

## S-Port Telemetrie-Konverter

### Einsatzbereich

Abgesehen von Firmware-Versionen für verschiedenen Controller und Taktfrequenzen gibt es zwei grundsätzlich unterschiedliche Varianten des S-Port Konverters.

(1) MSB -> S-Port Konverter:

Die erste Variante empfängt Telemetrie-Daten von MSB-Sensoren und stellt diese über das S-Port Protokoll zur Einspeisung in einen FrSky Empfänger zur Verfügung. Die Funkstrecke ist in diesem Fall FrSky ACCST oder ACCESS.

(2) M-Link -> S-Port Konverter:

Die zweite Variante verarbeitet Telemetrie-Daten von einem M-Link Sendemodul und gibt diese über das S-Port Protokoll zur Einspeisung in einen Sender mit OpenTx aus. Diese Variante ersetzt den früheren Konverter, der das alte FrSky D-Protokoll benutzt. Die Funkstrecke ist in diesem Fall M-Link.

Hinweis: ETHOS unterstützt das Einspeisen von Telemetriedaten bei Verwendung eines externen Moduls mit PPM Ansteuerung nicht.

### Hardware

Der Konverter ist für folgende Mikrocontroller verfügbar.

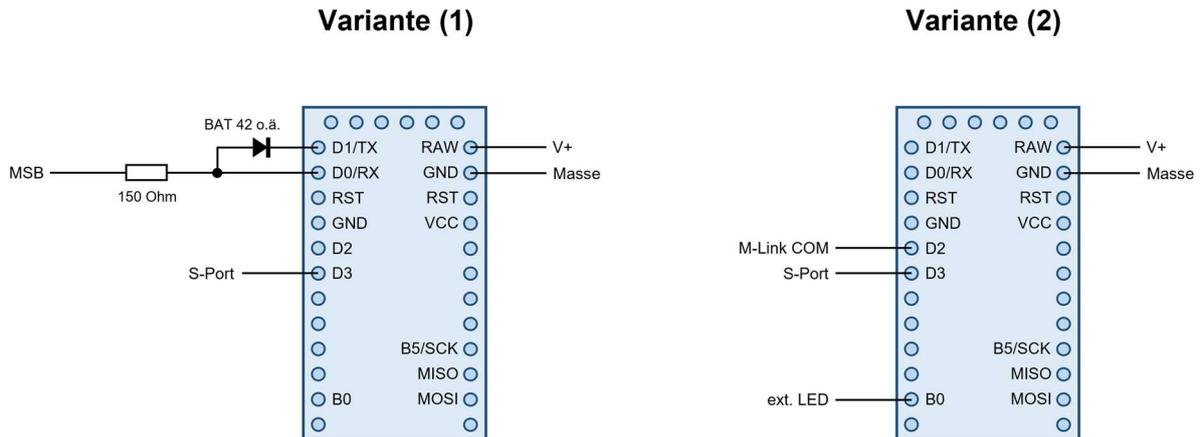
- ATmega328(P) mit 8MHz Takt
- ATmega328(P) mit 16MHz Takt
- ATtiny841 mit 8MHz Takt

Es können entsprechende Arduino Boards (oder Klone), aber auch DIY Boards verwendet werden. Es muss aber darauf geachtet werden, ggf. durch Verwendung eines entsprechenden Spannungsreglers zur Versorgung, dass an den seriellen Schnittstellen keine Signalpegel größer als 5V auftreten können. Außerdem sind natürlich die Grenzwerte für die Versorgungsspannung des µC zu beachten, wobei die Mindestspannung von der Taktfrequenz abhängt.

Folgende µC Ports werden für die Schnittstellen des Konverters verwendet.

	ATmega328(P)		ATtiny841	
	Variante (1)	Variante (2)	Variante (1)	Variante (2)
MSB (HW USART)	D0 + D1	-	A1 + A2	-
S-Port	D3	D3	B0	B0
M-Link HF-Modul	-	D2	-	A0
Status LED (Board)	B5	B5	B2	B2
Status LED (Extern)	-	B0	-	-

Ein Arduino Pro Mini Board wird gemäß folgender Abbildung verschaltet.



Die Verschaltung eines Wattuino Nanite 841 Boards oder eines DIY Boards muss nach eigenem Ermessen durchgeführt werden. Dabei ist zu beachten, dass das Nanite 841 Board keinen Spannungsregler besitzt.

