

RC MultiSwitch-D

... und RC MultiAdapter-DA, RC ServoSwitch-D sowie TipTip

Wilhelm Meier

Version 1.07, 24.02.2021: HW_1.2

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. Grundsätzliches | 5 |
| 6.2.1. Bedienung | 5 |
| 6.2.2. Konfiguration | 6 |
| 7. Software für OpenTX | 8 |
| 7.1. Installation der Software auf dem Sender | 9 |
| 7.1.1. Unterscheidung der verschiedenen Sendermodelle | 9 |
| 7.2. Konfiguration des Modells (Sender mit 480x272-Pixel <i>Farbdisplay</i>) | 9 |
| 7.2.1. Widget für die Bedienung | 9 |
| 7.2.2. Widget für die Konfiguration | 9 |
| 7.2.3. Mixer-Script | 10 |
| 7.2.4. Definition eines Übertragungskanals für das Schaltmodul am Empfänger | 10 |
| 7.2.5. Definition der Geber (Inputs) | 11 |
| 7.2.6. Flugphasen | 14 |
| 7.2.7. Spezielle Konfiguration für Sender mit Joystick (FrSky X12s) | 15 |
| 7.3. Konfiguration des Modells (Sender mit 128x64-Pixel oder 212x54-Pixel <i>monochrome Display</i>) | 17 |
| 7.3.1. Telemetrie-Seiten | 17 |
| 7.3.2. Mixer-Script und Übertragungskanal | 17 |
| 7.3.3. Geber und Flugphasen | 17 |
| 7.4. Konfigurationsdateien | 18 |
| 7.4.1. Die Datei <code>MODELS/swstd.lua</code> und weitere Dateien | 18 |
| 7.4.2. Das Widget <code>WMSW/main.lua</code> | 21 |
| 7.4.3. Das Widget <code>WMSWC/main.lua</code> (Konfiguration) | 22 |
| 7.5. Physische Schalter und die Schaltzustände | 23 |
| 7.5.1. ShortCuts / Abkürzungen | 23 |
| 7.5.2. Overlays / Ebenenumschaltung | 24 |
| 8. Einbau | 24 |
| 8.1. IBus oder SBus | 24 |
| 8.1.1. Ab Firmware V21 | 24 |
| 8.1.2. Bevor Firmware V21 | 24 |
| 8.2. Anschluß an den Akku | 25 |
| 8.3. Schutz | 25 |
| 9. Erste Einrichtung und Anlernen | 25 |
| 9.1. Kanal und Adresse | 25 |
| 9.2. Adressen beim RC MultiAdapter-DA | 26 |

| | |
|---|----|
| 10. Funktion | 27 |
| 10.1. Bedienung der Schaltfunktionen | 27 |
| 10.2. Konfiguration | 27 |
| 11. Betrieb | 28 |
| 12. Fehlersuche und Tests | 29 |
| 12.1. Checkliste IBus | 30 |
| 12.2. Checkliste SBus | 31 |
| 12.3. Gelegentliches Flackern oder Kanalfehlfunktion bei RC-Multi Adapter-DA..... | 32 |
| 13. Anerkennungen | 32 |
| 14. Kontakt | 32 |

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 Sachmängelhaftung.

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 Bausteinen 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 [Betrieb](#).

5. Spezieller Sicherheitshinweis: Kleinteile



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

6. Einführung

Der **RC MultiSwitch-D** ist ein digitales Schaltmodul (8-Kanal) zum Betrieb mit einem **OpenTX**-Sender, vorzugsweise eine **Radiomaster TX16s** oder anderen sog. *color-LCD* Sendern mit **OpenTX**. Im Gegensatz zu alten, im analogen Zeitmultiplex-Verfahren arbeitenden Schaltmodulen, wird bei dem **RC MultiSwitch-D** die Information für die zu schaltenden Funktion vollständig digital übertragen. Daher ist ein störungsfreier Betrieb gewährleistet.

Der **RC MultiSwitch-D** hat 8 Ausgänge, diese können ein- oder ausgeschaltet werden, entweder statisch, oder in zwei unterschiedlichen Blink-Modi. Zusätzlich kann der *Ein*-Zustand **PWM**-moduliert werden (auch beim Blinken ist die **PWM**-Modulation aktiv). Die Konfiguration der Zustände erfolgt komplett über den Sender und ein entsprechendes Menu.

Der **RC MultiSwitch-D** wird an den **IBus**- oder **SBus**-Ausgang eines Empfängers angeschlossen. Sollen mehrere **RC MultiSwitch-D** verwendet werden, so werden diese *alle* (parallel) an den IBus/SBus-Ausgang des Empfängers angeschlossen. Jeder **RC MultiSwitch-D** hat eine eindeutige *Adresse* und kann so vom Sender angesprochen 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-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.

Weiterhin kann die Software für **OpenTX** auch direkt Schaltmodule im sog. *TipTip* Verfahren bedienen. Hierzu ist im Moment kein Adapter **RC MultiAdapter-DA** notwendig. Stattdessen müssen die *TipTip*-Module wie üblich jeweils an einem eigenen Ausgang des Empfängers angeschlossen werden. Durch die Zusatzfunktionen ist das Bedienen (inkl. Beschriftung der Funktionen und Zustände) wie auch bei allen anderen Varianten (**RC MultiSwitch-D** und **RC MultiAdapter-DA**) komfortabel über das Menu im Sender möglich.

6.2. Grundsätzliches

6.2.1. Bedienung

Die Bedienung der Schaltmodule erfolgt grundsätzlich über ein *Menu*. Dieses ist als *Widget* in **OpenTx** realisiert.

In [Die Bedienoberfläche als Widget \(Seite 1\)](#) sieht man eine Seite (oben rechts: **Page: 1/4**) des Menus. Diese Seite ist für die Bedienung eines **RC MultiSwitch-D** konfiguriert: in der ersten Spalte stehen die *Namen* der Funktionen (hier: **Fun A** & **Fun H**) für die 8 Ausgänge des Schaltmoduls. Diese Namen können in der Konfigurationsdatei (s.a. [Die Datei MODELS/swstd.lua und weitere Dateien](#)) 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.

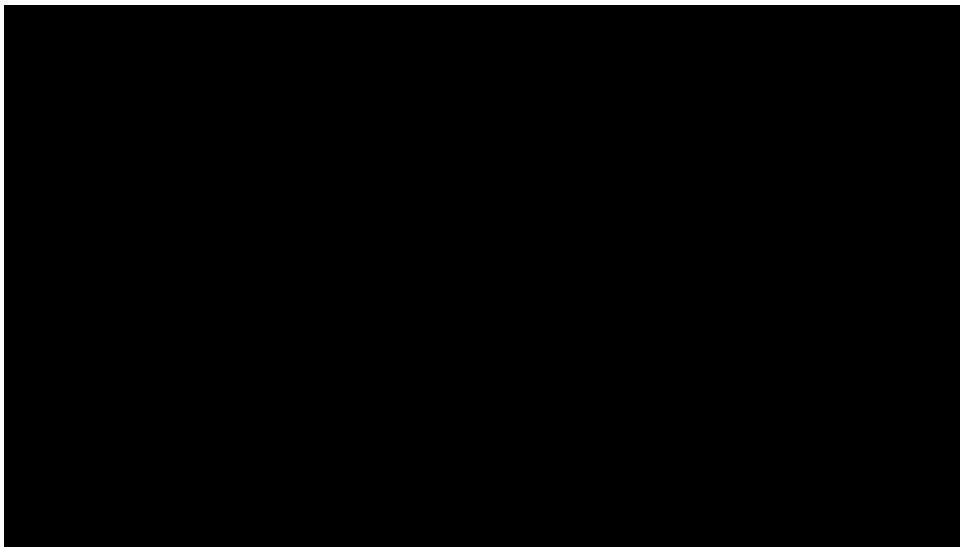


Abbildung 1. Die Bedienoberfläche als Widget (Seite 1)

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 **Fun A/sa**, dass der Funktion **Fun A** zusätzlich der *Shortcut* Schalter **sa** zugeordnet wurde. Dies geschieht auch in der Konfigurationsdatei. Mit einem 3-Positionenschalter sind natürlich nur die ersten *drei* Zustände **aus**, **ein** und **blink1** erreichbar.

