

# Bus-Systeme im RC-Modellbau

## *Eine Übersicht über digitale Bus-Systeme*

Wilhelm Meier

Version 0.4, 15.02.2021: -

# Inhalt

1. Vorwort .....	1
1.1. Lizenz .....	1
2. Überblick .....	2
3. Was ist ein sog. BUS? .....	2
4. Die digitalen Varianten .....	4
4.1. Technische Realisierung .....	4
4.2. Graupner / SJ .....	4
4.2.1. Aktoren: SumD .....	4
4.2.2. Sensoren: Hott-Sensor .....	5
4.2.3. Besonderheiten .....	5
4.3. FrSky bzw. Futaba .....	5
4.3.1. Aktoren: SBus .....	5
4.3.2. Sensoren: S.Port .....	5
4.3.3. Besonderheiten .....	6
4.4. FlySky .....	6
4.4.1. Aktoren: IBus .....	6
4.4.2. Sensoren: IBus .....	6
4.5. Spektrum: .....	6
4.5.1. Aktoren: XBus .....	6
4.5.2. Sensoren: XBus .....	6
5. Erweiterungen .....	7
5.1. <b>S</b> Bus-PWM-Adapter .....	7
5.2. <b>I</b> Bus-PWM-Adapter .....	7
6. Analoges .....	8
6.1. PWM-Servosignal .....	8
6.2. CPPM .....	8
6.3. Doppeltes Zeitmultiplex .....	8
6.3.1. Varianten .....	9
7. Die Protokolle im Detail .....	10
7.1. SumD .....	10
7.2. Hott-Sensor .....	10
7.3. SBus .....	10
7.4. S.Port .....	10
7.5. IBus-Servo .....	10
7.6. IBus-Sensor .....	10
8. Produkte .....	10
8.1. SumD / Hott-Sensorprotokoll .....	10
8.2. SBus / S.Port .....	10
8.3. IBus .....	10
9. Kontakt .....	10

# 1. Vorwort

## 1.1. Lizenz

Dieses Dokument wird unter der folgenden *Lizenz* veröffentlicht:



### 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.

## 2. Überblick

Dieses Dokument soll einen Überblick über die unterschiedlichen *Bus*-Systeme geben, die derzeit im RC-Modellbau eingesetzt werden, geben. Neben den *digitalen* Bus-Systemen wird auch noch auf *analoge* Verfahren eingegangen, um beides voneinander abzugrenzen.

## 3. Was ist ein sog. BUS?

Der Begriff **Daten-Bus** oder einfach nur **Bus** kommt aus der technischen Informatik, und er beschreibt eine *Verbindungseinrichtung* von mehreren Komponenten eines digitalen Rechensystems.

Im einfachsten Fall kann man sich einen *Bus* als eine Menge von parallelen elektrischen Leitungen vorstellen, mit denen Geräte miteinander verbunden werden, um miteinander zu kommunizieren bzw. elektrische Signale zu verteilen. Aus dem täglichen Leben kennt man solche Systeme wie etwa den **USB** (*universal serial bus*) am eigenen PC oder Notebook. Die elektrische Stromversorgung im Haus kann man auch als (analogen) Bus bezeichnen (hier wird allerdings meistens *nur* die elektrische Energie verteilt).

Die *einfachste* Form eines Bus-Systems ist der sog. *unidirektionale Master/Slave-Bus*: hier hat man **einen Master** und *mehrere Slaves*. Als *Master* bezeichnet man das Gerät, das die Kommunikation einleitet und Nachrichten versendet. Als *Slave* bezeichnet man die Geräte, die in diesem Fall nur Nachrichten empfangen. *Unidirektional* bedeutet, dass immer nur der Master sendet, die Slaves immer nur passiv mithören.

Ein andere Form eines Bus-Systems ist der sog. *bidirektionale Master/Slave-Bus*: der Unterschied zur unidirektionalen Variante ist, dass hier auch die Slaves etwas an den Master *zurücksenden* können. Dies tun sie aber eben nur nach einer *Aufforderung* durch den *Master*.

Andere Formen von Bus-Systemen sind derzeit im RC-Modellbau nicht üblich.



