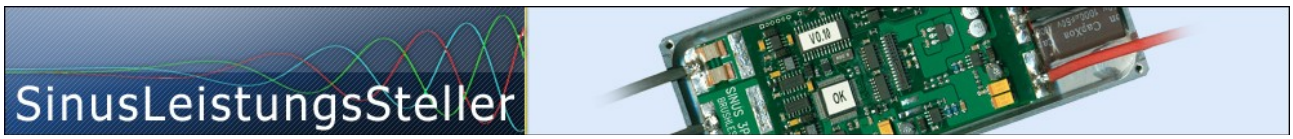


# Serial - Protocol - SLR

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

(based on Firmware V0.525, released 12. December 2019)

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

# Inhaltsverzeichnis

# 1. General Structure

| 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.
It is always a '!'-character due to initialize communication from the host to the SLR.
The returning packet, on the other hand, contains always the sync character '?'.

Byte 1: This byte (A) is implemented 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 consider this counter as a variable (not as constant) in your own host software!** Of course, further firmware version remains backward compatible, as described in detail below.

Byte 2: This byte (T) represents the actual **command (TAG)** to be processed by the SLR. For details see below. Returned packets return this byte unchanged at this position.

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 check-sum (CRC) is added, which consists of the sum of all bytes (excluding the check-sum itself).

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

# 2. Status request

Host sends to SLR:

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 |
|---|---|---|---|---|
| '!' | $4_d$ | 'S' | $7_d$ | CRC |

## 2.1. ACK, Status feedback

SLR sends to Host:

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|
| '?' | $34_d$ | 'S' | TP | TExt | T_F | U_F | C_F |

| Byte 8 | Byte 9 | Byte 10 | Byte 11 | Byte 12 | Byte 13 | Byte 14 | Byte 15 |
|---|---|---|---|---|---|---|---|
| Signal_L | Signal_H | $UBatt_{f4}$ | $UBatt_{f3}$ | $UBatt_{f2}$ | $UBatt_{f1}$ | $UZK_{f4}$ | $UZK_{f3}$ |

| Byte 16 | Byte 17 | Byte 18 | Byte 19 | Byte 20 | Byte 21 | Byte 22 | Byte 23 |
|---|---|---|---|---|---|---|---|
| $UZK_{f2}$ | $UZK_{f1}$ | $Idc_{f4}$ | $Idc_{f3}$ | $Idc_{f2}$ | $Idc_{f1}$ | $IQ_{f4}$ | $IQ_{f3}$ |

| Byte 24 | Byte 25 | Byte 26 | Byte 27 | Byte 28 | Byte 29 | Byte 30 | Byte 31 |
|---|---|---|---|---|---|---|---|
| $IQ_{f2}$ | $IQ_{f1}$ | $ID_{f4}$ | $ID_{f3}$ | $ID_{f2}$ | $ID_{f1}$ | $RPM_{f4}$ | $RPM_{f3}$ |

| Byte 32 | Byte 33 | Byte 34 |
|---|---|---|
| $RPM_{f2}$ | $RPM_{f1}$ | CRC |

**TP** (Temperature Power-Module):
There are three different types of temperature sensors:

**Type 1a** (KTY (R25=2k0 + 2k0)): conversion to °C:

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

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

**Type 1b** (KTY (R25=2k0 + 4k7)): conversion to °C:

$$T = -185{,}1 + \left(367 * \sqrt{\frac{954}{774 - TP} - 1}\right)$$

(0<=TP <=255 is mapped to -8,3°C<=T<= +150,1°C)

**Type 2** (NTC (R25=tbd + 4k7)): conversion to °C:

$$T = \frac{Beta}{\ln\left(\frac{TP * 4700}{(255 - TP) * R25}\right) + \left(\frac{Beta}{298}\right)} - 273$$

The Beta value and the resistance value R25 can be found and configured in the WMon under Extras / NTC_Config.

If the Beta value is set to 0 then this is a Type 1a sensor (KTY (R25=2k0 + 2k0)).
If the Beta value is set to 1 then this is a Type 1b sensor (KTY (R25=2k0 + 4k7)).