Im Bild [Die Bedienoberfläche als Widget \(Seite 2\)](#) sieht man dann die nächste Seite mit den Funktionen des nächsten Schaltmoduls. Hier sind die Menu-Seiten entsprechend der Platzierung der Funktionen auf den Schaltmodulen strukturiert. Das muss aber so nicht sein: man kann auch Funktionen unterschiedlicher Module eher *thematisch* auf einer Menu-Seite zusammen fassen.

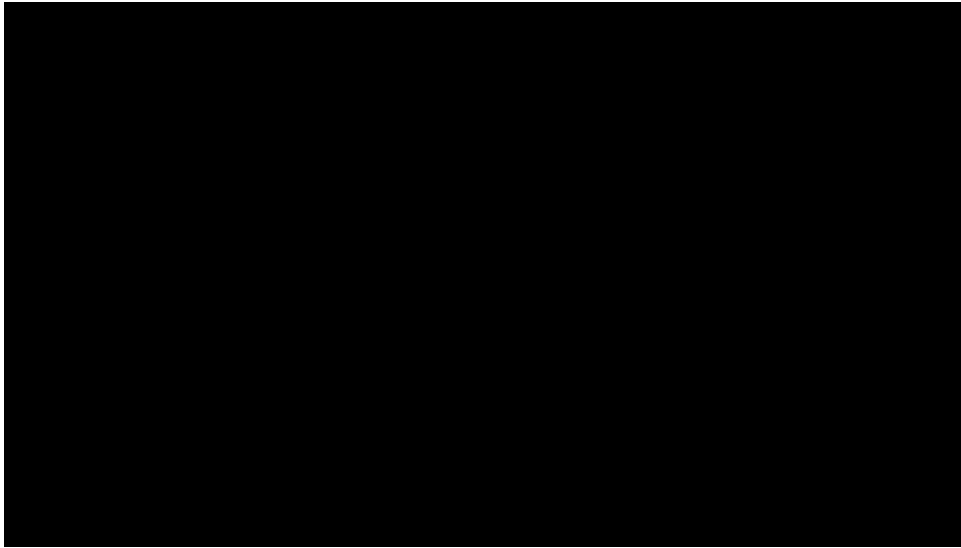


Abbildung 2. Die Bedienoberfläche als Widget (Seite 2)

Die *vierte* Seite dieses Beispiels stellt das Menu für ein **RC MultiAdapter-DA** dar. Hier hat jede Funktion nur *drei* Zustände. Die beiden **Ein**-Zustände entsprechen bei den üblichen analogen Zeitmultiplex-Schaltmodulen jeweils zwei unterschiedlichen Ausgängen. Natürlich können auch hier die Namen geändert werden.

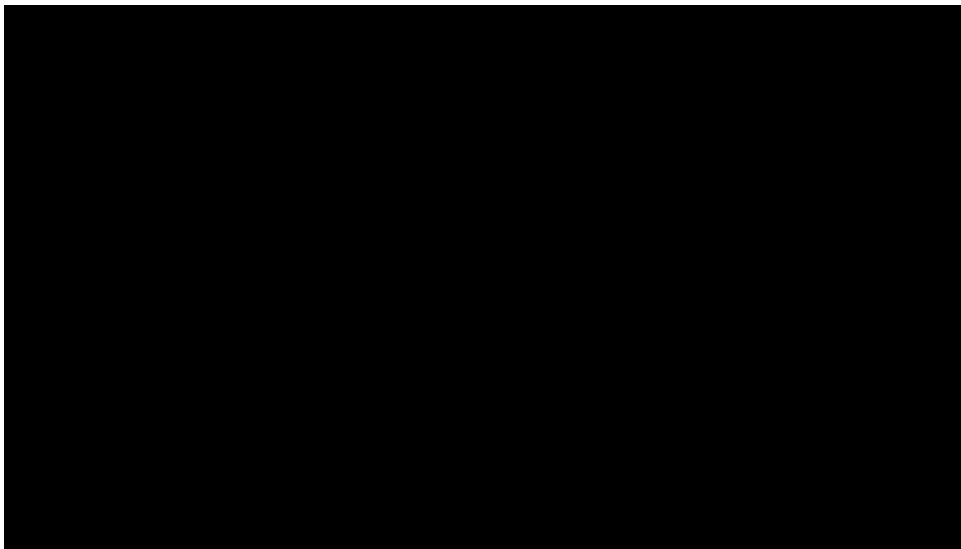


Abbildung 3. Die Bedienoberfläche als Widget (Seite 4)

Weil die Anzahl der phys. Schalter am Sender naturgemäß begrenzt ist (etwa: **sa**, **É 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**) ergibt sich eine sehr komfortable Bedienung.

6.2.2. Konfiguration

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

Im Bild [Die Konfigurationsoberfläche als Widget \(Funktionsspezifische Parameter\)](#) sind zu jeder Funktion die konfigurierbaren *Parameter* aufgelistet. Zudem besteht die Möglichkeit, ein *Reset* durchzuführen.

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.

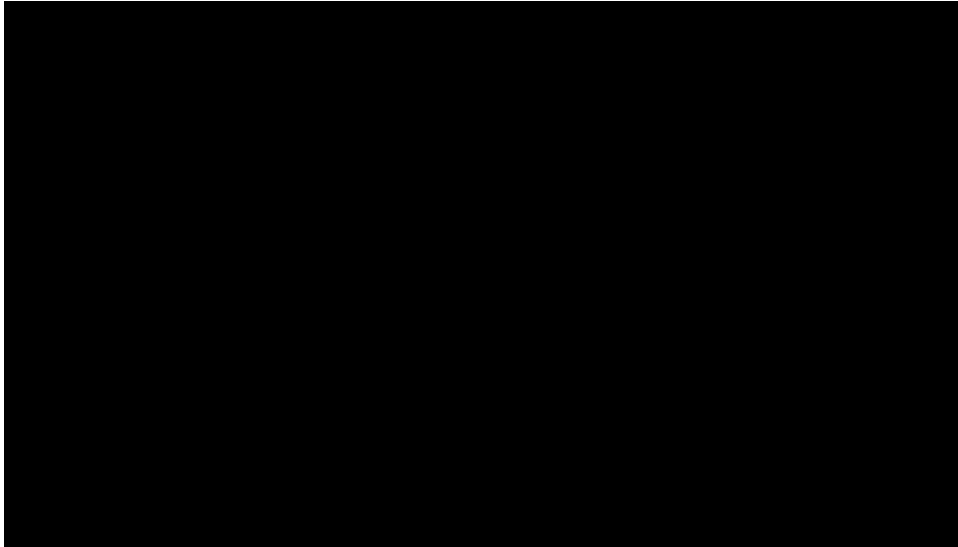


Abbildung 4. Die Konfigurationsoberfläche als Widget (Funktionsspezifische Parameter)

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. [Die Konfigurationsoberfläche als Widget \(modulglobale Parameter\)](#)).

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

Die Parameter **MPX0** ÷ **MPX4** beziehen sich *nur* auf das **RC MultiAdapter-DA**: hiermit kann die *Art* des am jeweiligen Ausgangs 0 ÷ 4 angeschlossenen analogen Zeitmultiplex-Schaltmodul festgelegt werden (Wert **1** := Graupner/JR, Wert **2** := Robbe, Wert **3** := CP).

