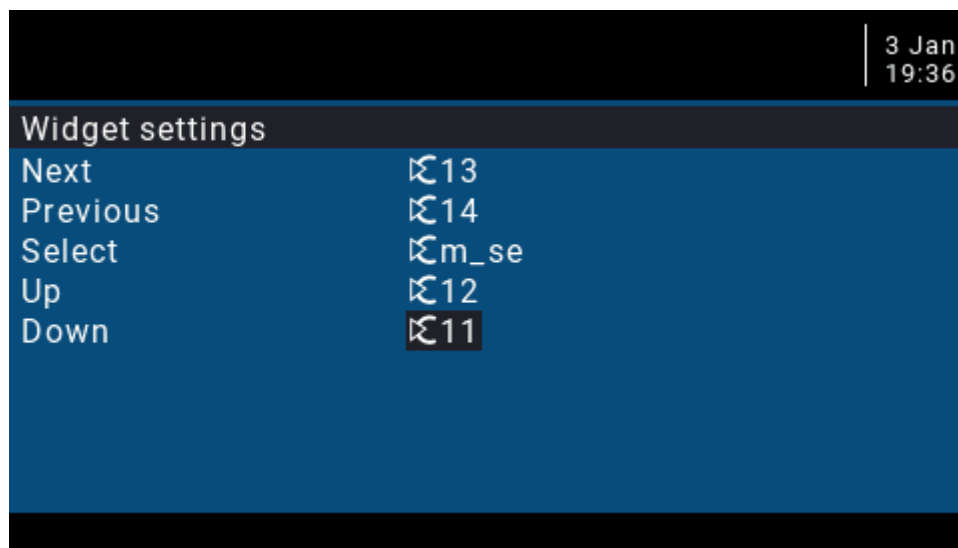


Abbildung 17. Konfiguration des Widget zur Verwendung der alternativen Inputs

Diese Konfiguration kann sowohl für das *Bedienungswidget* **wmsw** wie auch für das *Konfigurationswidget* **wmswc** durchgeführt werden.

### 7.2.7. Flugphasen

Damit die (oft unbenutzten) zusätzlichen Trimm-Taster für die Bedienung der Widgets benutzt werden können, müssen Sie als Trimm-Funktion deaktiviert werden. Die geschieht in der entsprechenden *Flugphase*, in der die Bedienung der Widgets möglich sein soll. Also am besten in allen Flugphasen bzw. in der Flugphase 0.

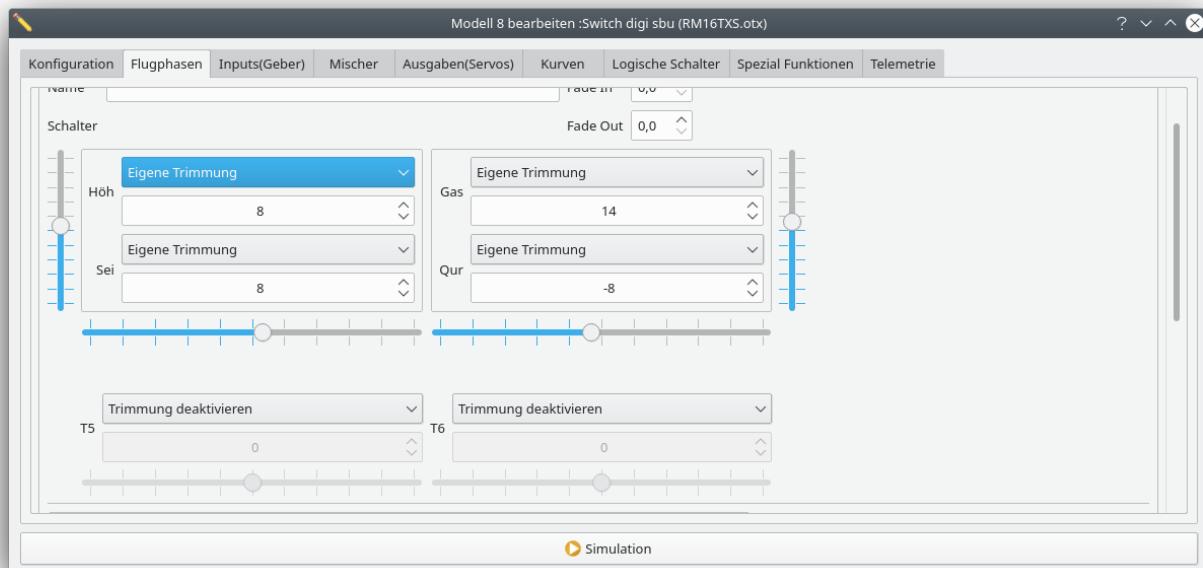


Abbildung 18. Deaktivierung der Trimmer T5 und T6 in der Flugphase 0 (Darstellung im Companion)

16 Jul 08:15									
FLIGHT MODES									
FM0	---	N/A	:0	:0	:0	:0	--	--	0.0 0.0
FM1	---	---	:0	:0	:0	:0	:0	:0	0.0 0.0
FM2	---	---	:0	:0	:0	:0	:0	:0	0.0 0.0
FM3	---	---	:0	:0	:0	:0	:0	:0	0.0 0.0
FM4	---	---	:0	:0	:0	:0	:0	:0	0.0 0.0
FM5	---	---	:0	:0	:0	:0	:0	:0	0.0 0.0
FM6	---	---	:0	:0	:0	:0	:0	:0	0.0 0.0
FM7	---	---	:0	:0	:0	:0	:0	:0	0.0 0.0
FM8	---	---	:0	:0	:0	:0	:0	:0	0.0 0.0

Abbildung 19. Deaktivierung der Trimmer T5 und T6 in der Flugphase 0 (Darstellung im Sender)

## 7.3. Konfiguration des Modells (Sender mit 128x64-Pixel oder 212x54-Pixel monochrome Display: Typ-1)

Die folgenden Einstellungen gelten nur für Sender mit 128x64-Pixel oder 212x54-Pixel monochromem Display.

### 7.3.1. Telemetrie-Seiten

Richten Sie je eine Telemetrie-Seite mit dem Skript `SCRIPTS/TELEMETRY/wmsw.lua` bzw. `SCRIPTS/TELEMETRY/wmswc.lua` ein.





Diese Sender haben sehr wenig RAM. Der Betrieb von *beiden* o.g. Scripten *gleichzeitig* ist oft nicht möglich und kann zu unterschiedlichen Fehlermeldungen führen. Sollte das passieren, so können Sie entweder nur das eine *oder* das andere Script einstellen. Die Einschränkung ist nicht so groß, da man das Konfigurationsscript ja nur selten (in der Werkstatt) benötigt.

### 7.3.2. Mixer-Script und Übertragungskanal

Für die Einrichtung des passenden *Mixer*-Scripts und *Übertragungskanals* folgende sie bitte [Abschnitt 7.2.3](#) bzw. [Abschnitt 7.2.4](#).

### 7.3.3. Geber und Flugphasen

Die *Bedienung* der Skripte erfolgt *vollständig* über die normalen Bedienelemente für die Benutzerschnittstelle: Navigationstasten bzw. Navigationsrad. Daher ist eine Konfiguration von speziellen Gebern und Flugphasen nicht notwendig.

## 7.4. Konfigurationsdateien

### 7.4.1. Die Datei `MODELS/swstd.lua` und weitere Dateien

In dem Ordner `MODELS` des SD-Karte werden modellspezifische Menu-Konfigurationen abgelegt.

Dabei gilt folgende Suchreihenfolge für ein Model mit dem Namen `ABC`:

1. `MODELS/ABC.lua`
2. `MODELS/swstd.lua` (bzw. `MODELS/swstdx.lua` für kleine Sender)

Sinnvollerweise kopiert man die Datei `swstd.lua` in eine Datei `ABC.lua` für das Model `ABC` und editiert diese entsprechend den eigenen Wünschen.