#### Status-LED:

Wenn das Programm auf einem Arduino Pro Mini oder Wattuino Nanite 841 Board läuft, wird der Status des Konverters über die auf dem Board vorhandene LED angezeigt.

Beim Arduino Pro Mini mit der Firmware für die Variante (2) liegt am Port B0 ein identisches Ansteuerungs-Signal für eine externe Status-LED an. Diese kann beim Einbau des Konverters in den Sender (z.B. in der Taranis X9E) in einem freien Schalterloch platziert und über einen passenden Vorwiderstand mit dem Port B0 und Masse verbunden werden. Die externe Status-LED arbeitet dann synchron mit der LED auf dem Arduino Board.

Der Start bzw. Reset des Konverters wird durch dreimaliges Ausführen einer speziellen Blinksequenz signalisiert.

- (1) MSB -> S-Port:            kurz - lang - kurz
- (2) M-Link -> S-Port:        kurz - kurz - lang

Danach haben die verschiedenen Blinkmuster der Status LED folgende Bedeutung.

	Variante (1)	Variante (2)
LED Aus	Der Konverter ist Aus.	Der Konverter empfängt keine Datenpakete vom M-Link Modul. (z.B. weil es für das ausgewählte Modell benutzt wird).
Dauerleuchten (wenige Sekunden)	Kommt nicht vor.	Der Konverter ignoriert die empfangenen Datenpakete.
Schnelles Blinken	Auf dem MSB antwortet kein Sensor.	Der Konverter empfängt vom Modul Datenpakete ohne gültige Telemetrie-Daten (z.B. weil der Empfänger ausgeschaltet ist.)
Langsames Blinken (normaler Betrieb)	Der Konverter empfängt gültige Sensordaten am MSB und gibt entsprechende Datenpakete am S-Port aus.	Der Konverter empfängt gültige Telemetrie-Daten vom M-Link Modul und gibt entsprechende Datenpakete am S-Port aus.

## **Multiplex Sensor Bus (MSB)**

Der Konverter reagiert dynamisch auf Änderungen am MSB, so dass es kein Problem ist, wenn ein Sensor eine längere Hochlaufzeit hat (z.B. GPS). Sobald gültige Telemetrie-Daten von diesem Sensor auf dem MSB vorhanden sind, werden sie vom Konverter verarbeitet. Man kann auch während des laufenden Betriebs Sensoren am MSB an- und abstecken. Wenn ein Sensor entfernt wird, werden die ihm zugeordneten S-Port Parameter nicht mehr gesendet.

Der Konverter als Variante (1) übernimmt die Rolle des Bus Masters am MSB, die normalerweise der M-Link Empfänger einnimmt. Das nominale Abfrageintervall am MSB beträgt 6ms, wobei das Abfrageschema der 16 MSB-Adressen dynamisch an die antwortenden Sensoren angepasst wird. Wenn eine definierte Anzahl aufeinanderfolgender Abfragen einer bestimmten MSB-Adresse nicht beantwortet wird, werden die entsprechenden Sensordaten als ungültig markiert.

Wenn der Konverter als Variante (2) keine gültigen Telemetrie-Daten mehr vom M-Link Sendemodul empfängt (z.B. weil der Empfänger ausgeschaltet wurde), schaltet er die Ausgabe von Telemetrie-Daten am S-Port ab, und der Sender erzeugt die entsprechende Sprachwarnung. Das ist natürlich mit einer gewissen zeitlichen Verzögerung verbunden, die davon abhängt, wie lange das M-Link Sendemodul braucht, um auf die Veränderung zu reagieren.

### **MSB-Adressen:**

Grundsätzlich können die MSB-Adressen frei gewählt werden. Es gibt allerdings einige Fälle, wo unterschiedliche S-Port Parameter mit MSB-Parametern gleicher Werteklasse assoziiert sind, so dass der Konverter sie auf dem MSB an Hand der Werteklasse nicht unterscheiden kann. In diesen Fällen wird es durch die Festlegung bestimmter MSB Adressen ermöglicht, eine Unterscheidung zu treffen, so dass der Konverter diese Parameter richtig zuordnen kann.

Die vordefinierten MSB-Adressen sind in folgender Tabelle zusammengefasst.

<b>Parameter</b>	<b>Werteklasse</b>	<b>MSB-Adresse</b>
Empfängerspannung	1	0
Baro-Höhe	8	6

Spannungen, die nicht die vordefinierte MSB Adresse für die Empfängerspannung haben, werden natürlich trotzdem als ganz normale Spannungswerte übertragen. Eine Baro-Höhe, die nicht auf der für sie vorgesehenen MSB Adresse vorhanden ist, wird am S-Port als DIY Parameter ausgegeben.