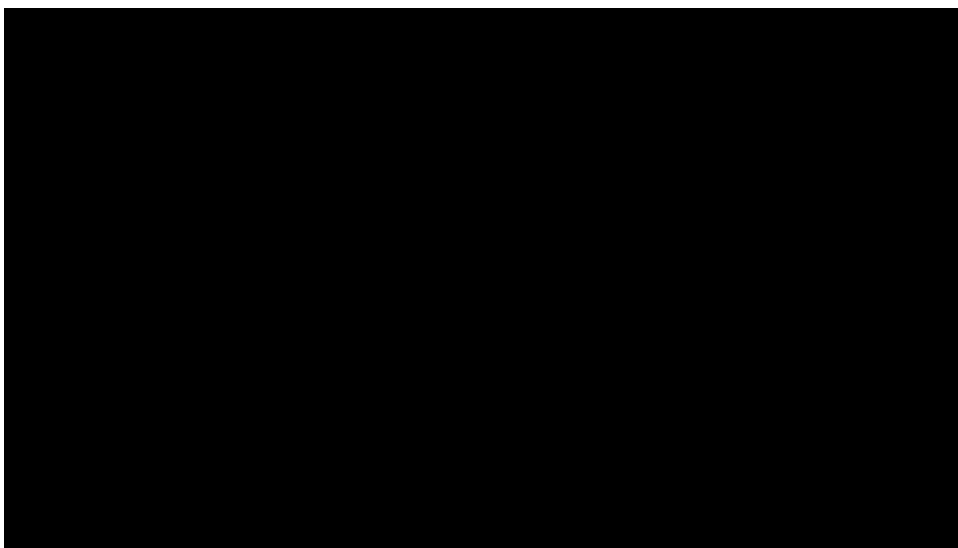


Abbildung 5. Die Konfigurationsoberfläche als Widget (modulglobale Parameter)

Da dieses Widget zum eigentlichen Betrieb *nicht* erforderlich ist, kann es *nach* der Konfiguration deaktiviert werden bzw. durch andere Widgets ersetzt werden.

7. Software für OpenTX

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

Lizenz der Software, des Übertragungsverfahrens und der Dokumentation

II



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 MultiSwitch-D** und **RC MultiAdapter-DA** sind die folgenden relevant:

¥ Widgets

```
" Widgets/WMSW/main.lua
" Widgets/WMSWC/main.lua
```

¥ Mixer-Skripte

```
" 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-Skripte (für monochrome Sender)