In der Datenstruktur `menu` können an folgenden Elementen Veränderungen vorgenommen werden:

- Namen der Funktionen (`menu.pages.items.name`)
- Namen der Zustände (`menu.pages.items.states`)
- Namen der phys. Schalter (`menu.pages.items.data.switch`)
- Adresse des Moduls (`menu.pages.items.data.module`)

*Unterscheidung zwischen SBus und IBus*

Leider muss durch die Beschränkungen des SBus-Protokolls noch an einer zweiten Stelle eine Änderung vorgenommen werden. Damit die folgende Änderung jeweils für ein bestimmtes Modell gültig ist, sollte sie auch in der modellspezifischen Datei bspw. `ABC.lua` vorgenommen werden:

- Für SBus: der Parameter `useSbus = 1`
- Für IBus, etc.: der Parameter `useSbus = 0`



Achten Sie ebenfalls darauf, dass Sie das *dazu passende* Mixer-Script `encsbm.lua`, `encxjt.lua` oder `encib.lua` eingerichtet haben (s.a. [Abschnitt 7.2.3](#)).

Diese Beschränkungen haben zur Folge, dass für SBus nur 16 verschiedenen Parameterwerte (etwa für den PWM-Wert oder das Blink-Intervall) eingestellt werden können.

Am *unteren, rechten* Rand des Widgets wird angezeigt, ob eine SBus oder IBus Konfiguration aktiv ist (zur Kontrolle).

Quelltext 1. Ausschnitt A (aus der Datei `swstd.lua`, `swstdm.lua`, ...)

```

1 local name = "Default";
2
3 local gVariable = 5;
4
5 local useSbus = 1; -- only 4 states, only 16 parameter values ①
6
7 local gstates1 = {"aus", "ein", "blink1", "blink2"}; ⑦
8 local gstates2 = {"aus", "ein 1", "ein 2"};
9 local gstates3 = {"Pos1", "Pos2", "Pos3", "Pos4", "Pos5"};
10
11 local menu = {
12   title = "WM MultiSwitch",
13
14   scrollUpDn = "ls", -- direct navigating
15   scrollLR = "rs",
16
17   parameterDial = "s1",
18
19   pageSwitch = "6pos";
20
21   remote = "trn16";
22
23   state = {
24     activeRow = 1,
25     activeCol = 1,
26     activePage = nil
27   },
28   pages = {
29     { -- template for digital multiswitch RC-MultiSwitch-D @ Address(1)
30       items = { ②
31         {name = "M1A", states = gstates1, state = 1, data = {switch = "sa", count = 1,
32           module = 1}}, ③

```

```

32     {name = "M1B", states = gstates1, state = 1, data = {switch = "sb", count = 2,
    module = 1}},
33     {name = "M1C", states = gstates1, state = 1, data = {switch = nil, count = 3, module
    = 1}},
34     {name = "M1D", states = gstates1, state = 1, data = {switch = "se", count = 4,
    module = 1}}, ⑤
35     {name = "M1E", states = gstates1, state = 1, data = {switch = nil, count = 5, module
    = 1}},
36     {name = "M1F", states = gstates1, state = 1, data = {switch = nil, count = 6, module
    = 1}},
37     {name = "M1G", states = gstates1, state = 1, data = {switch = nil, count = 7, module
    = 1}},
38     {name = "M1H", states = gstates1, state = 1, data = {switch = nil, count = 8, module
    = 1}},
39     }
40 },
41 { -- template for digital multiswitch RC-MultiSwitch-D @ Address(2)
42     items = { ④
43         {name = "M2A", states = gstates1, state = 1, data = {switch = "sc", count = 1,
    module = 2}},
44         {name = "M2B", states = gstates1, state = 1, data = {switch = "sd", count = 2,
    module = 2}},
45         {name = "M2C", states = gstates1, state = 1, data = {switch = nil, count = 3, module
    = 2}},
46         {name = "M2D", states = gstates1, state = 1, data = {switch = "se", count = 4,
    module = 2}}, ⑥
47         {name = "M2E", states = gstates1, state = 1, data = {switch = nil, count = 5, module
    = 2}},
48         {name = "M2F", states = gstates1, state = 1, data = {switch = nil, count = 6, module
    = 2}},
49         {name = "M2G", states = gstates1, state = 1, data = {switch = nil, count = 7, module
    = 2}},
50         {name = "M2H", states = gstates1, state = 1, data = {switch = nil, count = 8, module
    = 2}},
51     }
52 },
53 ...
54 };
55 ...

```

- ① Diese Namen für die Schaltzustände können frei gewählt werden.
- ② Dies ist die Konfiguration für das Module mit der Adresse 1 (s.a. `module = 1`) auf der Seite 1 des Menus (der *erste* `items` Block).
- ③ Den Namen der Funktion (`name = "M1A"`) können Sie beliebig ändern.
- ④ Dies ist die Konfiguration für das Module mit der Adresse 1 (s.a. `module = 2`)
- ⑤ Diese Funktion hat einen Schalter zugeordnet (`switch = "se"`).
- ⑥ Diese Funktion hat einen Schalter zugeordnet (`switch = "se"`). Da es derselbe Schalter ist, der schon für eine Funktion auf der Seite 1 verwendet wurde, ist dies ein sog. *Overlay* (im Menu dann besonders gekennzeichnet)
- ⑦ Dies sind die Namen der unterschiedlichen *Zustände*, die eine Funktion annehmen kann.

Wird ein phys. Schalter *einer* Schaltfunktion zugeordnet, dann kann diese Funktion *immer* mit diesem

Schalter bedient werden. Der Schalter ist dann ein *ShortCut* (Abkürzung).

Wird ein phys. Schalter *mehr als einer* Schaltfunktion zugeordnet, dann wird dieser Schalter *automatisch* zu einem *Overlay*. Dies wird dann im Menu durch ein *\** hinter dem Namen des Schalters dargestellt. Damit wirkt der Schalter dann *nur* auf die zugeordnete Funktion der *aktiven* Menu-Seite (sichtbar).



#### *ShortCuts (Abkürzungen)*

Die physischen Schalter stellen gewissermaßen *ShortCuts* (Abkürzungen) für die Bedienung des Menus dar. Deswegen darf *ein* Schalter auch nur *einer* Funktion in der *gesamten* Menu-Konfiguration zugeordnet werden. Das bedeutet, dass bspw. der Schalter *sa* immer auf die Funktion *Fun3* des Moduls *1* wirkt, und zwar unabhängig, welches Menu gerade angezeigt wird (etwa das Menu vom Modul *2*).

#### *Overlays (Ebenen)*

Manchmal ist es wünschenswert, dass ein phys. Schalter nicht *immer* auf *dieselbe* Funktion wirkt, sondern dass man:

1. zwischen den Menus der angeschlossenen Schaltmodule wählen kann (Ebenenumschaltung), und dann
2. ein phys. Schalter abhängig von dem so gewählten Schaltmodul eine andere Bedeutung bekommt.



Das bedeutet, dass bspw. der Schalter *sa* bei aktivem Menu des Moduls *1* auf dessen Funktion *Fun3* wirkt, bei aktivem Menu des Moduls *2* aber auf die Funktion *Fun3* des Moduls *2*.

Einige Modellbauer werden diese Funktion als *Ebenenumschaltung* kennen. Bei analogen Schaltmodulen hat man eine Ebenenumschaltung dadurch realisiert, dann man mit einem Schalter und Mischern etwa den Ausgangskanal für das Zeitmultiplexsignal umgeschaltet hat, um mehrere analoge Schaltmodule zu erreichen. Damit wurden die Schaltmodule *als Ganzes* umgeschaltet. Bei *RC-Quad-D* ist dies flexibler bezogen auf individuelle Schalter möglich.