Jede dieser MSB-Adressen kann aber auch für einen beliebigen Parameter mit einer anderen Werteklasse benutzt werden, wenn der Parameter, für den die Adresse reserviert ist, nicht vorhanden ist.

### **Min./Max. Werte:**

Da die Min. und Max. Werte aller Telemetrie-Parameter sowohl in ETHOS als auch in OpenTx standardmäßig immer zur Verfügung stehen, sollten in den MSB-Sensoren keine Min. und Max. Werte auf separaten MSB-Adressen konfiguriert werden.

### **Alarm Flags:**

Sowohl in ETHOS als auch in OpenTx werden die Alarime im Sender konfiguriert, daher ignoriert der Konverter alle am MSB vorhandenen Alarm-Flags.

## **S-Port Sensor IDs**

Es gibt 28 verschiedene Sensor (Physical) IDs, von denen ein großer Teil bereits für bestimmte FrSky Sensoren reserviert ist. Damit ein Mischbetrieb mit originalen FrSky Sensoren möglich ist, verwendet der Konverter andere Sensor IDs.

Die meisten Parameter werden vom Konverter unter einer einzigen gemeinsamen Sensor ID übertragen, nur für die Vario- und Empfängerdaten werden separate Sensor IDs verwendet. Die Verwendung einer eigenen Sensor ID für die Variodaten verbessert deren Datenrate, während die Sensor ID für die Empfängerdaten im S-Port Protokoll fest vorgegeben ist. Somit werden vom Konverter folgende Sensor IDs emuliert.

- 11 (hex. 0B) -> gemeinsame Sensor ID
- 13 (hex. 0D) -> Variodaten (Baro-Höhe, Steigrate)
- 24 (hex. 18) -> Empfängerdaten (RSSI, Empfängerspannung)

Bei der Konverter Variante (1) kann die ID für die Empfängerdaten nicht vorkommen, da die zugeordneten Parameter vom FrSky Empfänger selbst erzeugt werden.

## **S-Port Daten IDs**

Soweit möglich und sinnvoll, wird jeder MSB-Parameter am S-Port mit einer seiner Werteklasse entsprechenden Daten (Application) ID ausgegeben (unter Berücksichtigung der vordefinierten MSB-Adressen).

Wenn es für eine Werteklasse keine spezielle Daten ID gibt, oder wenn deren Verwendung deutliche Nachteile mit sich bringen würde, wird für diese Werteklasse eine Daten ID aus dem Bereich für DIY Sensoren verwendet. Beispielsweise haben Temperaturen auf dem MSB eine bessere Auflösung als im S-Port Protokoll. Um diese beizubehalten, verwendet der Konverter (statt der speziellen Daten IDs für Temperaturen) hier DIY IDs, und der Telemetriewert wird im Sender so konfiguriert, dass sich die richtige Anzeige ergibt.

Für alle relevanten S-Port Parameter (außer RSSI und Empfängerspannung) gibt es einen lückenlosen Bereich von 16 gleichwertigen Daten IDs. Die Daten ID für einen bestimmten Parameter wird vom Konverter gebildet, indem zur ersten Daten ID des entsprechenden Bereichs die MSB-Adresse als Offset addiert wird. Da das Low Nibble der ersten Daten ID jedes Bereichs 0 ist, reflektiert das Low Nibble der Daten ID eines Sensorwerts die zugehörige MSB-Adresse.

RSSI und Empfängerspannung) haben spezielle eindeutige Daten IDs und werden wie folgt aus den MSB-Parametern ermittelt.

- Wenn unter der für die Empfängerspannung festgelegten MSB-Adresse ein gültiger Spannungswert vorhanden ist, wird dieser als Empfängerspannung übertragen.
- Als RSSI wird der M-Link LQI übertragen, wobei die niedrigste MSB-Adresse mit passender Werteklasse verwendet wird.

Die folgende Tabelle enthält für alle MSB-Wertklassen die physikalische Einheit und Auflösung des MSB-Werts, sowie die vom Konverter verwendeten S-Port Daten IDs.