```
" SCRIPTS/TELEMETRY/wmsw.lua
" SCRIPTS/TELEMETRY/wmswc.lua
```

¥ 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*). 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`.

Kommen weitere Modelle hinzu, so wiederholen Sie diesen Schritt für jedes Modell.

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

Die 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 großes *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 hübsche) Konfiguration und die (hübschere) Bedienung abwechselnd auf *eine* Telemetrie-Seite legt.

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

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 MultiSwitch 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 MultiKanal Config x.y` dar.

7.2.3. Mixer-Script

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

¥ **enci b** für **IBus**-Empfänger

¥ **encsbm** für **SBus**-Empfänger und Sender mit MPM-Multiprotokoll-HF-Modul

¥ **encxj t** für **SBus**-Empfänger und Sender mit XJT-HF-Modul

7.2.4. Definition eines Übertragungskanals für das Schaltmodul 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. [Kanal und Adresse](#)).

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

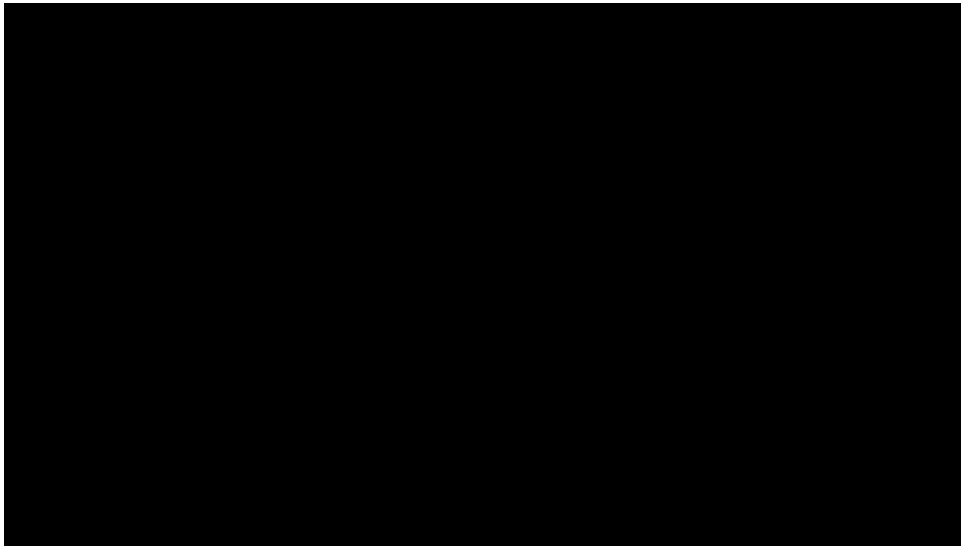


Abbildung 6. Das Mischer-Skript **encsbm**. lua

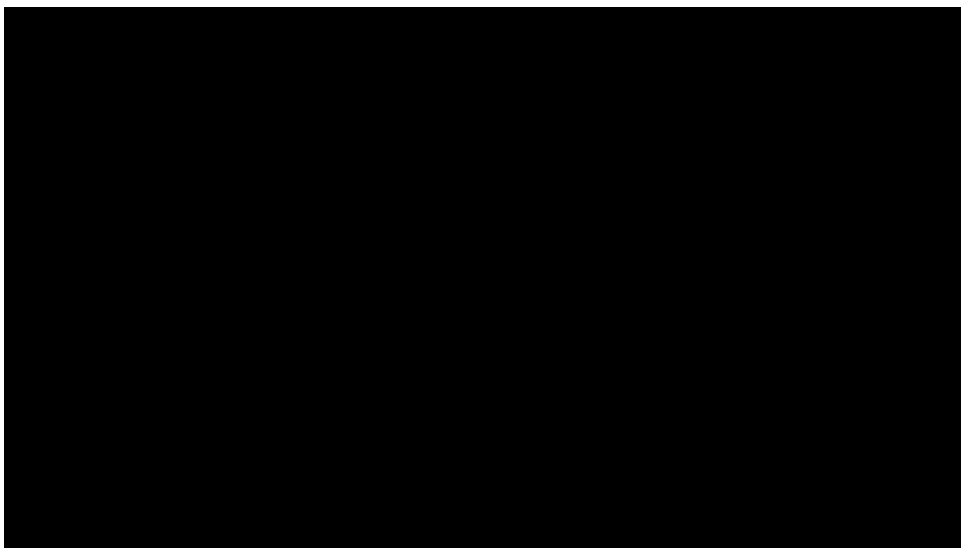


Abbildung 7. Hier wird der Übertragungskanal **10** wird für die Schaltmodule verwendet.



Abbildung 8. Als Quelle des Mischers des †bertragungskanal wird das Mischer-Skript *encsbm.lua* eingetragen

#

Mögliche Fehlerquelle im späteren Betrieb

Bei *jedem* Einschalten scannen die Schaltmodule *alle* †bertragungskanäle auf ein spezielles Konfigurationssignal, um den †bertragungskanal 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 †bertragungskanäle 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 †bertragungskanal 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 [Benötigte Geber zur Bedienung der Widgets \(Darstellung im Companion\)](#):

- ¥ I8 : Navigation von links nach rechts (und vom Zeilenende in den Zeilenanfang der nächsten Zeile)
- ¥ I9 : Navigation von rechts nach links (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 gleichmaßen notwendig.

Abbildung 9. Benötigte Geber zur Bedienung der Widgets (Darstellung im Companion)

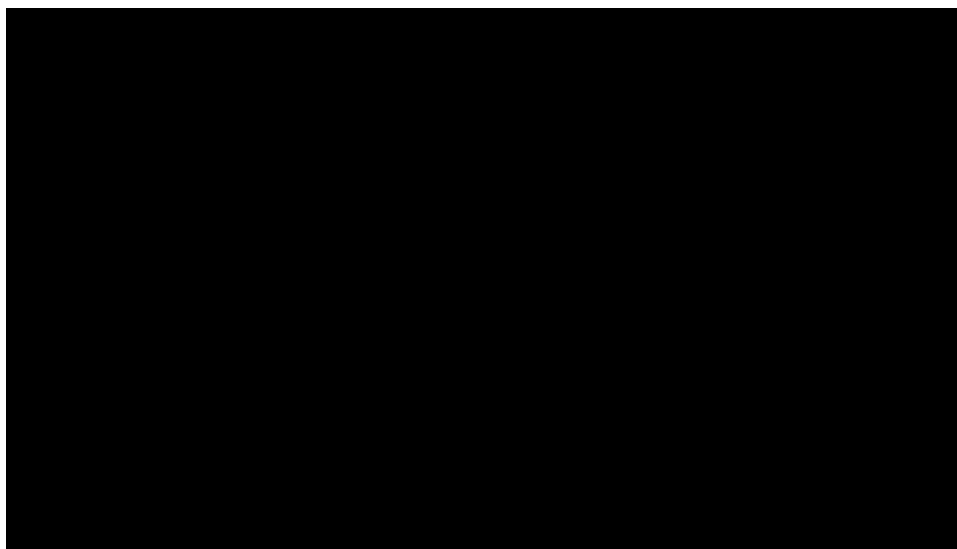


Abbildung 10. Benötigte Geber (Darstellung im Sender)

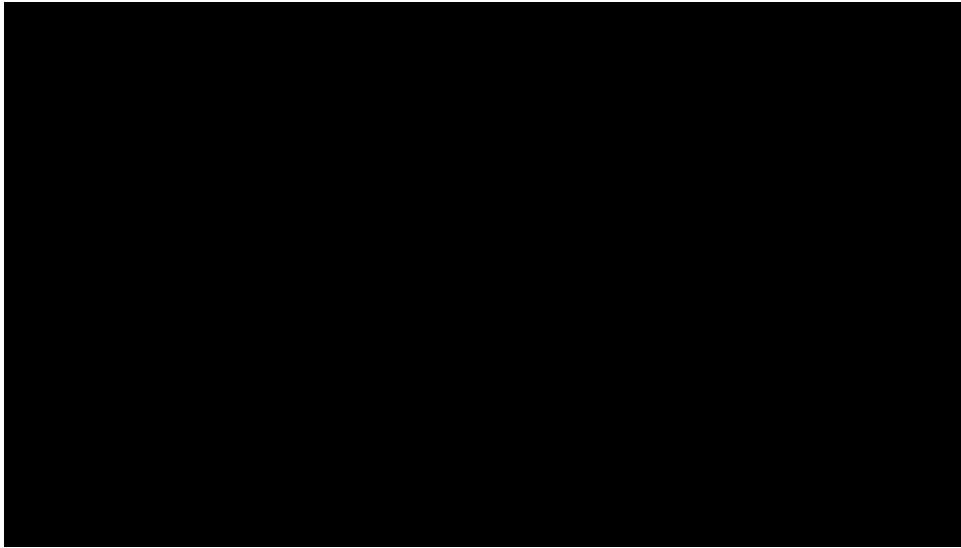


Abbildung 11. Benötigte Geber (Darstellung im Sender)

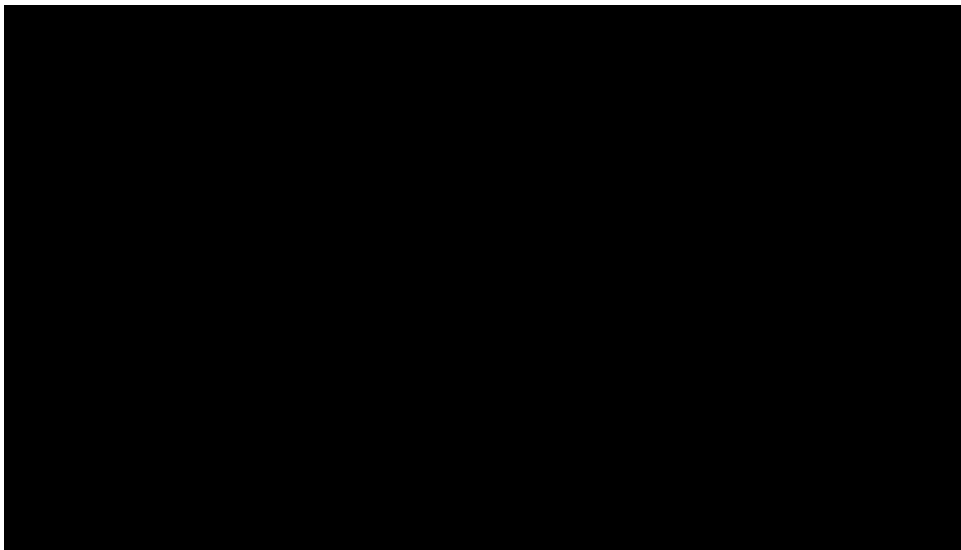


Abbildung 12. Benötigte Geber (Darstellung im Sender)

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 Datei `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 Auswahl von Menu-Punkten im Normalfall eingerichtet.

!