#### *Menuseiten und Schaltmodule*



Die Menu-Seiten mit jeweils 8 Zeilen müssen *nicht* immer der Zuordnung zu den phys. Schaltmodulen entsprechen: auf einer Menu-Seite können Funktionen *unterschiedlicher* Schaltmodule untergebracht werden. Denn jedes Modul hat eine *Adresse*, und diese Adresse ist Bestandteil der Funktionsdefinition in der Datei. Man kann also die Menu-Seiten auch eher nach Funktionsgruppen strukturieren als nach Schaltmodulen.

### 7.4.2. Das Widget *WMSW/main.lua*

In dieser Datei sind üblicherweise keine Änderungen notwendig.

Dieses Widget ist die *Zentrale* zur Bedienung aller Schaltfunktionen in allen Schaltmodulen. Am Beispiel der Datei *swstd.lua* sieht man eine Gliederung der Funktionen in Gruppen zu je 8 Funktionen, die sich

auf einem **RC-Quad-D** befinden. Für jeden Ausgang des **RC-Quad-D** können dann die unterschiedlichen Zustände ausgewählt werden.

Die Anwahl der Funktion und des Zustandes erfolgt mit **T5**-hoch und -herunter, die Selektion erfolgt mit **T6**-hoch.

Um auf die nächste Seite (für das nä. Schaltmodul **RC-Quad-D**) gelangen, navigiert man mit **T5**-herunter bis unter die letzte Zeile, oder mit **T5**-hoch bis vor die erste Zeile des aktuellen Menüs. Damit wird auf die anderen Menuseiten (andere **RC-Quad-D**-Module) umgeschaltet.

Alternativ kann mit dem 6-Positionentaster **6pos** umgeschaltet werden (zwischen den *ersten* 6 Seiten / Modulen).

Zusätzlich steht **LS** zur schnellen Auswahl der Menuzeile und **RS** für die Spalte zur Verfügung.

### 7.4.3. Das Widget **WMSWC/main.lua** (Konfiguration)

In dieser Datei sind üblicherweise keine Änderungen notwendig.

Dieses Widget dient zur *Konfiguration* aller Parameter jeder Funktion wie dem *Anlernen* von Übertragungskanal sowie der Adresse eines Moduls

Die einzelnen Parameter des Menüs lassen die *nur* mit **T5**-hoch und -herunter anwählen sowie **T6**-hoch selektieren. *Nach* der Selektion wird das Potentiometer **S1** zum Einstellen des Parameterwertes verwendet.

*Übersicht 1. Die Bedeutung der Parameter für das Modul **RC-Quad-D***

- **RES**: Reset aller Werte der Funktion (Wert *muss* auf **1** stehen, damit der Reset ausgeführt wird)
- **PWN**: Einstellen PWM-*Startrampe* beim *Anlaufen* einer Funktion (1=schnell, 31=langsam).
- **B1/I**: PWM-Grad für Richtung *vorwärts*. Der Kanal wird für 1 Sekunde eingeschaltet, dabei wird der Betriebsstrom gelernt.
- **B1/D**: Strombegrenzung für Richtung *vorwärts*. Dies ist *relativ* zum *gelernten* Strom.
- **B2/I**: PWM-Grad für Richtung *rückwärts*.
- **B2/D**: Strombegrenzung für Richtung *rückwärts*. Dies ist *relativ* zum *gelernten* Strom.
- **PT**: Der sog. PassThrough-Kanal: ein beliebiger **OpenTX**-Kanal (**1 ... 16**) kann zur Weiterleitung an den Ausgang des **RC-Quad-D** ausgewählt werden. Damit steuert man den PWM-Grad in den Grenzen, die durch **B1/I** und **B2/I** gesetzt wurden.
- **Ck**: Stromkennlinie für die Strombegrenzung im *PassThrough*-Betrieb (1=starke Abhängigkeit, 31=keine Abhängigkeit).

Es existiert eine zusätzliche Seite (am Ende, bzw. vor dem Anfang der Menu-Seiten, da diese zyklisch sind) mit modul-globalen Einstellungen.

*Übersicht 2. Bedeutung der modul-globalen Parameter*

- **Learn Ch/Adr**: Anlernen der Moduladresse und des Übertragungskanals (s.a. [Abschnitt 9.2](#))
- **TMpx**: (ab Firmware Version **V20**) Sensor-Id für die Zustandsübermittlung (nur **S.Port**)
- **TMode**: irrelevant
- **OMpx**: irrelevant

## 7.5. Physische Schalter und die Schaltzustände

In den Konfigurationsdateien für die Modelle, also etwa **MODELS/ABC.lua** für das Modell mit dem Namen **ABC** kann man den einzelnen Funktionen phys. Schalter mit Hilfe ihrer *Namen* zuweisen. Die Namen sind **sa**, **sb**, ... **se**, **sg** der 3-Positionen-Schalter. Damit kann man die ersten drei Zustände **aus**, **ein** und **blink1** erreichen.

### 7.5.1. ShortCuts / Abkürzungen

Wird *einer* Funktionen *genau* ein Schalter zugewiesen, so ist dieser Schalter ein *ShortCut* für die ersten drei Zustände dieser Funktion.

### 7.5.2. Overlays / Ebenenumschaltung

Werden *mehreren* Funktionen *derselbe* Schalter zugewiesen, so ist dieser Schalter ein *Overlay* für die ersten drei Zustände dieser Funktionen. Dies bedeutet, dass dieses phys. Schalter nur dann aktiv sind, wenn das entsprechende Menu *ausgewählt* ist (etwa durch **6pos**).

## 8. Einbau

### 8.1. Autoerkennung: IBus oder SBus



*Änderung in Hardware Version 4 / ab Firmware V20*

Ab der o.g. Hardware Version in Zusammenhang mit der o.g. Firmwareversion ist das Modul universell für den Betrieb an **IBus** oder **SBus** (invertiert *oder* nicht-invertiert) geeignet.

Alte Dokumentation: [Appendix A](#)

Schließen Sie das Modul am **IBus-Servo** oder **SBus** mit dem als **Servo** gekennzeichneten Verbinder an.

Schließen Sie das Modul am **IBus-Sensor** oder **S.Port** mit dem als **SIn** gekennzeichneten Verbinder an.

Bei **IBus** können Sie weitere Sensoren am **SOut**-Pin-Header anschließen.



*SBus ist nicht gleich SBus*

Leider ist der *Werte-Bereich*, in denen die einzelnen Kanäle eines SBus-Signal skaliert werden, nicht wohl definiert, und einige Hersteller legen unterschiedliche Grenzen des Wertebereiches fest. Dies führt zu Problemen!

Als Referenz wurde der Wertebereich eines **FrSky R8X**- Empfängers zu Grunde gelegt.

Abweichend hiervon sind:

- **FlySky FS-iA4B** in der Einstellung **SBus, PWM**
- **FlySky FS-iA6B** in der Einstellung **SBus, PWM**
- **FlySky FS-iA10B** in der Einstellung **SBus, PWM**

Mit den o.g. Empfängern funktioniert das Modul derzeit nur im Modus **IBus, PWM** oder **IBus, PPM**.

### 8.2. Anschluß an den Akku

Achten Sie auf richtige Polung (auf der Platine markiert).

### 8.3. Schutz

Um das Modul gegen Feuchtigkeit zu schützen, empfiehlt es sich, das Modul mit Polyurethan-Lack-Schutzlack (z.B. Kontakt 70) zu überziehen. Bitte kleben Sie jedoch vorher die Pfostenverbinder für alle externen Kontakte und die Pads für die weiteren Anschlußkabel ab. Schützen Sie auch ggf. vorhandene Schraubklemmen.

Ein Einschrumpfen mit Schrumpfschlauch ist auch möglich.



## 9. Erste Einrichtung und Anlernen

Im Auslieferungszustand sind die **RC-Quad-D** auf den Übertragungskanal **10** und die Adresse **0** eingestellt. Damit kann an einem Übertragungskanal natürlich nur ein Modul betrieben werden (jedes Modul *muss* eine eigene eindeutige Adresse haben).