### *Unidirektionaler Master-Slave-Bus im RC-Modellbau (Aktor-Bus)*

Vereinfacht: der RC-Empfänger sendet eine Nachricht, alle angeschlossenen Servos oder ESC (*electronic speed controller*) empfangen diese Nachricht und suchen sich das Passende heraus.



### *Bidirektionaler Master-Slave-Bus im RC-Modellbau (Sensor-Bus)*

Vereinfacht: der RC-Empfänger sendet eine spezielle Nachricht als Aufforderung zum Antworten, alle angeschlossenen Servos oder ESC (*electronic speed controller*) antworten, wenn sie angesprochen wurden.

Aus dem oben gesagten folgt, dass wir es *meistens* mit *zwei* Bus-Systemen im RC-Modellbau zu tun haben: einen *unidirektionalen* für die **Aktoren** (**Aktor-Bus**) und einen *bidirektionalen* für die **Sensoren** (**Sensor-Bus**). *Aktoren* können bspw. Servos sein, *Sensoren* können bspw. Spannungen oder Ströme messen.



#### *Was ist mit dem Sender?*

Der Sender spielt bei der Betrachtung *gar keine* Rolle. Voraussetzung ist natürlich, dass er mit dem jeweiligen Empfänger *gebunden* werden kann. Ob das möglich ist, ist eine Frage des verwendeten, digitalen *HF-Protokolls* zwischen Sender und Empfänger (z.B. Hott, oder AFHDS2A, ACCST, ...). Und das wiederum könnte der Inhalt eines weiteren Dokumentes werden, gerade wenn man an *Multiprotokoll-HF-Module* denkt.

## 4. Die digitalen Varianten

Mit Einzug der 2,4-GHz-Funktechnik in den RC-Modellbau haben die meisten Hersteller auch Bus-Systeme eingeführt. Wie oft zu beobachten entstehen nach dem Einführen erst einmal *inkompatible* Herstellerlösungen. So auch hier im RC-Modellbau sind die unten beschriebenen Bus-Systeme inkompatibel zueinander. Ob es in der Zukunft zu einer Konsolidierung etwa durch einen herstellerübergreifenden Standard (wie etwa **USB** in der Computertechnik) ist derzeit recht ungewiss.



### *Anzahl der Kanäle bei den Aktor-Bus-Systemen*

Die Aktor-Bus-Systeme wie etwa **SBus** übertragen meistens *mehr* Kanäle (bei **SBus**: 16) als der Empfänger Servo-Anschlüsse hat (s.a. **Erweiterungen**).

### 4.1. Technische Realisierung

Alle im folgenden beschriebenen Bus-Systeme arbeiten mit Hilfe einer **seriellen** Datenübertragung: bei den unidirektionalen Bus-Systemen ist es eine *no-duplex, asynchrone* serielle Übertragung, bei den bidirektionalen ist es eine *half-duplex, asynchrone* serielle Übertragung der einzelnen Bits. Formate und **Baud-Rate** (Geschwindigkeit) unterscheiden sich von Hersteller zu Hersteller wie auch die darauf aufbauenden **Protokolle**.



### *Servos oder andere Aktoren am Aktor-Bus*

Stand heute ist es noch recht selten, einzelne *Aktoren* am Aktor-Bus direkt anzuschließen. Es gibt zwar bspw. **SBus**-Servos, doch sind diese noch recht teuer. Außerdem müsste man hierbei jedes Servo immer extra konfigurieren, weil das Servo ja wissen muss, auf welchen der vielen übertragenen Kanäle es reagieren soll. Dies ist den meisten RC-Modellbauern zu umständlich. Derartige Servos kommen überlicherweise in der Robotik zum Einsatz. Für den RC-Modellbauer ist es einfacher, das Servo an den passenden Empfängeranschluss einzustecken.



### *Digitale Servos*

Als *digitale* Servos werden heute solche bezeichnet, die intern digital arbeiten. Das hat aber nicht mit der Schnittstelle / Verbindung zum RC-Empfänger zu tun: die ist nach wie vor analog (s.a. **PWM-Servosignal**). Rein technisch ist dies zwar nicht sinnvoll (s.a. **Servos oder andere Aktoren am Aktor-Bus**), aber man hat eine Schnittstelle, die sowohl von intern analogen wie auch von intern digitalen Servos verwendet werden kann.