Menu-Schnellanwahl mit dem 6-Positionentaster `6pos` (Kreistaste)

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 MultiSwitch-D* kann/muss man einige Parameter einstellen wie etwa die Blinkfrequenz oder die Einschaltdauer für den Schaltzustand *blink1* oder *blink2*. 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. 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 13. Deaktivierung der Trimmer *T5* und *T6* in der Flugphase *0* (Darstellung im Companion)

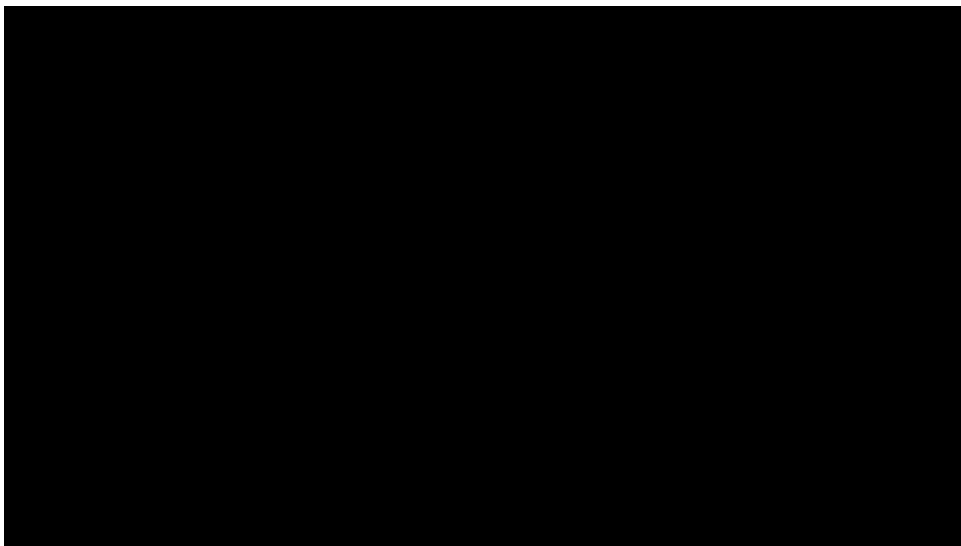


Abbildung 14. Deaktivierung der Trimmer *T5* und *T6* in der Flugphase *0* (Darstellung im Sender)

7.2.7. 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 [Kurve zur Umwandlung einer JoyStick-Bewegung in eine Navigation](#) 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). Getreu 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*.

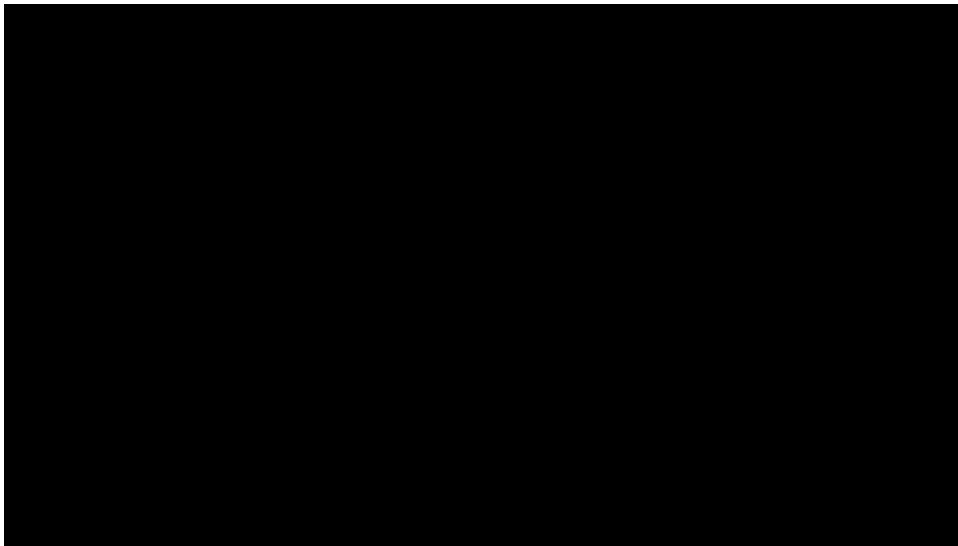


Abbildung 15. Kurve zur Umwandlung einer JoyStick-Bewegung in eine Navigation

In [Verwendung der Kurve aus Kurve zur Umwandlung einer JoyStick-Bewegung in eine Navigation für den Joystick in y-Richtung](#) sieht man die Verwendung der Kurve in *einem* Input.

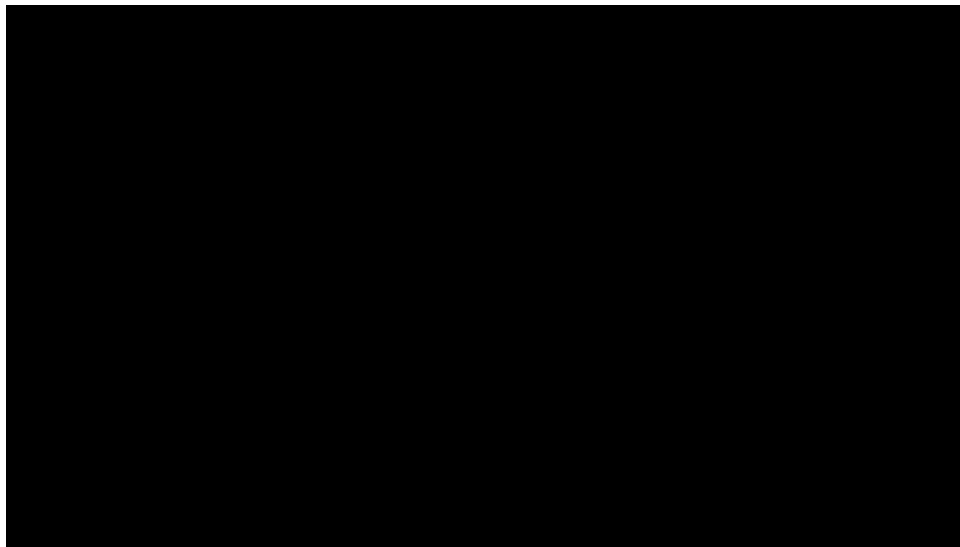


Abbildung 16. Verwendung der Kurve aus [Kurve zur Umwandlung einer JoyStick-Bewegung in eine Navigation](#) für den Joystick in y-Richtung

In [Konfiguration der alternativen Inputs 11É14 für die Bedienung](#) sieht man den Überblick über die gesamte Konfiguration aller vier Inputs. Man achte auf die *Spiegelung* der Kurve durch !.

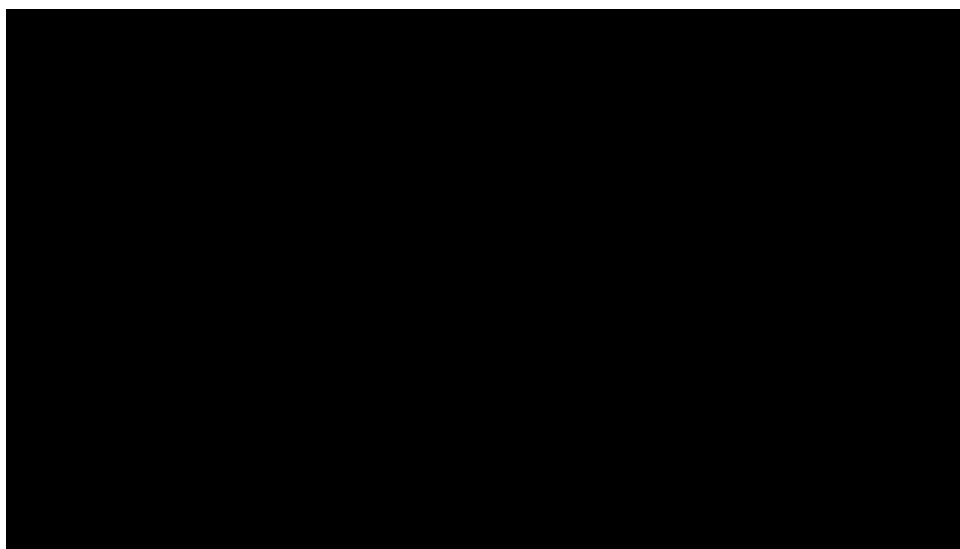


Abbildung 17. Konfiguration der alternativen Inputs 11É14 für die Bedienung

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

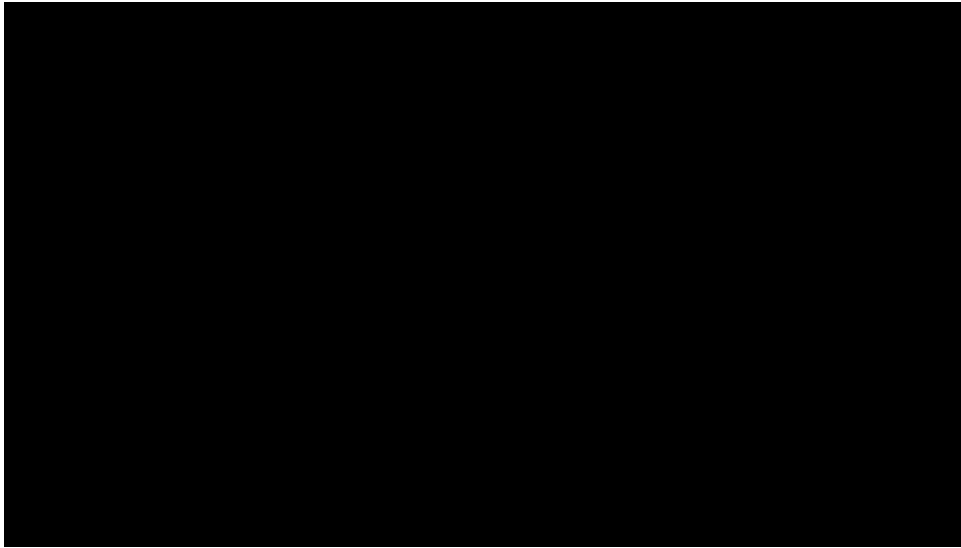


Abbildung 18. Konfiguration des Widget zur Verwendung der alternativen Inputs

Diese Konfiguration kann sowohl für das *Bedienungs*widget `wmsw` wie auch für das *Konfigurations*widget `wmswc` durchgeführt werden.

7.3. Konfiguration des Modells (Sender mit 128x64-Pixel oder 212x54-Pixel *monochrome Display*)

Die folgenden Einstellungen gelten nur für Sender mit 128x64-Pixel oder 212x54-Pixel *monochrome 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 [Mixer-Script](#) bzw. [Definition eines Übertragungskanals für das Schaltmodul am Empfänger](#).

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` der SD-Karte werden modellspezifische Menu-Konfigurationen abgelegt.

Dabei gilt folgende Suchreihenfolge für ein Modell 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 Modell `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. [Mixer-Script](#)).

Diese Beschränkungen haben zur Folge, dass 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).

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