### 9.1. Auto-Erkennung von **IBus** und **SBus**

Das Modul erkennt automatisch, ob es an einem **IBus** oder **SBus** (invertierend oder nicht-invertierend) angeschlossen ist. Die Telemetrie wird dazu jeweils passend eingestellt. Bis **Bus-LED** zeigt dies entsprechend an.

- Direkt nach dem Einschalten leuchtet die **Bus**-LED mindestens **0,5s** dauerhaft.
- Sollte kein **Bus**-System erkannt werden, so bleibt die **Bus**-LED dauerhaft an.
- Je nach erkanntem **Bus**-System gibt es folgendes Blink-Muster:
  - **IBus**: 1x kurzes Aufleuchten
  - **SBus** (invertiert): 2x kurzes Aufleuchten
  - **SBus** (nicht-invertiert): 3x kurzes Aufleuchten
  - **SumD**: 4x kurzes Aufleuchten
- Entsprechend dem erkanntem **Bus**-System wird die Telemetrie gewählt:
  - **IBus** → **IBus**-Telemetrie
  - **SBus** → **S.Port**
  - **SBus** → **S.Port**
  - **SumD** → **Hott**-Telemetrie

Diese Konfiguration bleibt bis zum Ausschalten erhalten. Bei jedem Einschalten wird erneut detektiert.

### 9.2. Kanal und Adresse

Das *Anlernen* von *Kanal* und *Adresse* geschieht in *einem* Schritt. Dazu *muss* das zu konfigurierende **RC-Quad-D** *allein* am betreffenden **IBus** oder **SBus** hängen. Weiterhin sollten außer dem zu konfigurierenden Übertragungskanal für die **RC-Quad-D**-Module alle anderen Übertragungskanäle im Sender *deaktiviert* werden (keine *Mischer* aktivieren).



#### *Spezielles Modell für das Anlernen der Module*

Um den Anlernvorgang zu erleichtern, kann man sich ein spezielles *Modell* nur für das Anlernen einrichten. Dieses Modell wird *niemals* für die Steuerung eines realen Modells verwendet. Bei diesem Modell sind *alle* Übertragungskanäle bis auf den Kanal für die digitalen Schaltmodule deaktiviert (keine *Mischer*).

Natürlich muss man dann im *realen* Modell *denselben* Übertragungskanal konfigurieren (*Mischer*).



### Anschluß des Moduls zum Anlernen

- Dass anzulernende Modul *muss allein* am IBus/SBus angeschlossen sein.
- An den Ausgängen des Moduls dürfen keine Verbraucher angeschlossen sein (aus Sicherheitsgründen).

### Ablauf:

1. Sender und Empfänger sowie Schaltmodule aus.
2. Ein RC-Quad-D an den IBus oder SBus anschließen.
3. Die Mischer-Skripte `encsbm.lua`, `encxjt.lua` oder `encib.lua` für den gewünschten Übertragungskanal einrichten (s.o.).
4. Sender einschalten.
5. Konfigurationsmenu auswählen
6. Im Konfigurationsmenu auf der *letzten* Seite (modul-spezifische Einstellungen) den Parameter **Learn Ch/Adr** selektieren. Oben rechts muss dann der Wert dunkel unterlegt sein. Mit dem Potentiometer **S1** die Adresse einstellen (Wert 1 ... 8 bedeutet Adresse 1 ... 8).
7. Empfänger einschalten
  - a. Die Logik des RC-Quad-D wird über den Empfänger versorgt. Ein Betriebsspannung für die Motoren ist zunächst nicht erforderlich.
  - b. Die Kontroll-LED leuchtet für 3 Sekunden, anschließend blinkt sie 1 ... 8 mal, je nach eingestellter Adresse.
  - c. Sollte dies nicht passieren, so wurde der Übertragungskanal nicht gefunden.
8. Nach ca. 2 weiteren Sekunden das RC-Quad-D zusammen mit dem Empfänger ausschalten.
9. Für weitere RC-Quad-D Module die nä. Adresse am Sender einstellen und selektieren.
10. Das nä. RC-Quad-D anschließen und einschalten
11. ...

### Kontrolle:

Nach dem Anlernen von Kanal und Adresse kann man auch sofort ins Bedien-Menu wechseln, dort die Seite für die Modul-Adresse auswählen und kontrollieren, ob sich das Schaltmodul ansprechen lässt. Schalten Sie bspw. die erste Funktion in den Zustand **ein**.

Das Modul ist zur Funktionskontrolle natürlich mit einer externen Betriebsspannung zu versorgen und ein Verbraucher ist anzuschließen.

Andernfalls ist der Anlernvorgang zu wiederholen.

# 10. Funktion

## 10.1. Bedienung der Schaltfunktionen

Mit dem Widget **WM MultiSwitch** können Schaltfunktionen ausgewählt werden.

Dies geschieht im Normalfall über **T5 - Up** und **T5 - Down** zur Navigation durch die Zeilen und Spalten des Menus. Mit **T6 - Up** kann eine Funktion ausgewählt (selektiert) werden.

Sind die beiden Schieber **LS** und **RS** vorhanden und erkannt worden, so kann mit **LS** in den Zeilen gescrollt werden und mit **RS** in den Spalten des Menus.

Sind im Menu *mehr* als eine Seite (also *mehr* als 8 Schaltfunktionen bzw. mehr als ein Schaltmodul) enthalten, so kommt man auf die nächste Seite, indem man auf den Eintrag *ganz unten rechts* navigiert und dann noch einmal nach unten navigiert (**T5 - Down**). Entsprechend auf die vorige Seite.

Wird der 6-Positionenschalter **6pos** erkannt, so kann damit schnell zwischen den ersten sechs Menu-Seiten navigiert werden.

### *ShortCuts*

Haben Schaltfunktionen *ShortCuts* zugeordnet (*ein* phys. Schalter: **sa ... sh**), so können diese Funktionen und die *ersten drei* Zustände mit diesem Schalter aktiviert werden. Dies gilt *unabhängig* von der gerade angezeigten Menu-Seite. Daher der Name *ShortCut* (Abkürzung). Oft benötigte Funktionen sollte also mit einem *eindeutigen* Schalter belegt werden.

Der Name des Schalters (bspw.: **sg**) wird dann nach dem Funktionsnamen (bspw.: **Fun A**) angegeben: dann **Fun A/sg**.

### *Overlays / Ebenenumschaltung*

Wurde eine Schalter *mehreren* Funktionen zugordnet, so wird zu einem *Overlay*. Dies wird dann mit einem *zusätzlichen !* nach dem Namen gekennzeichnet: etwa **Fun A/sb!** und **Nuf C/sb!**. Der Schalter **sb** schaltet die Zustände von **Fun A** nur dann, wenn die Menu-Seite mit **Fun A** aktiv (sichtbar) ist. Ist die Seite mit **Nuf C** sichtbar, so schaltet der Schalter **sb** die Zustände von **Nuf C**.

## 10.2. Konfiguration

Mit dem Widget **WM MultiSwitch Config** können Schaltfunktionen konfiguriert werden.

Dies über **T5 - Up** und **T5 - Down** zur Navigation durch die Zeilen und Spalten des Menus. Mit **T6 - Up** kann eine Funktion ausgewählt (selektiert) werden.



### *Keine unmittelbare Rückmeldung*

Im Gegensatz zu vielen anderen technischen Geräten, die über ein Menu-System konfiguriert werden, gibt es im Konfigurationsmenu hier nach der *Selektion* und dem *Einstellen* eine *Parameterwertes* keine besondere *Rückmeldung* an den Benutzer über den Erfolg oder Misserfolg! Daher ist hier sorgsam vorzugehen.

