



CAN-Bus - 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..... | 3 |
| 2 | Used Dataformats..... | 4 |
| 2.1 | Float32..... | 4 |
| 2.2 | Int32..... | 4 |
| 2.3 | Int16..... | 4 |
| 2.4 | Bits in Byte | 4 |
| 2.5 | Address..... | 4 |
| 3 | Command..... | 5 |
| 3.1 | Broadcast..... | 5 |
| 3.2 | ECU-Control..... | 6 |
| 3.3 | Signal reference value..... | 6 |
| 3.4 | Speed reference value..... | 6 |
| 3.5 | Current reference value..... | 7 |
| 3.6 | Ramps..... | 7 |
| 3.7 | Command with Address..... | 8 |
| 3.7.1 | Command with float32 as payload | 8 |
| 3.7.2 | Command with int32 as payload | 8 |
| 3.7.3 | Command with int16 as payload | 8 |
| 3.7.4 | Command with byte as payload | 8 |
| 3.8 | reserved..... | 8 |
| 4 | Feedback..... | 9 |
| 4.1 | Identifier..... | 9 |
| 4.2 | RPM and ServoSignal..... | 9 |
| 4.3 | Current..... | 9 |
| 4.4 | Voltage..... | 10 |
| 4.5 | Temperature..... | 10 |
| 4.6 | Faults and derate..... | 11 |
| 4.7 | Feedback with Address | 13 |
| 4.7.1 | Feedback with float32 as payload | 13 |
| 4.7.2 | Feedback with int32 as payload | 13 |
| 4.7.3 | Feedback with int16 as payload | 13 |
| 4.7.4 | Feedback with byte as payload | 13 |
| 4.8 | reserved..... | 13 |
| 5 | Table of used Addresses..... | 14 |

1 General Structure

All messages sent via CAN-Bus are using same frame-structure with 11-Bit-Identifier followed by up to 8 Bytes of payload-data.

| | | |
|-------------|--------|---------------------|
| ID: 11 Bits | | DATA: up to 8 Bytes |
| 4 Bits | 7 Bits | |
| MSG | NODE | |

ID: The ID consists of 4 bits for the MSG-number and 7 bits for the NODE-address.
MSG-numbers 0 to 7 are for **sending commands** to the SLR (see 2.1 - 2.8).
MSG-numbers 8 to 15 are for **receiving information** from the SLR (see 3.1 - 3.8).

DATA: 0 to 8 bytes of this frame are transferred.
Unused transmitted payload-data are random.

2 Used Dataformats

2.1 Float32

A float in IEEE 754 format is used.

| | | | | | | | |
|--------|----------|--------|----------|--------|--|--------|--|
| Byte 0 | | Byte 1 | | Byte 2 | | Byte 3 | |
| S | exponent | | mantissa | | | | |

S: Sign-bit

exponent: 8 bit exponent

mantissa: 23 bit mantissa

2.2 Int32

| | | | | | | | |
|-----------|--|----------|--|-----------|--|----------|--|
| Byte 0 | | Byte 1 | | Byte 2 | | Byte 3 | |
| High-Word | | | | Low-Word | | | |
| High-Byte | | Low-Byte | | High-Byte | | Low-Byte | |

2.3 Int16

| | | | |
|-----------|--|----------|--|
| Byte 0 | | Byte 1 | |
| High-Byte | | Low-Byte | |

2.4 Bits in Byte

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |

2.5 Address

| | | | |
|--------|-----------|-----------|--|
| Byte 0 | | Byte 1 | |
| R | Address_H | Address_L | |

R: Read/Write-Bit

Address: The address of the payload parameter, range: 0x0000 .. 0x7FFF

3 Command

3.1 Broadcast

A broadcast message will be received **simultaneously** by all nodes.

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0 | Node | -- | -- | -- | -- | -- | -- | -- | -- |

Broadcast with Node=0 is for everyone. Each SLR connected to the CAN-Bus sends a acknowledge. By this, a host can scan the CAN-Bus segment for active nodes.

To test if a single CAN-Bus node is still active, its node-address must be set. Only the addressed controller will send an acknowledge.