```

É1 local name = "Default";
É2
É3 local gVariable = 5;
É4
É5 local useSbus = 1; -- only 4 states, only 16 parameter values !
É6
É7 local gstates1 = {"aus", "ein", "blink1", "blink2"}; !
É8 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 = { "
31                 {name = "M1A", states = gstates1, state = 1, data = {switch = "sa", count = 1,
32 module = 1}}, #
33                 {name = "M1B", states = gstates1, state = 1, data = {switch = "sb", count = 2,
34 module = 1}},
35                 {name = "M1C", states = gstates1, state = 1, data = {switch = nil, count = 3, module
36 = 1}},
37                 {name = "M1D", states = gstates1, state = 1, data = {switch = "se", count = 4,
38 module = 1}}, $
39                 {name = "M1E", states = gstates1, state = 1, data = {switch = nil, count = 5, module
40 = 1}},
41                 {name = "M1F", states = gstates1, state = 1, data = {switch = nil, count = 6, module
42 = 1}},
43                 {name = "M1G", states = gstates1, state = 1, data = {switch = nil, count = 7, module
44 = 1}},
45                 {name = "M1H", states = gstates1, state = 1, data = {switch = nil, count = 8, module
46 = 1}},
47             }
48         },
49         { -- template for digital multiswitch RC-MultiSwitch-D @ Address(2)
50             items = { %
51                 {name = "M2A", states = gstates1, state = 1, data = {switch = "sc", count = 1,
52 module = 2}},
53                 {name = "M2B", states = gstates1, state = 1, data = {switch = "sd", count = 2,
54 module = 2}},
55                 {name = "M2C", states = gstates1, state = 1, data = {switch = nil, count = 3, module
56 = 2}},
57                 {name = "M2D", states = gstates1, state = 1, data = {switch = "se", count = 4,
58 module = 2}}, &
59                 {name = "M2E", states = gstates1, state = 1, data = {switch = nil, count = 5, module
60 = 2}},
61                 {name = "M2F", states = gstates1, state = 1, data = {switch = nil, count = 6, module
62 = 2}},
63                 {name = "M2G", states = gstates1, state = 1, data = {switch = nil, count = 7, module
64 = 2}},

```