### 4.2. Graupner / SJ

Die Empfänger haben einen Anschluß für das Aktor-Protokoll **SumD** und einen anderen für das Sensor-Protokoll **Hott-Sensor**. Bei manchen Empfängern mit *begrenzter* Anzahl von Anschlüssen muss man im Konfigurationsmenu des Empfängers ggf. die Bedeutung eines der Anschlüsse auf **SumD** oder **Hott-Sensor** umschalten (z.B. beim GR-12L).

#### 4.2.1. Aktoren: SumD

Es existieren verschiedene Versionen bezeichnet als **SUMD** und **SUMDV3**, die aufwärtskompatibel

zueinander sind. **SUMD** überträgt bis zu 32 Kanäle (je nach Sender aber meistens weniger), **SUMDV3** bis zu 32 Kanäle und 64 Schalter.

Bei neuerer Firmware auf den Empfängern können diese statt **SumD** auch **SBus** (s.a. **Aktoren: SBus**) erzeugen.

### 4.2.2. Sensoren: Hott-Sensor

Das Sensor-Protokoll fasst immer viele einzelne Sensoren zusammen. Diese Sensorgruppen werden dann etwa als **EAM** (*electric air module*) oder **GAM** (*general air module*) bezeichnet. Insgesamt existieren nur 6 unterschiedliche Sensor-Module (RX, GAM, EAM, ESC, Vario, GPS).

Dies hat zur Folge, dass man bei den Sendern mit monochromen LCD Anzeigen nur vorgefertigte Ansichten für ein Sensor-Modul auswählen kann. *Einzelne* Sensoren aus einem Modul können nicht einzeln angezeigt werden.

### 4.2.3. Besonderheiten

Das *Hott-Sensor*-Protokoll hat zudem noch zwei Modi: einen Daten- und einen Text-Modus. Im *Daten*-Modus werden die Sensorwerte eines Sensor-Moduls übertragen und am Sender angezeigt.

Im *Text*-Modus erzeugt das Sensor-Modul ein *Menu*, dieses wird dann im Sender angezeigt. Dies dient üblicherweise zum *Konfigurieren* des Moduls, etwas des Empfängers oder anderer angeschlossener Module wie etwa einen ESC. Im Sender gelangt man in diese spezielle Ansicht über **Menu** → **Telemetrie** → **Einstellungen**.

## 4.3. FrSky bzw. Futaba

Futaba hat ursprünglich den sog. **SBus** entwickelt. Er hat sich zu einem quasi-Standard entwickelt und wird trotz einiger Merkwürdigkeiten von vielen anderen Herstellern ebenfalls angeboten.

### 4.3.1. Aktoren: SBus

Eine der *Merkwürdigkeiten* ist, dass das elektrische Signal *invertiert* übertragen wird. Zudem ist die ursprüngliche Spezifikation so, dass +/- 12V als elektrische Pegel verwendet wird. Dies wird aber heute üblicherweise von den Mastern (Empfängern) nicht gemacht, sondern nur wie sonst üblich 0/3V bzw. 0/5V für die logischen Pegel verwendet. Andernfalls würden die angeschlossenen Slaves mit Sicherheit elektrisch zerstört. Trotzdem ist hier etwas Vorsicht geboten.

**SBus** überträgt immer 16 Kanäle (s.a. **SBus-PWM-Adapter**).

Achtung: nicht jeder FrSky-Empfänger hat **SBus** und **S.Port**.

### 4.3.2. Sensoren: S.Port

Die ist das *aktuelle* Sensorprotokoll, ein neueres, hierzu wieder inkompatibles heißt **FPort**. Auch bei **S.Port** wird mit *invertierten* Signalen gearbeitet.

### 4.3.3. Besonderheiten

Das **S.Port**-Protokoll ermöglicht auch die *Konfiguration* von angeschlossenen Slaves. Was damit genau möglich ist, ist aus der Bedienungsanleitung zu entnehmen. So kann bspw. bei **SBus**-Servos (ja, die gibt es) festgelegt werden, auf welchen *Kanal* der das Servo lauschen soll.

## 4.4. FlySky

FlySky hat versucht, die Schwachstellen des **Sbus** und **S.Port** zu vermeiden. Insbesondere sind die elektrischen Signale *nicht invertiert*, was es gerade für Selbstbauer interessant macht.