Wird jedoch ein **RC-MultiAdapter-DA** über das Konfigurationsmenu *angesprochen*, so leuchtet die Kontroll-LED auf, um anzuzeigen, dass ein Parameter des Moduls konfiguriert wird. Wechselt man wieder in das Funktionsmenu, so muss die Kontroll-LED erlöschen.



### *Delektion nach dem Einstellen eines Parameters*

Als Schutz vor einem unbeabsichtigten Ändern eines Parameterwertes, sollte man folgendermaßen vorgehen:

1. Zum Parameter navigieren
2. Parameter selektieren (oben rechts mit die Wertanzeige *invertiert* erscheinen).
3. Parameterwert einstellen
4. **Zu einem anderen Parameter navigieren (ohne ihn zu selektieren).** Die Invertierung der Wertanzeige ist beendet.

Der letzte Schritt hebt die vorige Selektion auf. Damit kann nichts mehr unbeabsichtigt verändert werden.

# 11. Details: Einstellungen im Einzelnen

Nach dem Anlernen des *Übertragungskanal*s und der *Adresse* des Moduls kann diese sofort über das Steuerungs-Widget bedient werden (sa. [Abschnitt 11.1](#)).

## 11.1. Betrieb ohne Abschaltung

Im Auslieferungszustand oder nach einem *Reset* der Funktion findet (ohne weitere Beschaltung (s.a. [Abschnitt 11.2.1](#)) **keine** Abschaltung der Funktion statt. Die Motoren laufen also ggf. in eine mechanische Begrenzung und können dadurch *Schaden* nehmen. Dies ist deswegen unbedingt zu vermeiden.

Natürlich gibt es auch Anwendungsfälle, in denen eine Funktionen *keine mechanische* Begrenzung ausweist. Die könnte sein:

- Betrieb eines Mikro-Motors zum Drehen eines Radarbalkens
- Querstrahlruder
- nicht-mechanische Anwendungen (Helligkeitssteuerung von Glühlampen und LEDs).

Für alle diese Anwendungsfälle kann das Modul jedoch trotzdem sinnvoll eingesetzt werden:

- man hat *vier* PWM-Steller (Fahrtenregler) mit Telemetrie (Strom) auf einem Modul.
- man hat eine Anlaufsteuerung (langsames Hochfahren bis auf Nenndrehzahl).
- man kann kleinere Motoren mit einer hohen Versorgungsspannung betreiben.

## 11.2. Betrieb mit Abschaltung

Der Betrieb *mit* einer Abschaltung der Motoren benötigt eine weitere Konfiguration / Verkabelung.

### 11.2.1. Abschaltung durch Endlagenkontakte

Die *einfachste* Form der Endlagenabschaltung eines jeden Kanals ist diejenige, die je Kanala *zwei* Endlagenkontakte verwenden. Wird *einer* der beiden Kontakte *geschlossen*, so wird der Motor unmittelbar abgeschaltet. Ein erneutes Anlaufen ist dann nur in die *Gegenrichtung* möglich. Bei Kontakte werden *parallel* angeschlossen.

[end1] | *end1.jpg*

*Abbildung 20. Anschluss der Endlagenschalter*

Der Ablauf ist damit der folgende:

- Einschalten der Funktion am Widget (Richtung **A**).
- Die Funktion erreicht die Endlage und der Endlagenkontakt wird ausgelöst (schließt).
- Die Funktion wird abgeschaltet (Motor stoppt).
- Ein Wiedereinschalten der Funktion in *dieselbe* Richtung **A** ist *nicht* möglich.
- Die Funktion kann nur in die *andere* Richtung **B** wieder gestartet werden.
- Wird die Funktion nicht zuvor gestoppt und erreicht auch hier die Endlage, so wiederholt sich der

Ablauf für die Richtung **B**.

Alle diese Zustandsänderungen lassen sich auch durch Telemetrie beobachten (s.a. [Abschnitt 11.4.2](#)).

### 11.2.2. Abschaltung durch Stromüberwachung

Diese Betriebsart wird *aktiviert* durch ein Anlernen des Abschaltstromes.



#### *Motorstrom ~ Drehmoment*

Das Drehmoment eines Elektromotors ist direkt proportional zum Strom, der die Wicklung des Motors durchfließt. Wird ein laufende Elektromotor abgebremst durch äußere Einflüsse, so sinkt seine Drehzahl. Dies bedeutet (durch Absinken der Gegen-EMK) eine *Erhöhung* des Motorstromes. Wie stark diese Erhöhung ist, hängt wiederum von *Innenwiderstand* des Motors und anderer Faktoren ab. Der Motor produziert ein erhöhtes Drehmoment.

Bremst man den Motor zum Stillstand ab (mechanische Begrenzung), so ist dieser Anstieg am stärksten.

Allerdings wird dann die gesamte zugeführte elektrische Leistung als *Verlustleistung* in der Wicklung des Motors in *Wärme* umgesetzt. Dies kann einen Motor zerstören (kleine Motoren meistens schneller als große Motoren, da deren Wärmekapazität kleiner ist).

Um ein *zuverlässiges* und *schonendes* Abschalten zu ermöglichen, wenn der Motor einer Funktion in eine mechanische Begrenzung läuft, muss der *niedrigste* Abschaltstrom eingestellt werden, mit dem trotzdem ein sicherer Normalbetrieb möglich ist. Das Modul überwacht permanent den Stromfluss durch den Motor und schaltet den Motor unmittelbar nach Überschreiten des *Abschaltstromes* aus. Ist der Abschaltstrom zu niedrig, so kann es durch ungünstige äußere Einflüsse auch im Normalbetrieb dazu kommen, dass abgeschaltet wird. Etwa bei tiefen Umgebungstemperaturen läuft eine Mechanik durch die sich verändernde Öl-Viskosität schwergängiger als bei höheren Temperaturen. Zudem hängt dies auch oft mit der Laufrichtung der Mechanik / des Motors zusammen.

Auf der anderen Seite kommt beim *Anlaufen* eines Motors zu einem *erhöhten* Motorstrom. Dieser Effekt kann durch die zu beschleunigenden Schwungmassen noch verstärkt werden. Aus diesem Grunde bietet das Modul die Möglichkeit, ein sog. *Anlaufphase* festzulegen. In dieser Phase findet *keine* strombedingte Abschaltung statt. Die Dauer dieser Phase sollte daher so *kurz wie möglich* sein.

Das Bild [Abbildung 21](#) verdeutlicht (hoffentlich) die Zusammenhänge.

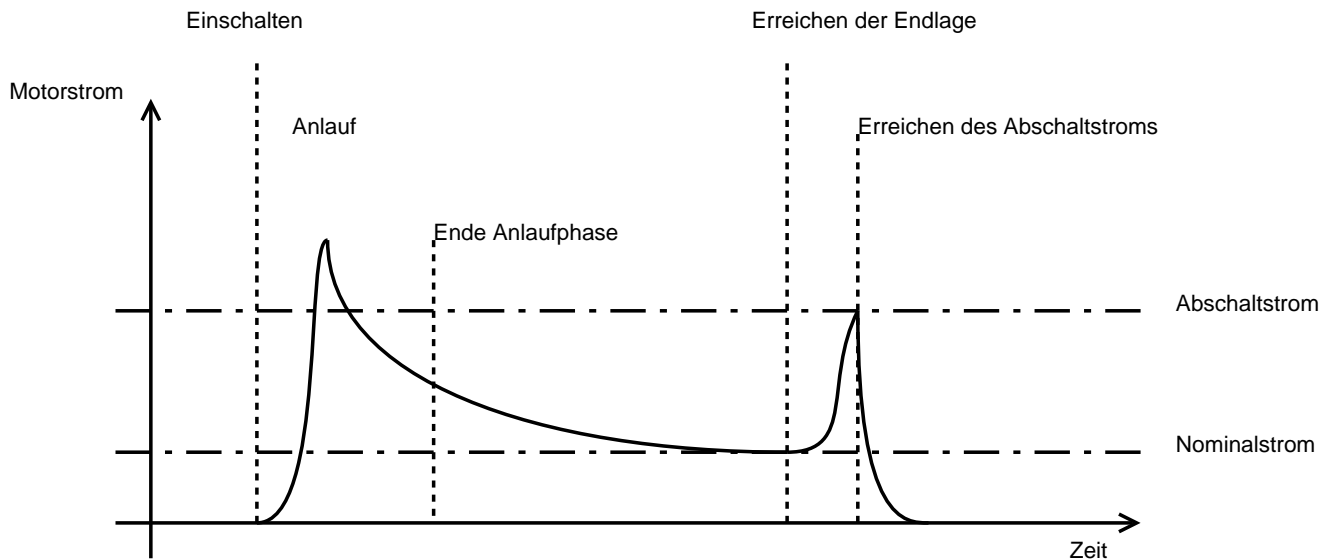


Abbildung 21. Zeitlicher Verlauf des Motorstromes

Das Einstellen eines Abschaltstromes ist also immer ein *Kompromiss*.

