

## 1. Topologie, Grundsätzliches

Sowohl SLSi als auch cSLSi unterstützt eine serielle (RS232) Schnittstelle u.a. zum Auslesen von Status-Information. Das Software-Protokoll ist für alle cSLSi/SLSi identisch.

Grundsätzlich ist nach dem Senden eines Befehls das Antwort-Telegramm abzuwarten. Erst nach Erhalt der Antwort darf ein neuer Befehl gesendet werden. Die Kommunikation kann nur vom Host initiiert werden. Die Einstellungen der seriellen Schnittstelle sind 115kBd, 1Startbit, 8 Datenbits, 1 Stoppbit, keine Parität.

### 1.1 SLS

Die serielle Verbindung ist beim SLS über eine 3,5mm Stereo-Klinkenbuchse realisiert. Die Pinbelegung des 3,5mm Klinkensteckers des Verbindungskabels ist nachfolgend gezeigt:



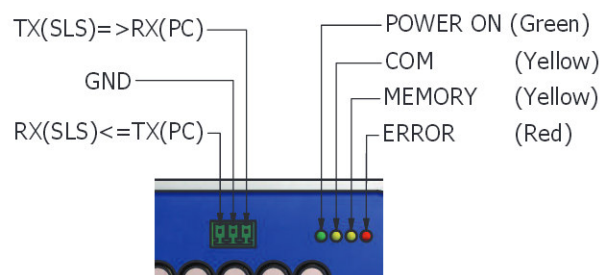
### 1.2 cSLSi

Die serielle Verbindung ist beim cSLS über ein Buchsenleiste im 2,54mm Raster mit runden 0,5mm Pins realisiert. Der markierte PIN zeigt zu den LEDs.



### 1.3 SLSi

Die Pinbelegung des Steckers am SLSi finden Sie nachfolgend:



## **2. Protokoll**

### 2.1 Struktur des Protokolls

Byte 0	Byte 1	Byte 2	Byte 3 .. (A-1)	Byte A
S	A	T	D	CRC

- Byte 0: Das erste Byte (S) in einem Kommunikations-Rahmen ist als **Sync-Byte** implementiert. Dabei kommt das Zeichen '!' für eine initiale Kommunikation vom Host zum SLS zum Einsatz. Das rücklaufende Paket enthält dagegen das Sync-Zeichen '?'.
- Byte 1: Dieses Byte (A) dient als **Zähler**. Es werden die Bytes von (Byte 0 .. Byte (A-1)) gezählt. **Der tatsächliche Wert des Zählers kann sich mit einer späteren Firmware-Version erhöhen! Wir empfehlen daher den Zähler von Anfang an als Variable in der Host-Software anzulegen!** (Natürlich bleiben neue Firmware-Version abwärtskompatibel, sofern unten im Detail beschrieben.)
- Byte 2: Diese Byte (T) stellt den eigentlichen **Befehl/Tag** dar, der vom SLS abgearbeitet werden soll. Einzelheiten siehe weiter unten. Rücklaufende Pakete schicken dieses Byte unverändert zurück.
- Byte 3..(A-1): Optionale **Daten** bzw. Parameter für den auszuführenden Befehl bzw. Ergebnisdaten im Antwort-Telegramm.
- Byte A: Nach den Datenbytes wird noch eine **Checksumme** (CRC) angehängt, die aus der Summe aller Bytes (excl. der Checksumme selbst) besteht.

Hin- und rücklaufende Pakete haben die gleiche, oben beschriebene Struktur.