Achtung: nicht jeder FlySky-Empfänger hat **IBus**.

Hat ein FlySky-Empfänger **IBus** (etwa die **FS-iA`x`B**-Serie), so kann man den *Aktor-IBus*-Ausgang auch auf **SBus** umschalten. Dies *muss* durch den Sender erfolgen.

### 4.4.1. Aktoren: IBus

Es gibt zwei Varianten: **IBus-14** und **IBus-18**, die entsprechend jeweils 14 oder 18 Kanäle übertragen (s.a. **IBus-PWM-Adapter**).

### 4.4.2. Sensoren: IBus

Zwar ist der Sensor-**IBus** logisch ein Bus-System, technisch ist er jedoch als sog. **Daisy-Chain** ausgeführt. Dies bedeutet, dass die anzuschließenden Sensoren als *Sensor-Kette* anzuschließen sind: Empfänger → Sensor\_1 → Sensor\_2 → ...

Leider können in dieser Kette nur bis zu 15 Sensoren angeschlossen werden.

## 4.5. Spektrum:

### 4.5.1. Aktoren: XBus

### 4.5.2. Sensoren: XBus



## 5. Erweiterungen

### 5.1. SBus-PWM-Adapter

Hat man etwa einen kleinen 4-Kanal Empfänger, so kann man hier nur 4 Servos direkt anschließen. Vom Sender werden aber in der Regel *mehr* Kanäle übertragen. Um die weiteren Kanäle nutzen zu können, gibt es auf dem Markt sog. **SBus2PWM**-Adapter: sollte der Empfänger einen **SBus**-Ausgang besitzen, kann man diesen Adapter dort anschließen und bekommt bis zu 16 Kanal-Ausgänge.

### 5.2. IBus-PWM-Adapter

Es gilt dasselbe wie unter **SBus-PWM-Adapter**.

## 6. Analoges

Es gibt auch *analoge* Bus-Systeme, auch den der Begriff **Bus** originär aus der Informatik stammt und damit eigentlich immer *digitale* Bus-Systeme gemeint sind.

Der generelle Nachteil von den analogen Verfahren ist die erhöhte Störanfälligkeit.

### 6.1. PWM-Servosignal

Fast alle RC-Empfänger haben sog. Servo-Ausgänge. Es gibt jedoch mittlerweile einige Klein- und Kleinst-Empfänger, die aus Platzgründen auf die analogen Servo-Ausgänge verzichten, oder solche, die speziell für den *Copter*- oder *Dronen*-Betrieb ausgerichtet sind, wo eine digitale Übertragung alle Kanäle über ein Bus besser geeignet ist.

Man kann die *Verbindung* von einem Servosignal-Ausgang des RC-Empfängers zu einem Servo auch als einen *Bus* auffassen: in diesem Fall mit einem Master und nur *einem* Slave (dem Servo). Auf diesem Bus wird auch nur *eine* Information übertragen, eben genau die gewünschte Position für dieses eine Servo.

Es spricht aber nicht dagegen, auf zwei oder mehr Servos an demselben Servosignal-Ausgang anzuschließen. Diese Servos bewegen sich dann immer gleichförmig. Dies kann sinnvoll sein, wenn etwa an *räumlich getrennten* Orten *dieselben* Bewegungen ausgeführt werden sollen, etwa bei mehreren Rudern, die nicht durch ein Gestänge verbunden werden können (s.a. Langeklappen im Flugmodellbau). Dafür setzt man elektrisch gerne ein sog. *Y-Kabel* ein.



#### *Pulse-Width-Modulation (PWM)*

Die gewünschte Position des Servoantriebs wird hierbei als die zeitliche Impulslänge eines elektrischen Impulses analog codiert: eine Impulslänge von ca. 1,5ms bedeutet üblicherweise Neutralposition, ca. 2ms oder ca. 1ms die jeweiligen Endpositionen (leider ist dies auch von Hersteller zu Hersteller unterschiedlich, weswegen man auch bspw. ESCs *\_anlernen* muss, damit korrekt die Lagen erkannt werden). Diese Impulse werden aus historischen Gründen alle ca. 20ms gesendet. Die Aktualisierungsrate ist damit ca. 50Hz.

