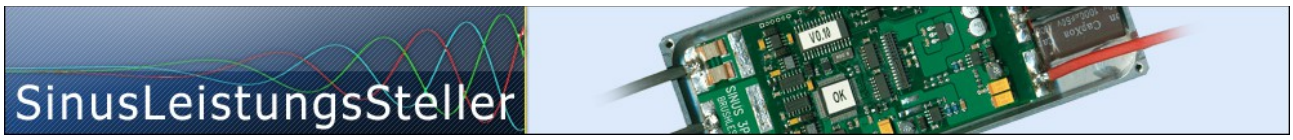

Serial - Protocol - SLS

Low-level documentation
needed for implementation
of own host software.

(based on Firmware V1.514, released 26.August 2021)

Ing.-Büro Zimmermann - Industriestraße 7 - D-97297 Waldbüttelbrunn
Tel.: +49 (0) 931/78011030 - www.SinusLeistungsSteller.de - info@SinusLeistungsSteller.de

Inhaltsverzeichnis

1.Structure of the protocol.....3
2.Status-query:.....4
2.1. Servo signal override:.....8
2.2.Servosignal offset:.....8
2.3.Option Control Panel (chargeable):.....9
3.Error reset/ SW-Reset.....10

1. Structure of the protocol

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

Byte 0: The first byte (S) in a communication frame is implemented as a sync byte. Here comes the sign '!' for an initial communication from the host to the SLS. The returning packet, on the other hand, contains the sync character '?'.

Byte 1: This byte (A) serves as a counter. The bytes of (Byte 0 .. Byte (A-1)) are counted. **The actual value of the counter may increase with a later firmware version! Therefore we recommend to set the counter from the beginning as a variable in the host software!** (Of course, new firmware version remains backwards compatible, as described in detail below.)

Byte 2: This byte (T) represents the actual **command / tag** to be processed by the SLS. For details see below. Returned packets return this byte unchanged.

Byte 3..(A-1): Optional data or parameters for the command to be executed or result data in the response telegram.

Byte A: After the data bytes, a checksum (CRC) is added, which consists of the sum of all bytes (excluding the checksum itself).

Leading and return packets have the same structure as described above.

The software-protocol is identical for all SLS.
Basically, the response message is seen after sending a command. Only after receiving the response, a new command can be send. The communication can only be initiated by the host. The settings of the serial interface are 115kBd, 1 start bit, 8 data bits, 1 stop bit, no parity.

2. Status-query:

Status request:

Host sends to SLS:

Byte 0	Byte 1	Byte 2	Byte 3
'!'	3_d	'S'	CRC

ACK: Status feedback:

SLS sends to Host:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
'?'	65_d	'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
--	--	Signal_L	Signal_H	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 (Powermoduletemperature): Conversion to °C:

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

(0<=TP <=255 is displayed on -15,5°C<=T<= +125,4°C)

TE (Elkotemperature (only by some SLSi)): Conversion to °C: see TP

UZK_L (low-Byte)
U**ZK_H** (high-Byte): Conversion: U[V] = (UZK * MaxUZK) / 1023

with MaxUZK:
24V ECU: 27,78V
42V ECU: 46,67V
60V ECU: 66,11V
(Scaling factor for the voltage in volts)

AMPS_L
AMPS_H : maximum <u>releasable</u> current in 0.1 A steps

IQ_L
IQ_H: actual current (IQ)
Conversion: I[A_{eff}] = IQ * AMPS / 10 / 4095
Bit 15 is the sign bit

ID_L
ID_H : actual current(ID)
Conversion: I[A_{eff}] = ID * AMPS / 10 / 4095
Bit 15 is the sign bit

MtrCur_Limit_L
MtrCur_Limit_H : in the data set <u>released</u> motor current
Conversion: IMot[A_{eff}] = MtrCur_Limit * AMPS / 10 / 4095
Bit 15 is the sign bit

RegCur_Limit_L
RegCur_Limit_H : in the data set <u>released</u> generator current
Conversion: IGen[A_{eff}] = RegCur_Limit * AMPS / 10 / 4095
Bit 15 is the sign bit

Signal_L
Signal_H: For the µs signal, the uppermost 6 bits must be masked to 0.
The top bit indicates if the signal is valid.

MaxRPM_L
MaxRPM_H : maximum releasable motor speed in U/min

RPM_L
RPM_H: actual speed
Conversion: N[U/min] = (RPM * MaxRPM) / 10922
Bit 15 is the sign bit (direction)

RPM_Limit_L
RPM_Limit_H : in the data set released speed
Conversion: N [U/min] =(RPM_Limit * MaxRPM) / 10922
Bit 15 is the sign bit

T_F (overtemperature fault):
Bit 7-0

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

The value in curly brackets can be found in the Windows Monitor under Parameter-> SLS!