## 2.2 Status-Abfrage:

**Status anfordern:**

Host sendet an SLS:

Byte 0	Byte 1	Byte 2	Byte 3
'!	3 <sub>d</sub>	'S'	CRC

**ACK: Status Rückmeldung:**

SLS sendet an Host:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
'?	65 <sub>d</sub>	'S'	TP	UZK_L	UZK_H	Iq_L	Iq_H

Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15
RPM_L	RPM_H	T_F	U_F	C_F	--	--	--

Byte 16	Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	Byte 22	Byte 23
Tmax_DR	Umin_DR	Umax_DR	AMPS_H	AMPS_L	MaxRPM_H	MaxRPM_L	--

Byte 24	Byte 25	Byte 26	Byte 27	Byte 28	Byte 29	Byte 30	Byte 31
--	--	--	--	RPM_Limit_L	RPM_Limit_H	MtrCur_Limit_L	MtrCur_Limit_H

Byte 32	Byte 33	Byte 34	Byte 35	Byte 36	Byte 37	Byte 38	Byte 39
RegCur_Limit_L	RegCur_Limit_L	--	--	--	--	--	--

Byte 40	Byte 41	Byte 42	Byte 43	Byte 44	Byte 45	Byte 46	Byte 47
--	--	Id_L	Id_H	--	--	--	--

Byte 48	Byte 49	Byte 50	Byte 51	Byte 52	Byte 53	Byte 54	Byte 55
--	--	--	--	--	--	--	--

Byte 56	Byte 57	Byte 58	Byte 59	Byte 60	Byte 61	Byte 62	Byte 63
--	--	--	--	TE	--	--	--

Byte 64	Byte 65
--	CRC

**TP** (Powermodultemperatur): Umrechnung auf °C:

$$T = -178,4 + \left( 249 * \sqrt{\frac{854}{598 - TP} - 1} \right)$$

( $0 \leq TP \leq 255$  wird abgebildet auf  $-15,5^\circ\text{C} \leq T \leq +125,4^\circ\text{C}$ )

**TE** (Elktemperatur (nur bei SLSi)): Umrechnung auf °C: siehe TP

**UZK\_L** (low-Byte)

**UZK\_H** (high-Byte):

Umrechnung:  $U[\text{V}] = (\text{UZK} * \text{MaxUZK}) / 1023$

mit MaxUZK:

24V Regler: 27,78V

42V Regler: 46,67V

60V Regler: 66,11V

(Skalierungsfaktor für die Spannung in Volt)

**AMPS\_L**

**AMPS\_H** :

maximal freigebbarer Strom in 0,1 A Schritten

**Iq\_L**

**Iq\_H**:

aktueller IST-Strom (Iq)

Umrechnung:  $I[\text{A}_{\text{eff}}] = \text{Iac} * \text{AMPS} / 10 / 4095$

Bit 15 ist das Vorzeichenbit

**Id\_L**

**Id\_H** :

aktueller IST-Strom(Id)

Umrechnung:  $I[\text{A}_{\text{eff}}] = \text{Id} * \text{AMPS} / 10 / 4095$

Bit 15 ist das Vorzeichenbit

**MtrCur\_Limit\_L**

**MtrCur\_Limit\_H** :

im Datensatz freigegebener Motorstrom

Umrechnung:  $\text{IMot}[\text{A}_{\text{eff}}] = \text{MtrCur\_Limit} * \text{AMPS} / 10 / 4095$

Bit 15 ist das Vorzeichenbit

**RegCur\_Limit\_L**

**RegCur\_Limit\_H** :

im Datensatz freigegebener Rückspeisestrom

Umrechnung:  $\text{IGen}[\text{A}_{\text{eff}}] = \text{RegCur\_Limit} * \text{AMPS} / 10 / 4095$

Bit 15 ist das Vorzeichenbit

**MaxRPM\_L**  
**MaxRPM\_H :** maximal freigebbare Motordrehzahl in U/min

**RPM\_L**  
**RPM\_H:** aktuelle IST-Drehzahl  
 Umrechnung:  $N[U/min] = (RPM * MaxRPM) / 10922$   
 Bit 15 ist das Vorzeichenbit (Drehrichtung)

**RPM\_Limit\_L**  
**RPM\_Limit\_H :** im Datensatz freigegebene Drehzahl  
 Umrechnung:  $N [U/min] = (RPM\_Limit * MaxRPM) / 10922$   
 Bit 15 ist das Vorzeichenbit