### 6.2. CPPM

Als **CPPM** (*combined pulse position modulation*) oder auch manchmal nur **PPM** wird die analoge Übertragung von bis zu 8 Servokanälen bezeichnet (s.a. **PPM**). Dies kommt einem digitalen Aktor-Bus schon recht nahe, werden hier doch mehrere Informationen übertragen. Dieses Verfahren findet heute nur noch selten statt, teilweise wird es *innerhalb* eines *Senders* eingesetzt, um die Kanäle zum HF-Modul zu übertragen. Sehr selten kommt es noch im Copter- oder Dronen-Bereich zum Einsatz, um die Kanäle zum *Flightcontroller* zu übertragen (hier herrscht **SBus** vor).

**CPPM** wird als *Zeitmultiplex*-Verfahren bezeichnet.

### 6.3. Doppeltes Zeitmultiplex

Die Kodierung von mehreren Informationen wiederholend und zeitlich nacheinander wird oft als *Multiplex*-Verfahren bezeichnet. Dies ist bei **CPPM** ja der Fall.

Im Funktions- und Schiffsmodellbau hat man oft Anforderung, dass man einfache *ein/aus* Schaltfunktionen auslösen möchte. Eine zum Geber proportionale Servoabtriebsbewegung ist hierbei nicht gefragt.

Natürlich kann man das mit einem propotionalen Übertragungskanal erreichen, indem man bspw. hierfür einfach die Extremlagen verwendet. Selbst bei 16 Übertragungskanälen ist man dann allerdings schnell an der Grenze des Machbaren, denn mehr als 16 Schaltfunktionen sind nicht selten.

Dasselbe Prinzip des Zeitmultiplex kann man nun auch bezogen auf einen *einzelnen* Übertragungskanal anwenden: hier werden dann - wieder nacheinander - die Schaltzustände übertragen. An einem analogen Servoausgang des RC-Empfängers werden also nacheinander die Schaltzustände als Servo-Impulse ausgegeben. Damit verringert sich natürlich die Aktualisierungsrate nochmals, die Schaltfunktionen reagieren damit nicht unmittelbar. Weil auch noch 1 bis 2 \_Synchronimpulse übertragen werden müssen, kann es bis zu 200ms dauern, bis eine Reaktion erfolgt.

### 6.3.1. Varianten

Damit das analoge Schaltmodul erkennen kann, an welcher Stelle bei den übertragenen Impulsen die Zählung für die Schaltfunktionen beginnt, muss ein sog. *Synchronimpuls* (oder zwei) übertragen werden. Als *Synchronimpulse* muss eine Impulslänge gewählt werden, die bei den anderen Impulsen *nicht* vorkommt.

Es haben sich Verfahren eingebürgert, die einen oder zwei extrem lange (>2ms) oder kurze Impulse (<1ms) verwenden, so dass sich insgesamt vier Möglichkeiten ergeben. Die Schaltzustände werden dann mit Impulsen von etwa 1,8ms, 1ms und 1,2ms übertragen. Damit kann jede Schaltfunktion drei Zustände annehmen: *aus*, *ein\_1* und *ein\_2*.

- 1 langer Synchronimpuls, danach bis zu 8 Schalt- oder Proportional-Impulse
  - CP-Elektronik
- 2 lange Synchronimpulse, danach bis zu 8 Schalt- oder Proportional-Impulse
  - Graupner / JR MK4
  - Graupner / JR MK8
- 1 kurzer Synchronimpuls, danach bis zu 8 Schalt- oder Proportional-Impulse
  - Robbe (in umgekehrter Reihenfolge)
- 2 kurze Synchronimpulse, danach bis zu 8 Schalt- oder Proportional-Impulse
  - kein Einsatz bekannt

## 7. Die Protokolle im Detail

### 7.1. SumD

### 7.2. Hott-Sensor

### 7.3. SBus

### 7.4. S.Port

### 7.5. IBus-Servo

### 7.6. IBus-Sensor

## 8. Produkte

### 8.1. SumD / Hott-Sensorprotokoll

- Fa. Graupner

### 8.2. SBus / S.Port

- Fa. FrSky
- Fa. Futaba
- Drittanbieter
  - (... viele ...)
  - WM

### 8.3. IBus

- Fa. FlySky
- Drittanbieter
  - WM

## 9. Kontakt

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