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 (over- /undervoltage fault):
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	--

The value in curly brackets can be found in the Windows Monitor under Parameter-> SLS!

Bit 7 **SO_OV:** SwitchOff OverVolt (24V ECU: U > 25,66V
42V ECU: U > 43,11V
60V ECU: 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 ECU: U < 7,98V
42V ECU: U < 15,51V
60V ECU: U < 15,51V
to SN599 at the 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 (Error of the internal control):
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 **PL_F:** PhaseLoss_Flt -> stop+retry
Bit 6 **unimplemented:** read as '0'
Bit 5 **ZS_F:** ZeroSpd_Flt -> stop+retry
Bit 4 **I_F:** I_Offset_Flt -> stop+retry
Bit 3 **OS_F:** OvrSpd_Flt -> stop+retry
Bit 2 **LL_F:** Loadless_Fault -> stop+retry
Bit 1 **2PH_F:** 2-Phase-PWM
Bit 0 **FS:** Failsafe_STOP (SLS stopped) -> check -> clear Error

Tmax_DR (Temp Max derate Register)**:**
Umin_DR (Overvoltage derate Register)**:**
Umax_DR (Undervoltage derate Register)**:** These 3 registers give more detail on how far is down regulated.
Value ranges for the 3 Derate registers:
0x40 (it is not yet regulated) until
0x00 (controller switches off)

NACK: transmission error:

SLS sends to Host:

Byte 0	Byte 1	Byte 2	Byte 3
'?'	3_d	'?'	CRC

2.1. Servo signal override:

Status request:

Host sends to SLS:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
'!'	7_d	'S'	1_d	Active	Signal_L	Signal_H	CRC

To prevent a timeout, the TAG "Servo signal override" must be sent cyclically. The timeout is 300ms.

Active : 0xAA Signal will be overwritten.
0x00 Signal will not be overwritten.

Signal_L
Signal_H : Signal specification in µs permissible range:
800 .. 2200 µs

ACK: Status feedback:

SLS sends to Host see 2

2.2. Servosignal offset:

Status request:

Host sends to SLS:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
'!'	5_d	'S'	2_d	Servooffset	CRC

Servooffset : Servo signal offset in µsec
Value range: + / - 127 µsec
The sent offset becomes permanent stored in the controller.

ACK: Status feedback:

SLS sends to Host see 2

2.3. Option Control Panel (chargeable):

Status request:

Host sends to SLS:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
'!'	14_d	'S'	3_d	Active	Control	SPD_WMon_L	SPD_WMon_H

Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14
MtrCur_WMon_L	MtrCur_WMon_H	RegCur_WMon_L	RegCur_WMon_H	Accel_WMon	Decel_WMon	CRC

To prevent a timeout, the TAG "Servo signal override" must be sent cyclically. The timeout is 300ms.

Active : 0xAA Signal will be overwritten.
0x00 Signal will not be overwritten.

Control:
Bit 7-0

U-0	U-0	U-0	U-0	U-0	W	W	W
--	--	--	--	--	Direction	Start/Stop	parking brake_ active

Bit 7-3 **unimplemented:** read as '0'
Bit 2 Direction
Bit 1 Stop = 0; Start = 1
Bit 0 parking brake active = 1, the motor is braked by short circuit of the 3 phases.

SPD_WM_L
SPD_WM_H : speed specification, value range: 0 .. 1023
1023 corresponds to the engine speed released in the parameters **(RPM_Limit)**

MtrCur_WMon_L
MtrCur_WMon_H : motor current specification, value range: 0 .. 1023
1023 corresponds to the motor current released in the parameters **(MtrCur_Limit)**

RegCur_WMon_L
RegCur_WMon_H : generator current specification, value range: 0 .. 1023
1023 corresponds to the motor current released in the parameters **(RegCur_Limit)**

Accel_WMon: acceleration rate, value range: 1 .. 255
Decel_WMon: deceleration rate, value range: 1 .. 255

ACK: Status feedback:

SLS sends to Host see 2

3. Error reset/ SW-Reset

Error reset:

Host sends to SLS:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
'!'	4_d	'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:** complete software will be restarted (via reset vector)
Bit 6-5 **unimplemented:** write as '0'
Bit 4 **Clear:** all errors are cleared
Bit 3-0 **unimplemented:** write as '0'

ACK -> acknowledge:

SLS sends to Host:

Byte 0	Byte 1	Byte 2	Byte 3
'?'	3_d	'R'	CRC

NACK: transmission error:

SLS sends to Host:

Byte 0	Byte 1	Byte 2	Byte 3
'?'	3_d	'?'	CRC