3.2 ECU-Control

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|---------|--------|--------|--------|--------|--------|--------|--------|
| 1 | Node | Control | -- | -- | -- | -- | -- | -- | -- |

Control:

Bit 7-0

| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
|-------|-------|-----|--------|-----|-----------|-------|------------------|
| start | reset | | source | | direction | bench | parking brake |

- Bit 7 **start:** Stop = 0; Start = 1
- Bit 6-5 **reset:** 00 no action
 01 **Clear:** all errors are cleared (no reset/reboot)
 10 **Reboot:** complete software will be restarted (via reset vector)
 11 reserved
- Bit 4-3 **source:** 00 no reference via CAN-Bus
 01 reference by ServoSignal (like ServoSimulation)
 10 reference by RPM/current (like ControlPanel)
 11 reserved
- Bit 2 **direction:** 0: CW
 1: CCW
- Bit 1 **bench:** 0: bench control disabled
 1: bench control enabled
- Bit 0 **parking brake:** the motor brakes by short circuit of the 3 phases

3.3 Signal reference value

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 2 | Node | Signal | | -- | -- | -- | -- | -- | -- |

Signal : Signal reference value [μ s], range: 800 .. 2200 μ s, int16

3.4 Speed reference value

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 3 | Node | Speed | | | -- | -- | -- | -- | -- |

Speed : speed reference value [rpm], float32

3.5 Current reference value

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 4 | Node | Imot | | | Igen | | | | |

Imot : motor current specification [Aac], float32

Igen : generator current specification [Aac], float32

3.6 Ramps

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 5 | Node | accel | decel | -- | -- | -- | -- | -- | -- |

accel: acceleration rate, value range: 1 .. 255, byte

decel: deceleration rate, value range: 1 .. 255, byte

3.7 Command with Address

R: If this bit is cleared, the payload is transmitted to the node.

If this bit is set, the node is commanded to send addressed parameter.

Address: The address of the payload parameter, range: 0x0000 .. 0x7FFF

Payload: There are different types of payload described in the next sections.

3.7.1 Command with **float32** as payload

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | |
|--------|--------|---------|--------|--------|---------|--------|--------|--------|--------|--|
| 6 | Node | Address | | | Payload | | | | | |

3.7.2 Command with **int32** as payload

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | |
|--------|--------|---------|--------|--------|---------|--------|--------|--------|--------|--|
| 6 | Node | Address | | | Payload | | | | | |

3.7.3 Command with **int16** as payload

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|---------|--------|---------|--------|--------|--------|--------|--------|
| 6 | Node | Address | | Payload | | -- | -- | -- | -- |

3.7.4 Command with **byte** as payload

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|---------|--------|---------|--------|--------|--------|--------|--------|
| 6 | Node | Address | | Payload | -- | -- | -- | -- | -- |

A table of the used addresses can be found in chapter 5 at the end of the document!

3.8 reserved

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 7 | Node | -- | -- | -- | -- | -- | -- | -- | -- |

free MSGBox for possible extensions

4 Feedback

4.1 Identifier

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|---------|----------|----------|--------|--------|--------|--------|--------|
| 8 | Node | Project | Hardware | Firmware | | SN | | -- | -- |

Project: project-ID (=5 for all SLR), byte
Hardware: hardware-ID, byte
Firmware: firmware-version (hexadecimal, eg. 0x0426 is V0.426), int16
SN: serial number of the SLR, int16

4.2 RPM and ServoSignal

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 9 | Node | RPM | | | | Signal | | -- | -- |

RPM: speed [rpm], float32

Signal_L, Signal_H: ServoSignal [μ s], the uppermost 4 bits must be masked (12-bit value), int16

4.3 Current

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 10 | Node | IQ | | | | ID | | | |

IQ: current IQ [Aac], float32

ID: current ID [Aac], float32

4.4 Voltage

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 11 | Node | Ubatt | | | | UZK | | | |

UBatt: battery voltage [V], float32

UZK: intermediate circuit voltage [V], float32

4.5 Temperature

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 12 | Node | TP | | TExt | | Idc | | | |

TP (Powermodule-Temperature): int16

There are two different types of temperature sensors:

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

$$T = -178,4 + (249 * \sqrt{\frac{3416}{4095 - TP} - 1})$$

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

$$T = -185,1 + (367 * \sqrt{\frac{3816}{4095 - TP} - 1})$$

Type 2 (NTC): Conversion to °C:

$$T = \frac{Beta}{\ln\left(\frac{TP * 4700}{(4095 - TP) * R25}\right) + \left(\frac{Beta}{298}\right)} - 273$$

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 Type1a sensor (KTY (R25=2k0 + 2k0)).