Daher **muss** der *normale* Betriebsstrom der Funktion gelernt werden. Anschließend kann daraus der *Abschaltstrom* abgeleitet werden. Dies geschieht dann für beide Richtungen **A** und **B** getrennt bei jeder Funktion.

Man geht hier schrittweise vor: zunächst wird für *eine* Funktion und *eine* Richtung eine Einstellung der Parameter im Konfigurations-Widget vorgenommen.

Je Funktion und Richtung wird folgendermaßen verfahren (**bitte** lesen Sie den folgenden Ablauf komplett durch, **bevor** Sie eine Funktion im Einbauzustand testen):

1. Parameter **B1/I**: Einstellen des PWM-Grades (Geschwindigkeit / Helligkeit) und damit auch *Anlernen* des für den *Normalbetrieb* notwendigen Stromes. Der PWM-Grade kann mit Potentiometer **S1** eingestellt werden (s.a. [Abschnitt 7.4.3](#)).
2. Der Motor läuft *nach der Selektion* dieses Parameters für 1s in Richtung **A**. Bitte sorgen sie für einen blockadefreien Lauf.
3. Der Motor schaltet selbständig ab und der *Nominalstrom* ist erlernt.
4. *Ohne* eine weitere Änderung ist nun der *Abschaltstrom* das 1,5-fache des Nominalstromes.
5. Mit dieser Einstellung sollte die Funktion jetzt würde das *Bedienwidget* getestet werden. Bitte achten Sie auf eine Abschaltung: dies wird durch Aufleuchten der Kontroll-LED angezeigt.
6. Findet eine sofortige Abschaltung statt (LED), so kann entweder der Abschaltstrom zu gering sein, oder - besonders bei größeren Motoren - die Abschaltung ist maßgeblich durch den *Anlaufvorgang* herbei geführt worden.
7. Findet eine Abschaltung (LED) statt, *ohne* dass die Mechanik unter zu hohe Last kommt, so ist die Einstellung brauchbar.
8. Findet **keine** Abschaltung statt, bedeutet dies eine Überlastsituation sowohl für den Motor als auch für die Mechanik. Es **muss** nun schnellstmöglich eine Abschaltung **manuell** durchgeführt werden.
9. Sollte die Abschaltung nicht oder zu spät erfolgen, wechseln Sie wieder in des Konfigurations-Widget und verändern Sie den Abschaltstrom zu kleineren Werten. Andernfalls vergrößern Sie den Wert.



10. Parameter **B1/D**: Zum Verkleinern des Abschaltstromes und damit zum früheren Abschalten stellen Sie schrittweise kleinere Werte ein (8: Standardwert).
11. Nach einer Veränderung testen Sie wieder!
12. Erfolgt die Abschaltung *sofort*, so kann auch ein *erhöhter Anlaufstrom* dafür die Ursache sein. Um dies als Ursache auszuschließen, stellen Sie für die Anlaufphase eine sicher ausreichende Zeitspanne ein. Bei kleinen Motoren etwa 600ms, bei mittleren Motoren etwa 1200ms und bei großen Motoren etwa 2400ms.
13. Parameter **PWM**: hiermit wird die Anlaufzeit in *Vielfachen* von 200ms eingestellt: der Wert von 3 ergibt also 600ms.
14. Wiederholen Sie nun den Test (zunächst ohne den Abschaltstrom verändert zu haben).
15. Achten Sie auch den Motor: erreicht der Motor in der Anlaufphase schon seine nominelle Drehzahl?
16. Hat der Motor in der *Anlaufphase* seine nominelle Drehzahl erreicht und erfolgt die Abschaltung anschließend, so ist die Anlaufzeit ausreichend aber der Abschaltstrom zu niedrig. Vergrößern Sie daher den Abschaltstrom und testen Sie wieder (*ohne* die Anlaufdauer zu verändern).
17. Andernfalls verlängern Sie die Anlaufphase weiter.
18. Haben Sie eine befriedigende Einstellung für den Abschaltstrom gefunden, kann es erforderlich sein, die Anlaufphase wieder zu verkürzen.



Das obige Vorgehen erscheint sehr komplex. Vergleichen Sie daher das Vorgehen mit der grafischen Darstellung in [Abbildung 21](#).

## 11.3. Proportionaler Betrieb

## 11.4. Telemetrie

Das Modul kann an den jeweiligen *Telemetrie*-Bus angeschlossen werden (**IBus**, **SBus**).

Das Modul stellt dabei die aktuellen *Stromwerte* eines jeden Kanals und auch den *Funktionszustand* eines jeden Kanals zur Verfügung.

Sollten die Sensorwerte des Moduls nicht in der Telemetrie des **OpenTx**-Systems *nicht* sichtbar sein, so starten die *Sensorsuche* im *Telemetrie-Menu* des Senders erneut. Nach dem Erscheinen der Sensorwerte *stoppen* sie die Sensorsuche wieder.

### 11.4.1. Funktionsströme

Die Ströme jedes Funktionskanals werden als Sensortyp *Strom* (**curr**) angezeigt. Die Zuordnung zu den einzelnen Ausgängen des Moduls erfolgt *normalerweise* in *aufsteigender* Reihenfolge. Dies ist aber durchaus davon abhängig, welche *weiteren* Sensoren am Sensorbus angeschlossen sind.



Es ist daher empfehlenswert, die Telemetrieseiten im Sender erst dann endgültig einzurichten, wenn die *Verkabelung* im Modell abgeschlossen ist. Insbesondere sollte die Position in der **IBus-daisy-chain** nicht mehr geändert werden.

## 11.4.2. Funktionszustände

In den *üblichen* Telemetrieprotokollen existieren meistens *keine* Sensortypen, um einheitenlose Zustandsinformation zu übertragen. Daher werden folgende Sensoren verwendet:

### *SBus/S.Port*

Für ein **S.Port**-System werden sog. *DIY*-Typen verwendet.

### *IBus*

Für ein **IBus**-System werden Sensoren des Typs *Flightmode (FM)* verwendet.

Die Zustände einer *jeden* Funktion lassen sich auch via Telemetrie beobachten.

Table 1. Tabelle der Zustands-Codes

Code	Bedeutung
1000	Initialisierung
1001	Aus
1002	Aus (wartend)
1010	An (Richtung A)
1012	An (Richtung B)
1011	Anlauf (Richtung A)
1013	Anlauf (Richtung B)
102x	(intern)
103x	(intern)
1040	Stromabschaltung (Richtung A)
1041	Stromabschaltung (Richtung B)
1042	Kontaktabschaltung (Richtung A)
1043	Kontaktabschaltung (Richtung B)
1050	Betriebsart <b>passthru</b> (Richtung A)
1051	Betriebsart <b>passthru</b> (Richtung B)
1052	Anlauf Betriebsart <b>passthru</b> (Richtung A)
1053	Anlauf Betriebsart <b>passthru</b> (Richtung B)

## 12. Betrieb



Beachten Sie unbedingt die Anweisungen unter [Abschnitt 9](#).