Werteklasse	Beschreibung	Einheit / Auflösung	S-Port Daten IDs
0	Sonderdaten	-	0x5100 ... 0x510F (DIY ID)
1	Spannung	0,1 V	0x0210 ... 0x021F 0xF104 (Empfängerspannung)
2	Strom	0,1 A	0x0200 ... 0x020F
3	Steigrate	0,1 m/s	0x0110 ... 0x011F
4	Geschwindigkeit	0,1 km/h	0x5100 ... 0x510F (DIY ID)
5	Drehzahl	10/100 U/min *)	0x0500 ... 0x050F
6	Temperatur	0,1 °C	0x5100 ... 0x510F (DIY ID)
7	Richtung	0,1 °	0x5100 ... 0x510F (DIY ID)
8	Höhe / kurze Distanz	1 m	0x0106 (Baro-Höhe) 0x5100 ... 0x510F (DIY ID)
9	Füllstand	1 %	0x0600 ... 0x060F
10	LQI	1 %	0xF101 (RSSI)
11	Ladung (Verbrauch)	1 mAh	0x5100 ... 0x510F (DIY ID)
12	Volumen	1 ml	0x0A10 ... 0x0A1F
13	lange Distanz	0,1 km	0x5100 ... 0x510F (DIY ID)
14	g-Rate	0,1 g	0x0700 ... 0x070F **)
15	<i>nicht definiert</i>	-	-

\*) Für die Drehzahl gibt es auf dem MSB zwei mögliche Auflösungen (10 U/min und 100 U/min), wobei eine Auflösung von 10 U/min durch ein negatives Vorzeichen gekennzeichnet wird.

\*\*\*) Alle am MSB vorhandenen g-Raten werden mit einer Daten ID aus dem im S-Port Protokoll definierten Bereich für die x-Achse übertragen. Die Namen müssen dann im Sender so angepasst werden, dass sie die Achsen der auf dem MSB vorhandenen g-Raten reflektieren. Da die Anzahl von g-Raten auf dem MSB variieren kann, wäre ansonsten eine komplizierte Logik nötig, um die MSB-Parameter den Achsen richtig zuzuordnen. Außerdem müssten bestimmte Regeln bei der Vergabe der MSB-Adressen beachtet werden, und die Anzahl von g-Raten Parametern wäre beschränkt.

#### Sonderdaten:

Es werden auch MSB-Daten mit Werteklasse 0 übertragen, sofern eine gültige Unterklasse im High Byte des MSB-Sensorwerts vorhanden ist. Derzeit gibt es nur die Unterklasse 0x01 (ECU Status- und Fehlermeldungen).

Für Sonderdaten der Unterklasse 0x01 reflektieren die untersten 7 Bits des übertragenen S-Port Parameters die oberen 7 Bits aus dem Low Byte des MSB-Sensorwerts. Die Unterklasse wird nicht mitübertragen, so dass der Wert des S-Port Parameters direkt dem übertragenen Code der ECU Meldung entspricht. Die erlaubten Codes sind in der MSB-Spezifikation definiert.

#### Umrechnung und Einstellungen im Sender

Der MSB-Sensorwert wird in die Einheit und Auflösung des entsprechenden S-Port Parameters umgerechnet. Für eine optimale Darstellung auf dem Senderdisplay sind dann unter Umständen die Default Einstellungen, die durch die zugeordnete Daten ID festgelegt sind, abzuändern.

Manche Parameter auf dem MSB können negative Werte haben, die aber über den S-Port nicht übertragen werden, da sie der Sender nicht erkennen würde. In diesen Fällen wird der absolute Betrag des MSB-Werts verwendet.

Für eine Reihe von Parametern ist die Auflösung des Werts am S-Port besser (kleiner) als die des entsprechenden Werts auf dem MSB. Dann sollte man die Präzision der Anzeige im Sender entsprechend anpassen, um permanente Nullen in den hinteren Stellen zu vermeiden.

In ETHOS muss für alle Parameter mit einer Daten ID aus dem DIY Bereich ein DIY Sensor angelegt und entsprechend konfiguriert werden.

Beide auf dem MSB möglichen Auflösungen für Drehzahlen werden vom Konverter in U/min mit positivem Vorzeichen umgerechnet.

Eine Besonderheit gibt es bei den vom Multiplex GPS gelieferten Werten für die zurückgelegte Wegstrecke (2D oder 3D). Hier wird der Wert auf dem MSB bis zu einer Strecke von 10km mit Werteklasse 8 (Auflösung 1m), darüber jedoch mit Werteklasse 13 (Auflösung 0,1km) übertragen. Da im Sender eine Umschaltung der Einheit (abhängig vom Zahlenwert) nicht möglich ist, werden auch die mit Werteklasse 13 übertragenen Zahlenwerte in m umgerechnet. Das Umschalten der Werteklasse hat dann keinen Einfluss auf die Darstellung im Sender, da sich dabei ja die MSB-Adresse (und damit die Daten ID) nicht ändert.