If the Beta value is set to 1 then this is a Type1b sensor (KTY (R25=2k0 + 4k7)).

TExt (external Temperature, only by some SLR): Conversion to °C: see TP, int16

Idc: current from the battery [Adc], float32
(not available on each SLR)

C_F (Error of the internal control):

Bit 7-0

| | | | | | | | |
|------|------|------|-----|------|------|-----|-----|
| R-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-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-Phase-PWM
Bit 0 **FS**: Failsafe_STOP (SLR stopped) -> check -> clear Error

4.7 Feedback with Address

R: Read/Write-Bit is reflected as it was sent by Command-Message (see 3.7)

Address: The address of the payload parameter, range: 0x0000 .. 0x7FFF

Payload: There are different types of payload described in the next sections.

4.7.1 Feedback with **float32** as payload

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|---------|--------|---------|--------|--------|--------|--------|--------|
| 14 | Node | Address | | Payload | | | | -- | -- |

4.7.2 Feedback with **int32** as payload

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|---------|--------|---------|--------|--------|--------|--------|--------|
| 14 | Node | Address | | Payload | | | | -- | -- |

4.7.3 Feedback with **int16** as payload

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|---------|--------|---------|--------|--------|--------|--------|--------|
| 14 | Node | Address | | Payload | | -- | -- | -- | -- |

4.7.4 Feedback with **byte** as payload

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|---------|--------|---------|--------|--------|--------|--------|--------|
| 14 | Node | Address | | Payload | -- | -- | -- | -- | -- |

A table of the used addresses can be found in chapter 5 at the end of the document!

4.8 reserved

| 4 Bits | 7 Bits | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 15 | Node | -- | -- | -- | -- | -- | -- | -- | -- |

free MSGbox for future extensions

5 Table of used Addresses

| Address | Payload | Format |
|---------|---|---------|
| 0x0200 | <p>MSGBox: to activate a MSGBox, the associated bit must be set.</p> <p>Bit0: The identifier MSGBox (Telegram-number: 8) is activated if this bit is set, otherwise it is deactivated.</p> <p>Bit1: The RPM and μs Signal MSGBox (Telegram-number: 9) is activated if this bit is set, otherwise it is deactivated.</p> <p>Bit2: The current MSGBox (Telegram-number: 10) is activated if this bit is set, otherwise it is deactivated.</p> <p>Bit3: The voltage MSGBox (Telegram-number: 11) is activated if this bit is set, otherwise it is deactivated.</p> <p>Bit4: The temperature MSGBox (Telegram-number: 12) is activated if this bit is set, otherwise it is deactivated.</p> <p>Bit5: The fault and derate MSGBox (Telegram-number: 13) is activated if this bit is set, otherwise it is deactivated.</p> <p>Bit6: The answer with sub ID MSGBox (Telegram-number: 14) is activated if this bit is set, otherwise it is deactivated.</p> <p>Bit7: This MSGBox (Telegram-number: 15) is not used yet.</p> | byte |
| 0x0201 | delay: Time gap to next transmission in steps of 1ms, range: 0 .. 32767 ms | int16 |
| 0x0202 | I_FieldWeak: Field weakening current specification [Aac], float32 | float32 |
| 0x0203 | NTC BETA onboard: BETA value of the internal NTC (if T1 was measured with NTC, if a KTY13 is installed, this value is 0) | int16 |
| 0x0204 | NTC2/ext BETA(25/80): BETA-value of the NTC for the temperature range 25/80 | int16 |
| 0x0205 | NTC2/ext R25: resistance of the NTC at 25 ° C | int16 |
| 0x02FF | Dummy32: only as sample | int32 |

At the moment only few addresses are used.

Future SLR-firmware-versions might map further parameters into the available address-range as needed.