```

50      {name = "M2H", states = gstates1, state = 1, data = {swi tch = nil, count = 8, module
51      E = 2}},
52      },
53      ...
54 };
55 ...

```

! Diese Namen für die Schaltzustände können frei gewählt werden.

" 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).

Den Namen der Funktion (`name = "M1A"`) können Sie beliebig ändern.

% Dies ist die Konfiguration für das Module mit der Adresse 1 (s.a. `module = 2`)

\$ Diese Funktion hat einen Schalter zugeordnet (`swi tch = "se"`).

& Diese Funktion hat einen Schalter zugeordnet (`swi tch = "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)

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 Menüs 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 Menü des Moduls **1** auf dessen Funktion **Fun3** wirkt, bei aktivem Menü 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, dass 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 MultiSwitch-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 MultiSwitch-D** befinden. Für jeden Ausgang des **RC MultiSwitch-D** können dann die unterschiedlichen Zustände ausgewählt werden.

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

Um auf die nächste Seite (für das nächste Schaltmodul **RC MultiSwitch-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 MultiSwitch-D**) umgeschaltet.

Alternativ kann mit dem 6-Positionentaster **6pos** umgeschaltet werden.

Zusätzlich stehen **LS** zu 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 \dagger bertragungskanal 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:

- ¥ **RES**: Reset aller Werte der Funktion (Wert *muss* auf 1 stehen, damit de Reset ausgeführt wird)
- ¥ **PWN**: Einstellen der PWM-Modulation, kann direkt an der *Helligkeit* der Kontroll-LED beobachtet werden. Beim RC-ServoSwitch-D wird hier die *Geschwindigkeit* eingestellt (1=langsam, 31=schnell).
- ¥ **B1/I**: Intervall des Blinkens für Schaltzustand *blink1* bzw. *Position1* (aus) beim RC-ServoSwitch-D
- ¥ **B1/D**: Einschaltdauer des Blinkens für Schaltzustand *blink1* bzw. *Position2* beim RC-ServoSwitch-D
- ¥ **B2/I**: Intervall des Blinkens für Schaltzustand *blink2* bzw. *Position3* beim RC-ServoSwitch-D
- ¥ **B2/D**: Einschaltdauer des Blinkens für Schaltzustand *blink2* bzw. *Position4* beim RC-ServoSwitch-D
- ¥ **PThu**: Der sog. PassThrough-Kanal: ein beliebiger OpenTX-Kanal (1 ÷ 16) kann zur Weiterleitung an den Ausgang des Schaltmoduls ausgewählt werden. Beim RC-ServoSwitch kann bei einem Wert von 1 mit Hilfe des Kanals 1 des Senders (fest zugeordnet) die *Position1* gelernt werden, bei einem Wert 2 die *Position2* u.s.w. Bei einem Wert von 11 wird der Follow-Mode als *Copy Positions*, bei einem Wert von 12 wird der Follow-Mode als *Own-Positions* eingestellt, der Wert von 10 stellt wieder den normalen Bewegungsmodus ein.
- ¥ **Min**: (unbenutzt / nicht vorhanden in neueren Versionen)
- ¥ **Max**: (unbenutzt / nicht vorhanden in neueren Versionen)

Es existiert eine zusätzliche Seite (am Ende) mit modul-globalen Einstellungen:

- ¥ **Learn Ch/Adr**: *Anlernen* der Moduladresse und des \dagger bertragungskanals (s.a. [Kanal und Adresse](#))
- ¥ **TMpx**: Zeitmultiplex-Verfahren für den Ausgang mit der jeweilige Adresse (nur bei RC-MultiAdapter-DA, s.u.)
- ¥ **TMode**:
 - a. Anpassung der Impulslänge für ein/aus im Zeitmultiplex-Verfahren für den Ausgang mit der jeweilige Adresse (nur bei RC-MultiAdapter-DA, s.a. [RC-MultiAdapter-DA: Anpassungen an Toleranzen der analogen Schaltmodule](#))
 - b. Testmuster (nur bei RC-MultiSwitch-D)
- ¥ **OMpx**: Länge des Synchron-Impulses für analoge Schaltmodul (nur bei RC-MultiAdapter-DA), damit lassen sich Toleranzen der analogen Schaltmodule ausgleichen (s.a. [RC-MultiAdapter-DA: Anpassungen an Toleranzen der analogen Schaltmodule](#)).
- ¥ **Test**: Verschiedene Test-Modi (nur zur Funktionsüberprüfung des Moduls).

Bei der Einstellung des Zeitmultiplexverfahrens gelten folgende Zuordnungen von eingestelltem Wert und Verfahren:

- ¥ Wert 0 # Graupner-8K

- ¥ Wert 1 # Graupner-4K
- ¥ Wert 2 # Robbe
- ¥ Wert 3 # CP-Elektronik 8-Ausgänge
- ¥ Wert 4 # CP-Elektronik 16-Ausgänge
- ¥ Wert 5 # unbenutzt
- ¥ alle anderen Werte # Graupner-8K

!

Einrichten von sog. Multi-Prop-Modulen

Zum Betrieb von sog. Multi-Prop-Modulen ist es nötig, für die *gewünschten* Kanäle Zuordnungen zu den **OpenTx**-Kanälen zu machen. Dies macht man mit dem **PassThru**-Parameter (s.o.).

Konfiguriert man *keinen* **PassThru**-Kanal, so bewegt sich das dort angeschlossene Servo von *Endstellung_1* über *Neutral* zu *Endstellung_2*.

!

RC_MultiAdapter-DA: Anpassungen an Toleranzen der analogen Schaltmodule

Es kommt vor, dass die analogen Schaltmodule mit einem sporadischen *Flackern* der Ausgänge reagieren. Um dies zu kompensieren, dient der Parameter **Ompx**. Neutrale Werte sind bei **Ibus** etwa 16, bei **SBus** etwa 8. Es kann sein, dass dieser Wert beim Auftreten des Flackern vergrößert werden muss.

Seltener kommt es vor, dass zwar *kein* Flackern auftritt, jedoch die *ein*-Positionen nicht immer erkannt werden. *nur* in diesem Fall den Parameter **TMode** verändern.

!

RC_MultiAdapter-DA: Anpassungen an CP-Elektronik Schaltmodule

Kanalzählung von 8 zu 1 (anders herum) bzw. 16 zu 1

LED des CP-Moduls muss *erlöschen* # gültiges Signal

Falls kein gültiges Signal erkannt wird:

¥ **Ompx** \$ 8`

¥ **TMode** \$ 8

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. IBus oder SBus

8.1.1. Ab Firmware V21

Ab dieser *Firmware*-Version werden die Module auf eine *automatische* Erkennung des Bus-Systems (*IBus*, *SBus*, *SBus* (invertiert)) umgestellt. Auch ein *Inverter* bei *Sbus* (s.a. *Einfacher Inverter mit einem Kleinsignal-MosFet*) *entfällt*.

| Version | Modul mit <i>Bus</i> -Erkennung |
|---------|---------------------------------|
| > V21 | RC-MultiAdapter-DA |

8.1.2. Bevor Firmware V21

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 MultiSwitch-D* oder *RC MultiAdapter-DA* eingeschleift werden (wie ein Verlängerungskabel).

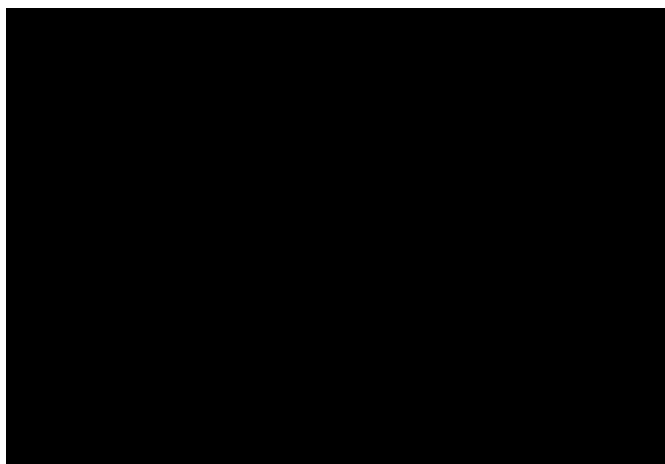