Die folgenden beiden Tabellen definieren für jeden vom Konverter benutzten Bereich von S-Port Daten IDs die notwendige Sensorkonfiguration im Sender für OpenTx bzw. ETHOS. Die Präzision entspricht dabei der Anzahl von Nachkommastellen.

Oft ist es sinnvoll, den Namen nach seinen eigenen Bedürfnissen zu ändern, in jedem Fall sollte man bei Konverter Variante (2) den RSSI in LQI umbenennen.

#### OpenTx / EdgeTX Sensorkonfiguration:

S-Port Daten IDs	Sensorkonfiguration		
	Name	Einheit	Präzision
0x0100 ... 0x010F	Alt	m	0
0x0110 ... 0x011F	VSpd	m/s	1
0x0200 ... 0x020F	Curr	A	1
0x0210 ... 0x021F	VFAS	V	1
0x0500 ... 0x050F	RPM	rpm	0
0x0600 ... 0x060F	Fuel	%	0
0x0700 ... 0x070F	AccX	g	1
0x0A10 ... 0x0A1F	Fuel	ml	0
0x5100 ... 0x510F	510x *)		
Geschwindigkeit		km/h	1
Temperatur		°C	1
Richtung		°	1
Distanz / GPS-Höhe		m	0
Ladung		mAh	0
0xF101	RSSI	%	0
0xF104 **)	RxBt	V	1

\*) x entspricht der MSB-Adresse des Parameters.

\*\*) Für die Empfängerspannung muss ein Umrechnungsfaktor von 25,5 eingestellt werden.

**ETHOS Sensorkonfiguration:**

S-Port Daten IDs	Sensorkonfiguration		
	Name	Einheit	Präzision
0x0100 ... 0x010F	Flughöhe	m	0
0x0110 ... 0x011F	Steigrate Vario	m/s	1
0x0200 ... 0x020F	FASS	A	1
0x0210 ... 0x021F	VFAS	V	1
0x0500 ... 0x050F	U/min	U/min	0
0x0600 ... 0x060F	Kraftstoff	%	0
0x0700 ... 0x070F	AccX	g	1
0x0A10 ... 0x0A1F	DIY0A1x *)	ml	Protokoll: 2 Anzeige: 0
0x5100 ... 0x510F	DIY510x *)		
Geschwindigkeit		km/h	1
Temperatur		°C	1
Richtung		°	1
Distanz / GPS-Höhe **)		m	0
Ladung		mAh	0

\*) x entspricht der MSB-Adresse des Parameters.

\*\*) Mit ETHOS können auch drei Nachkommastellen angezeigt werden, daher kann für diese Parameter alternativ auch die Einheit km mit Präzision 3 für das Protokoll und Präzision 1 für die Anzeige gewählt werden.

Normalerweise wird für DIY Sensoren die Präzision für Protokoll und Anzeige gleich eingestellt. Unterschiedliche Werte sind nur dann erforderlich, wenn der Konverter (der ja nicht zwischen OpenTX und ETHOS unterscheidet) für die Übertragung eines bestimmten Parameters die von OpenTX erwartete Auflösung einhalten muss, die entsprechende MSB Werteklasse aber eine schlechtere (größere) Auflösung hat. Das ist beispielsweise für die Übertragung einer Füllmenge der Fall, da hier OpenTx eine Auflösung von 0,01 ml erwartet, während die MSB Auflösung für die Werteklasse 12 (Volumen) 1 ml ist. Würde man hier für die Anzeige ebenfalls Präzision 2 einstellen, dann wären die Nachkommastellen immer 0.

Unter Umständen muss der Wertebereich für einzelne Sensoren vergrößert werden, damit alle relevanten Sensorwerte darin enthalten sind und richtig angezeigt werden.

Es gibt Daten IDs, die ETHOS mit der normalen Sensorsuche nicht findet. Dazu gehören die IDs für die "Fuel Quantity" (entspricht MSB-Werteklasse 12) und die speziellen DIY Daten IDs. In diesen Fällen muss man einen DIY Sensor anlegen und im entsprechenden Menü dann die automatische Erkennung aktivieren.