**T\_F** (Übertemperaturfehler):  
 Bit 7-0

R-0	R-0	R-0	U-0	U-0	U-0	U-0	U-0
SO_T	CMT	LMT	--	--	--	--	--

**Der Wert in geschweiften Klammern ist dem Windows-Monitor unter Parameter->SLS zu entnehmen!**

Bit 7 **SO\_T:** SwitchOff OverTemp (TP > 100°C, TE > 112°C) =>Failsafe (SLS stopped)  
 Bit 6 **CMT:** CutOff MaxTemp (TP > 90°C, TE > 106°C)  
 Bit 5 **LMT:** Limit MaxTemp(TP > {Temp\_Lim}, TE > 90°C)  
 Bit 4-0 **unimplemented:** read as '0'

**U\_F** (Über- /Unterspannungsfehler):  
 Bit 7-0

R-0	R-0	R-0	U-0	R-0	R-0	R-0	U-0
SO_OV	CMV	LMV	--	SO_UV	CUV	LUV	--

**Die Werte in geschweiften Klammern sind dem Windows-Monitor unter Parameter->Batterie zu entnehmen!**

Bit 7 **SO\_OV:** SwitchOff OverVolt (24V Regler: U > 25,66V  
 42V Regler: U > 43,11V  
 60V Regler: U > 61,07V)  
 Bit 6 **CMV:** CutOff MaxVolt (U > {U\_Batt\_max})  
 Bit 5 **LMV:** Limit\_MaxVolt (U > {U\_Batt\_max} - 1V)  
 Bit 4 **unimplemented:** read as '0'  
 Bit 3 **SO\_UV:** SwitchOff UnderVolt (24V Regler: U < 7,98V  
 42V Regler: U < 15,51V  
 60V Regler: U < 15,51V  
 bis SN599 bei den SLS: U < 22,49V)  
 Bit 2 **CUV:** CutOff\_MinVolt (U < {U\_Batt\_Low})  
 Bit 1 **LUV:** Limit\_MinVolt (U < {U\_Batt\_Lim})  
 Bit 0 **unimplemented:** read as '0'

**C\_F** (Fehler der internen Kontrolle):

Bit 7-0

R-0	U-0	R-0	U-0	R-0	U-0	U-0	R-0
PL_F	--	ZS_F	I_F	OS_F	LL_F	2PH_F	FS

Bit 7	<b>PL_F:</b> PhaseLoss_Flt -> stop+retry
Bit 6	<b>unimplemented:</b> read as '0'
Bit 5	<b>ZS_F:</b> ZeroSpd_Flt -> stop+retry
Bit 4	<b>I_F:</b> I_Offset_Flt -> stop+retry
Bit 3	<b>OS_F:</b> OvrSpd_Flt -> stop+retry
Bit 2	<b>LL_F:</b> Loadless_Fault -> stop+retry
Bit 1	<b>2PH_F:</b> 2-Phasen-PWM
Bit 0	<b>FS:</b> Failsafe_STOP (SLS stopped) -> check -> clear Error

**Tmax\_DR** (Temp Max derate Register):**Umin\_DR** (Überspannungs derate Register):**Umax\_DR** (Unterspannungs derate Register):

Diese 3 Register geben genauer an, wie weit abgeregelt wird. Wertebereiche für die 3 Derate-Register:  
 0x40 (Es wird noch nicht abgeregelt) bis  
 0x00 (Regler schaltet ab)

**NACK: Übertragungsfehler:**

SLS sendet an Host:

Byte 0	Byte 1	Byte 2	Byte 3
'?'	3 <sub>d</sub>	'?'	CRC

## 2.2.1 Servosignal überschreiben:

**Status anfordern:**

Host sendet an SLS:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
'!	7 <sub>d</sub>	'S'	1 <sub>d</sub>	Aktiv	Signal_L	Signal_H	CRC



Um einen Timeout zu verhindern muss das TAG "Servosignal überschreiben" zyklisch gesendet werden. Der Timeout beträgt 300ms.

**Aktiv :**

0xAA Signal wird überschrieben.

0x00 Signal wird nicht überschrieben.