Die üblichen Sicherheitsvorkehrungen im Betrieb mit ferngesteuerten Modellen, insbesondere Schiffsmoellen sind einzuhalten.



Beachten Sie **alle** folgenden Hinweise zum Betrieb.



Eine Verwendung des Moduls in Rennbooten oder Flugmodellen ist nicht zulässig.



Das Modul darf nicht in Kontakt mit Wasser, Wasserdampf oder anderen Flüssigkeiten kommen. Wasser oder Wasserdampf bzw. andere Flüssigkeiten können zu einem Totalausfall und damit zu einem Modellverlust sowie Personenschäden führen.



Das Modul verbraucht im Ruhezustand nur sehr wenig Strom. Trotzdem darf ein dauerhafter Anschluß an einen **unüberwachten** Akku nicht erfolgen. Hier besteht Brandgefahr! Gefahr von Personenschäden!



Beim Betrieb ist die Erwärmung des Moduls zwingend zu überwachen! Eine Überhitzung kann zu einem Totalausfall und damit zu einem Modellverlust führen. Gefahr von Personenschäden!



Die Spannungsversorgung ist Modul ist im Betrieb zu überwachen. Bei Unterspannung kann das Modul abschalten oder bei gleichzeitiger hoher Stromaufnahme überhitzen und so zu einem Totalausfall und damit zu einem Modellverlust sowie Personenschäden führen



Die erforderlichen Kabelquerschnitte für die Verbindung mit dem Akku und auch mit dem elektrischen Verbraucher sind unbedingt einzuhalten. Hier besteht Brandgefahr. Gefahr von Personenschäden!



Beim Betrieb ist der maximale Stromdurchfluß zu begrenzen und zu überwachen. Ein zu langer und zu hoher Stromfluß kann zu einem Totalausfall und damit zu einem Modellverlust sowie Personenschäden führen.



Das Modul ist nicht kurzschlußfest. Ein Kurzschluß führt zu einem Totalausfall und damit zu einem Modellverlust sowie Personenschäden.



Der maximale Schaltstrom ist unbedingt einzuhalten und darf nicht überschritten werden. Ein zu hoher Schaltstrom kann zu einem Totalausfall und damit zu einem Modellverlust sowie Personenschäden führen.



Die Kapazitäten (Elkos, Siebelkos) am Ausgang des Moduls, etwa in Fahrtreglern (Stellern) für Motoren, dürfen 10.000µF nicht überschreiten. Zu hohe Kapazitäten können zu einem Totalausfall und damit zu einem Modellverlust sowie Personenschäden führen.



Das Modul darf keinen Vibrationen ausgesetzt werden. Treffen Sie entsprechende Vorkehrungen zu einem vibrationsgeschützten Einbau. Zu starke Vibrationen können zu einem Totalausfall und damit zu einem Modellverlust sowie Personenschäden führen.



Das Modul darf nur innerhalb eines Temperaturbereiches von -10°C bis +55°C betrieben werden. Ein Betrieb außerhalb dieses Bereiches kann zu einem Totalausfall und damit zu einem Modellverlust sowie Personenschäden führen.

## 13. Fehlersuche und Tests

Das **RC-Quad-D** kann dem Benutzer keine direkte Rückmeldung über ihre Einstellungen geben. Daher ist die Fehlersuche auf ein klar strukturiertes Vorgehen angewiesen.

Die häufigsten Fehlerquellen sind:

- Unterscheidung **IBus** / **SBus**
  - Unterscheidung **SBus** mit **XJT**-Sendemodul oder MPM-Sendemodul.
- **SBus**-Inverter
- Addressvergabe

### 13.1. Checkliste **IBus**

**IBus** ist ein Bus-System der Fa. *FlySky* und kommt daher nur bei *FlySky*-Empfängern vor.

#### Richtiger **IBus**-Ausgang?

An den Empfängern gibt es zwei **IBus**-Ausgänge: **IBus-Servo** und **IBus-Sensor**. Es muss der **IBus-Servo** Ausgang gewählt werden.

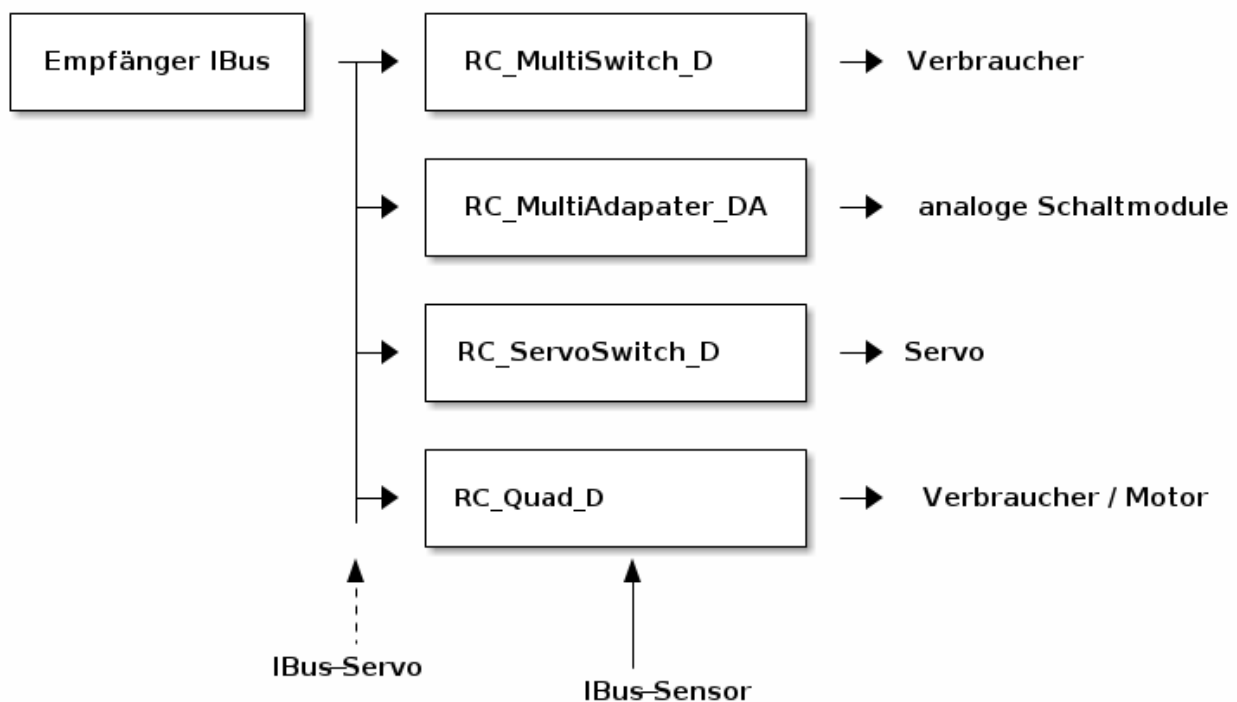


Abbildung 22. Aufbau mit einem Empfänger, der **IBus-Servo** liefert

#### Korrekte Verkabelung?

Anders als beim **IBus-Sensor** werden beim **IBus-Servo** - wie bei einem Bus-System üblich - alle Teilnehmer *parallel* geschaltet.

#### Richtige Konfiguration Multi-Protokol-HF-Modul?

Das MPM im Sender muss auf **MULTI FlySky\_2A PWM,IBUS** oder **MULTI FlySky\_2A PWM,IB16** eingestellt sein.

### Richtige Konfiguration des Models?

In den Konfigurationsdateien muss *ebenfalls* der richtige Bus-Typ ausgewählt werden. Dies geschieht am Anfang der Konfigurationsdatei.