**TExt** (External Temperature - only at some SLR available): conversion to °C: see above (TP)

**T_F** (over-temperature 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 >Parameter > Limits!**

Bit 7 **SO_T:** SwitchOff OverTemp (TP > 100°C) =>Failsafe (SLR stopped)
Bit 6 **CMT:** CutOff MaxTemp (TP > 90°C, TExt > {T2_upper Lim})
Bit 5 **LMT:** Limit MaxTemp(TP > {T1_lower Lim}, TExt > {T2_lower Lim})
Bit 4-0 **unimplemented:** read as '0'

**U_F** (over- /under-voltage 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 >Parameter > Limits!**

Bit 7 **SO_OV:** SwitchOff OverVolt (mSLR-28-xx: Udc > 29V
mSLR-36-xx: Udc > 37V
mSLR-60-100: Udc > 61V
SLRi-60-xxx: Udc > 61V)
Bit 6 **CMV:** CutOff MaxVolt (U > {Udc_upper Lim})
Bit 5 **LMV:** Limit_MaxVolt (U > {Udc_lower Lim})
Bit 4 **unimplemented:** read as '0'
Bit 3 **SO_UV:** SwitchOff UnderVolt (mSLR-28-xx: Udc < 4,8V
mSLR-36-xx: Udc < 7,8V
mSLR-36-55: Udc < 9,7V
mSLR-60-100: Udc < 15V
SLRi-60-xxx: Udc < 15V)
Bit 2 **CUV:** CutOff_MinVolt (U <{Udc_lower Lim})
Bit 1 **LUV:** Limit_MinVolt (U < {Udc_upper 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 | HW_F | ZS_F | I_F | OS_F | LL_F | 2PH | FS |

Bit 7 **PL_F:** PhaseLoss_Flt -> stop+retry
Bit 6 **HW_F:** HW-Overcurrent_Flt-> stop+retry
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:** 2-Phasen-PWM
Bit 0 **FS:** Failsafe_STOP (SLR stopped) -> check -> clear Error

**Signal_L** (low-Byte)
**Signal_H** (high-Byte): For the µs signal, the uppermost 4 bits must be masked to 0.

**Ubatt:** The battery voltage as a 32 bit float number in [V]

**UZK:** The intermediate circuit voltage as a 32 bit float number in [V]

**Idc**: The current from the battery as a 32 bit float number in [A] (not available on every SLR)

**IQ**: The current IQ as a 32 bit float number in [Aac]

**ID**: The current ID as a 32 bit float number in [Aac]

**RPM_H**: The speed as a 32 bit float number in [rpm]

## 2.2. NACK, transmission error

SLR sends to Host:

| Byte 0 | Byte 1 | Byte 2 | Byte 3 |
|---|---|---|---|
| '?' | $3_d$ | '?' | CRC |

# 3. Servo signal overwrite

Host sends to SLR:

| 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 must be sent cyclic. 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

## 3.1. ACK, Status feedback

SLR sends to Host:

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| '?' | $5_d$ | 'S' | Signal_L | Signal_H | CRC |

## 3.2. NACK, transmission error

SLR sends to Host:

| Byte 0 | Byte 1 | Byte 2 | Byte 3 |
|---|---|---|---|
| '?' | $3_d$ | '?' | CRC |

# 4. Option Control Panel (chargeable)

Host sends to SLR:

| 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 must be sent cyclic. 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 brakes by short circuit of the 3 phases.

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

**MtrCur_WMon_L**
**MtrCur_WMonl_H :** motor current reference value scaled: 0 .. 1023
1023 corresponds to the motor curent released in the parameters **(MtrCur_Limit)**

**RegCur_WMon_L**
**RegCur_WMon_H :** generator current reference value, scaled: 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

## 4.1. ACK, Status feedback

SLR sends to Host:
(see 2. Status request)

## 4.2. NACK, Status feedback

SLR sends to Host:

| Byte 0 | Byte 1 | Byte 2 | Byte 3 |
|---|---|---|---|
| '?' | $3_d$ | '?' | CRC |

# 5. Error reset/ SW-reset

Host sends to SLR:

| 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'

## 5.1. ACK, acknowledge

SLR sends to Host:

| Byte 0 | Byte 1 | Byte 2 | Byte 3 |
|---|---|---|---|
| '?' | $3_d$ | 'R' | CRC |

## 5.2. NACK, transmission error

SLR sends to Host:

| Byte 0 | Byte 1 | Byte 2 | Byte 3 |
|---|---|---|---|
| '?' | $3_d$ | '?' | CRC |