Abbildung 19. Einfacher Inverter mit einem Kleinsignal-MosFet

Ein Inverter wie in *Einfacher Inverter mit einem Kleinsignal-MosFet* 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:

¥ FrSky FS-i A4B in der Einstellung SBus, PWM

¥ FrSky FS-i A6B in der Einstellung SBus, PWM

¥ FrSky FS-i A10B in der Einstellung SBus, PWM

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

8.2. Anschluß an den Akku

Achten Sie auf richtige Polung.

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 den Pfostenverbinder für den Kontakt, den die Löt-Pads für die Anschlußkabel ab.

Ein Einschrumpfen mit Schrumpfschlauch ist möglich.

9. Erste Einrichtung und Anlernen

Im Auslieferungszustand sind die RC MultiSwitch-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. Kanal und Adresse

Das Anlernen von Kanal und Adresse geschieht in einem Schritt. Dazu *muss* das zu konfigurierende RC MultiSwitch-D *allein* am betreffenden I Bus oder SBus hängen. Weiterhin sollten außer dem zu konfigurierenden Übertragungskanal für die RC MultiSwitch-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).



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.

Ablauf:

1. Sender und Empfänger sowie Schaltmodule aus.
2. Ein RC Mul ti Swi tch-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
8. das zu konfigurierende RC Mul ti Swi tch-D einschalten, sofern nicht über BEC versorgt (neue Version) (RC-Mul ti Adapter-DA wird vom Empfänger versorgt).
 - a. RC-Mul ti Adapter-DA und RC-ServoSwi tch-D: die Kontroll-LED leuchtet für 3 Sekunden, anschließend blinkt sie `1 ÷ 8` mal, je nach eingestellter Adresse. Sollte dies nicht passieren, so wurde der Übertragungskanal nicht gefunden.
 - b. RC-Mul ti Swi tch-D: entsprechend der gelernten Adresse leuchtet ein Kanal auf.
9. nach ca. 2 weiteren Sekunden das RC Mul ti Swi tch-D ausschalten
10. Für weitere RC Mul ti Swi tch-D-Module die n. Adresse selektieren
11. das n. RC Mul ti Swi tch-D anschließen und einschalten
12. ÷

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. Andernfalls ist der Anlernvorgang zu wiederholen.

9.2. Adressen beim RC Mul ti Adapter-DA

Weil der RC Mul ti Adapter-DA insgesamt 5 alte Schaltmodule ansteuern kann, belegt er auch 5 *unmittelbar aufeinanderfolgende* Adressen.

Damit gilt das folgende Schema für die Zuordnung von Adressen zu den Ausgängen des RC Mul ti Adapter-DA und den daran angeschlossenen Schaltmodulen:

- ⌘ erlernte Adresse `A` # Ausgang `S2`
- ⌘ Adresse `(A+1)` # Ausgang `S1`
- ⌘ Adresse `(A+2)` # Ausgang `Q0`

¥ Adresse (A+3) # Ausgang S3

¥ Adresse (A+4) # Ausgang S4

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 É 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 zugeordnet, 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.

Deletion 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. Betrieb

Beachten Sie unbedingt die Anweisungen unter [\[first\]](#).

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

Beachten Sie alle folgenden Hinweise zum Betrieb.

Eine Verwendung des Moduls in Rennbooten oder Flogmodellen 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.

12. Fehlersuche und Tests

Weder der RC-MultiSwitch-D noch der RC-MultiAdapter-DA können dem Benutzer direkt Rückmeldung über ihre Einstellungen geben. Daher ist die Fehlersuche auf ein klar strukturiertes Vorgehen angewiesen.

Die häufigsten Fehlerquellen sind:

⌘ Unterscheidung I_{Bus} / S_{Bus}

" Unterscheidung S_{Bus} mit XJT-Sendemodul oder MPM-Sendemodul.

⌘ S_{Bus}-Inverter

⌘ Addressvergabe

12.1. Checkliste I_{Bus}

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

Richtiger I_{Bus}-Ausgang?

An den Empfängern gibt es zwei I_{Bus}-Ausgänge: I_{Bus}-Servo und I_{Bus}-Sensor. Es muss der I_{Bus}-Servo Ausgang gewählt werden.

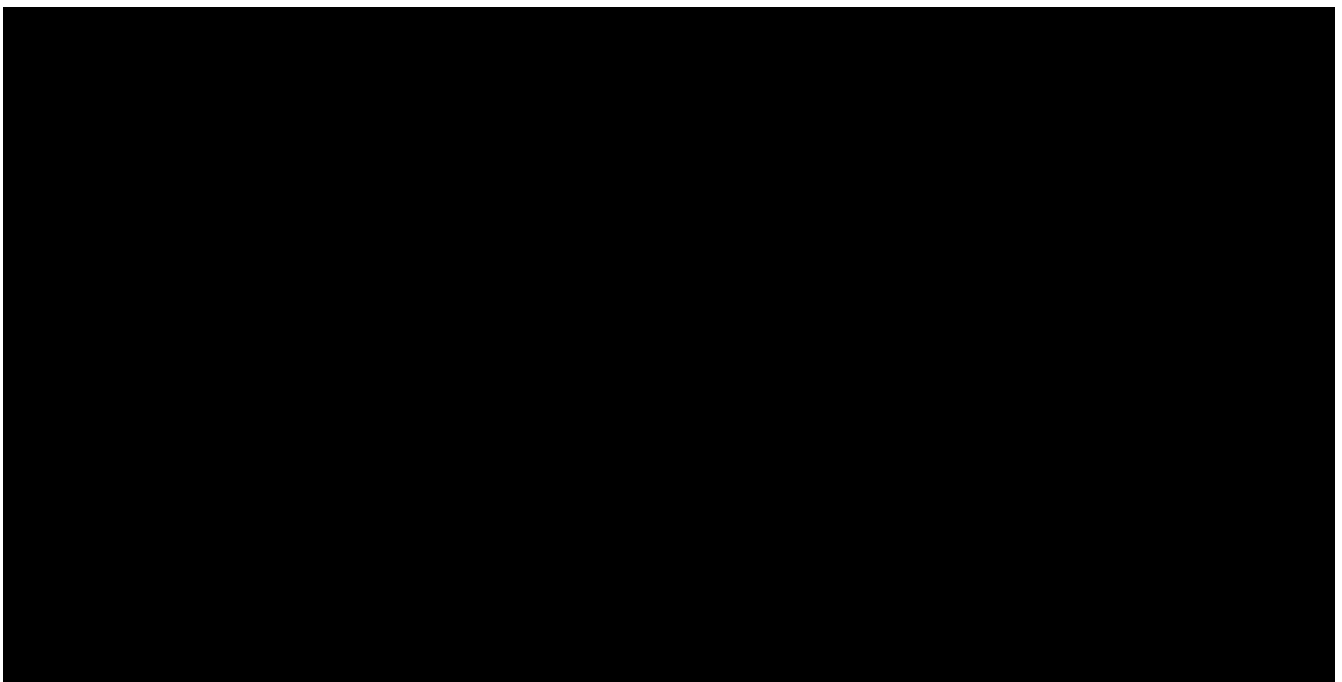


Abbildung 20. Aufbau mit einem Empfänger, der I_{Bus}-Servo liefert

Korrekte Verkabelung?

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

Beim RC-MultiAdapter-DA existieren keine Kontroll-LEDs. Man also zur Funktionskontrolle ein analoges Schaltmodul anschließen. Und auch hier an dessen Ausgänge z.B. Leuchtdioden zur Kontrolle anschließen sowie eine Spannungsversorgung.

Richtige Konfiguration Multi-Protokol-HF-Modul?

Das MPM im Sender muss auf MULTI FlySky_2A PWM, I_{Bus} eingestellt sein.

Richtige Konfiguration des Modells?

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

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

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

! 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 `enci b` eingerichtet sein.

12.2. 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.

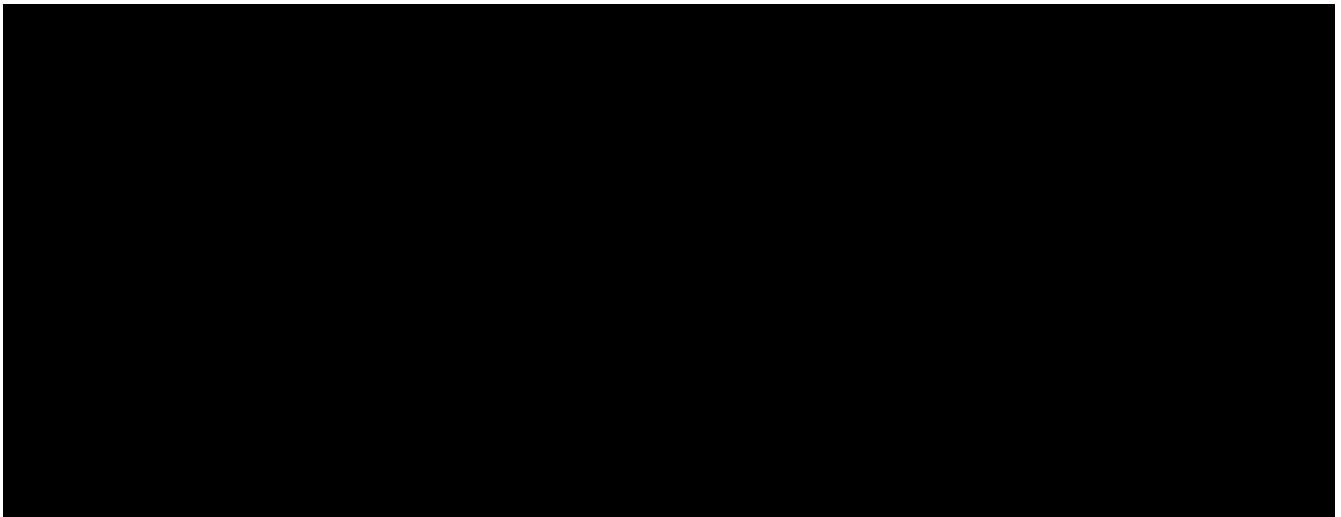


Abbildung 21. Aufbau mit einem Empfänger, der SBus-Servo liefert

Korrekte Verkabelung?

Für RC-MultiSwitch-D und RC-MultiAdapter-DA benötigt man einen SBus-Inverter. Andere SBus-Teilnehmer sind nach wie vor vor dem Inverter anzuschließen.

Beim RC-MultiAdapter-DA existieren keine *Kontroll-LEDs*. Man also zur Funktionskontrolle ein analoge Schaltmodul anschließen. Und auch hier an dessen Ausgänge z.B. Leuchtdioden zur Kontrolle anschließen sowie eine Spannungsversorgung.

Richtige Konfiguration des Modells?

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

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 !
6
7 local gstates1 = {"aus", "ein", "blink1", "blink2"};
8 local gstates2 = {"aus", "ein 1", "ein 2"}; ----
```

! 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.

12.3. Gelegentliches Flackern oder Kanalfehlfunktion bei RC-MultiAdapter-DA

Manchmal kommt es zu Toleranzen bei den alten, analogen Schaltmodulen. Bitte nach [RC-MultiAdapter-DA: Anpassungen an Toleranzen der analogen Schaltmodule](#) verfahren.

13. Anerkennungen

Die folgenden Personen haben durch ihr unermüdliches Testen, Bereitstellung von Sender und Empfängern, und ihr konstruktives Feedback wesentlich zum Erfolg des Projektes beigetragen:

¥ Stefan Becker

¥ Jürgen Jahn

¥ Axel Müller

¥ Johannes Unrath

14. Kontakt

Anfragen: wilhelm.wm.meier@googlemail.com