Quelltext 2. Ausschnitt aus der Datei `swstd.lua`, `swstdm.lua` oder `swstds.lua`

```
1 local name = "Default";
2
3 local gVariable = 5;
4
5 local useSbus = 0; -- only 4 states, only 16 parameter values ①
6
7 local gstates1 = {"aus", "ein", "blink1", "blink2"};
8 local gstates2 = {"aus", "ein 1", "ein 2"};----
```

① Richtige Einstellung für IBus ist: `useSbus = 0`

Achten Sie auf die korrekte Auswahl der Konfigurationsdatei (ggf. Name des Modells): der verwendete Name wird am Fuß des *Widget* angezeigt.

### Korrektes Mixer-Script?

Es muss das Script **encib** eingerichtet sein.

## 13.2. Checkliste SBus

SBus wird von vielen Herstellern als Bus-System verwendet, so auch von der Fa. FrSky.

### Richtiger SBus-Ausgang?

Viele SBus-fähige Empfänger haben auch eine Telemetrie-Möglichkeit. Der dafür verwendete Bus heißt **S.Port** und ist anders am Empfänger gekennzeichnet.

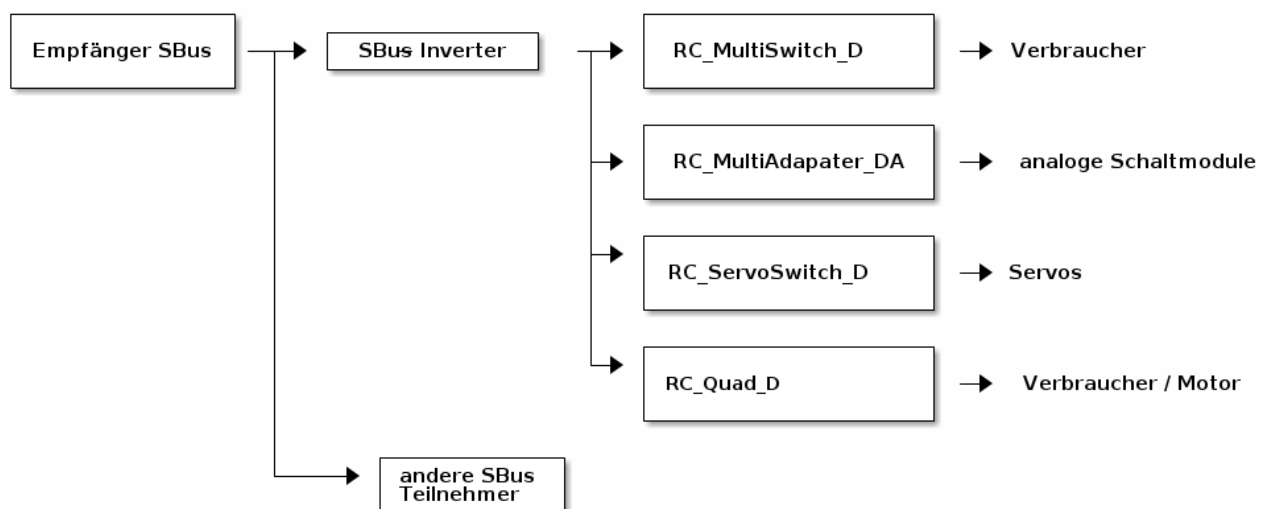


Abbildung 23. Aufbau mit einem Empfänger, der SBus-Servo liefert

### Korrekte Verkabelung?

Für das **RC-Quad-D** benötigt man einen **SBus-Inverter**. *Andere SBus-Teilnehmer* sind nach wie vor *vor* dem Inverter anzuschließen.

### Richtige Konfiguration des Modells?

In den Konfigurationsdateien muss *ebenfalls* der richtige Bus-Typ ausgewählt werden. Dies geschieht am Anfang der Konfigurationsdatei.

Quelltext 3. Ausschnitt aus der Datei `swstd.lua`, `swstdm.lua` oder `swstds.lua`

```
1 local name = "Default";
2
3 local gVariable = 5;
4
5 local useSbus = 1; -- only 4 states, only 16 parameter values ①
6
7 local gstates1 = {"aus", "ein", "blink1", "blink2"};
8 local gstates2 = {"aus", "ein 1", "ein 2"};----
```

① Richtige Einstellung für **SBus** ist: `useSbus = 1`

Achten Sie auf die korrekte Auswahl der Konfigurationsdatei (ggf. Name des Modells): der verwendete Name wird am Fuß des *Widget* angezeigt.

### Korrektes Mixer-Script?

Je nach verwendetem HF-Modul im Sender ist eine Unterscheidung zu treffen:

**XJT-Modul** es muss das Script `encxjt` eingerichtet sein.

**MPM-Modul** es muss das Script `encsbm` eingerichtet sein.



# 14. Veraltete Informationen

## Appendix A

# Appendix A: Einbau

## A.1. IBus oder SBus

Die Module sind fest programmiert für **IBus** oder **SBus**-Eingangssignale.

**Zusätzlich** braucht man für den Anschluß an ein **SBus** noch einen **Signal-Inverter**. Dieser muss in die Zuleitung vom **SBus** zum **RC Quad-D** oder **RC MultiAdapter-DA** oder **RC-MultiSwitch-D** eingeschleift werden (wie ein Verlängerungskabel).

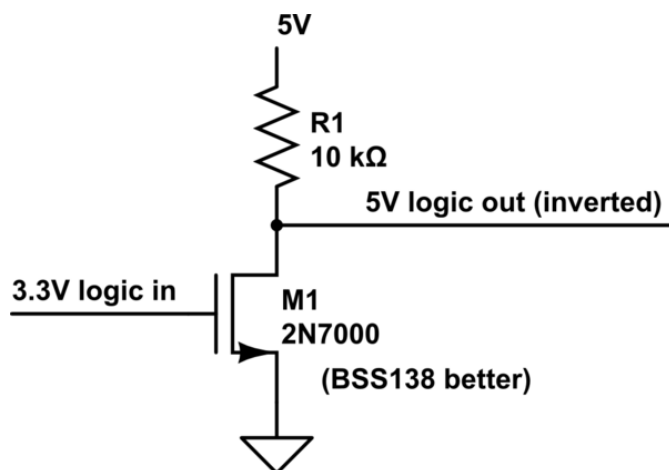


Abbildung 24. Einfacher Inverter mit einem Kleinsignal-MosFet

Ein Inverter wie in [Abbildung 24](#) kann einfach in ein Servokabel eingebaut werden.

### *SBus ist nicht gleich SBus*

Leider ist der *Werte-Bereich*, in denen die einzelnen Kanäle eines SBus-Signal skaliert werden, nicht wohl definiert, und einige Hersteller legen unterschiedliche Grenzen des Wertebereiches fest. Dies führt zu Problemen!



Als Referenz wurde der Wertebereich eines **FrSky R8X**- Empfängers zu Grunde gelegt.

Abweichend hiervon sind:

- **FlySky FS-iA4B** in der Einstellung **SBus, PWM**
- **FlySky FS-iA6B** in der Einstellung **SBus, PWM**
- **FlySky FS-iA10B** in der Einstellung **SBus, PWM**

Mit den o.g. Empfängern funktioniert das Modul derzeit nur im Modus **IBus, PWM** oder **IBus, PPM**.

## A.2. Anschluß an den Akku

Achten Sie auf richtige Polung (auf der Platine markiert).

## A.3. Schutz

Um das Modul gegen Feuchtigkeit zu schützen, empfiehlt es sich, das Modul mit Polyurethan-Lack-

Schutzlack (z.B. Kontakt 70) zu überziehen. Bitte kleben Sie jedoch vorher die Pfostenverbinder für alle externen Kontakte und die Pads für die weiteren Anschlußkabel ab. Schützen Sie auch ggf. vorhandene Schraubklemmen.

Ein Einschrumpfen mit Schrumpfschlauch ist auch möglich.

# 15. Kontakt

Anfragen: [wilhelm.wm.meier@gmail.com](mailto:wilhelm.wm.meier@gmail.com)