**Signal\_L****Signal\_H :**Signalvorgabe in  $\mu\text{s}$  zulässiger Bereich:800 .. 2200  $\mu\text{s}$ **ACK: Status Rückmeldung:**

SLS sendet an Host siehe 2.2

## 2.2.2 Servosignal offset:

**Status anfordern:**

Host sendet an SLS:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
'!	5 <sub>d</sub>	'S'	2 <sub>d</sub>	Servooffset	CRC

**Servooffset :**Servosignaloffset in  $\mu\text{sec}$ Wertebereich: + / - 127  $\mu\text{sec}$ 

Der gesendete Offset wird dauerhaft im Regler gespeichert.

**ACK: Status Rückmeldung:**

SLS sendet an Host siehe 2.2

## 2.2.3 Kontrolle über Bedienpult (kostenpflichtige Option):

**Status anfordern:**

Host sendet an SLS:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
'!	14 <sub>d</sub>	'S'	3 <sub>d</sub>	Aktiv	Control	SPD_WM <sub>on_L</sub>	SPD_WM <sub>on_H</sub>

Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14
MtrCur_W Mon_L	MtrCur_W Mon_H	RegCur_W Mon_L	RegCur_W Mon_H	Accel_WM on	Decel_WM on	CRC



Um einen Timeout zu verhindern muss das TAG "Kontrolle über Bedienpult" zyklisch gesendet werden. Der Timeout beträgt 300ms.

**Aktiv :** 0xAA Signal wird überschrieben.  
0x00 Signal wird nicht überschrieben.

**Control:**

Bit 7-0

U-0	U-0	U-0	U-0	U-0	W	W	W
--	--	--	--	--	Direction	Start/Stop	Feststellbremse_Aktiv

Bit 7-3 **unimplemented:** read as '0'  
 Bit 2 Direction  
 Bit 1 Stop = 0; Start = 1  
 Bit 0 Feststellbremse aktiv = 1, der Motor wird durch Kurzschluss der 3 Phasen gebremst.

**SPD\_WM\_L****SPD\_WM\_H :**

Drehzahlvorgabe, Wertebereich: 0 .. 1023  
 1023 entspricht der im Datensatz freigegebenen Motordrehzahl  
**(RPM\_Limit)**

**MtrCur\_WMon\_L****MtrCur\_WMon\_H :**

Motorstromvorgabe, Wertebereich: 0 .. 1023  
 1023 entspricht dem im Datensatz freigegebenem Strom  
**(MtrCur\_Limit)**

**RegCur\_WMon\_L****RegCur\_WMon\_H :**

Rückspeisestromvorgabe, Wertebereich: 0 .. 1023  
 1023 entspricht dem im Datensatz freigegebenem Strom  
**(RegCur\_Limit)**

**Accel\_WMon:****Decel\_WMon:**

Beschleunigungsrate, Wertebereich: 1 .. 255  
 Bremsrate, Wertebereich: 1 .. 255

**ACK: Status Rückmeldung:**

SLS sendet an Host siehe 2.2



## 2.3 Fehlerrücksetzen/Reset

**Fehler rücksetzen:**

Host sendet an SLS:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
'!	4 <sub>d</sub>	'R'	PAR	CRC

**PAR:****Bit 7-0**

W-0	U-0	U-0	W-0	U-0	U-0	U-0	U-0
Reboot	0	0	Clear	0	0	0	0

Bit 7            **Reboot:** komplette Software wird neu gestartet (via Reset-Vektor)Bit 6-5        **unimplemented:** write as '0'Bit 4            **Clear:** alle Fehler werden zurückgesetztBit 3-0        **unimplemented:** write as '0'**ACK -> Quittierung:**

SLS sendet an Host:

Byte 0	Byte 1	Byte 2	Byte 3
'?'	3 <sub>d</sub>	'R'	CRC

**NACK -> Übertragungsfehler:**

SLS sendet an Host:

Byte 0	Byte 1	Byte 2	Byte 3
'?'	3 <sub>d</sub>	'?'	CRC