M A N U A L

CAN - BUS
for digital motor controller

BAMOBIL-xx
BAMOCAR-xx
DS-xx / DPC-xx

UNITEK
Industrie Elektronik
G m b H

Hans-Paul-Kaysser-Straße 1
71397 Leutenbach-Nellmersbach

Tel: 07195 / 92 83 - 0
contact@unitek.eu
www.unitek.eu

Editin / Version

2024/ V2

Table of contens

1 Safety 2
1.1 Safety advices 2
1.2 Regulations and guidelines 2
2 General information 4
2.1 Logic functions 4
3 CAN BUS connections 5
3.1 Connections 5
4 Software 7
4.1 Format description 7
4.2 Head field 8
4.3 COB ID bits (CAN OBJECT ID) 8
4.4 RTR bit (REMOTE TRANSMISSION REQUEST) 8
4.5 DLC bits (DATA LENGTH CODE) 8
4.6 Data field 9
4.7 REGID 9
4.8 Data 9
5 Examples 10
5.1 Receiving CAN BUS data 10
5.2 Transmission of CAN data from the DSxx and BAxx servo to the CAN BUS 10
5.3 Sending from the master to the CAN bus to the DS servo 11
5.4 Transmission from the DS servo to the CAN BUS 14
6 Units 19
6.1 Conversion of the measuring units 19

1 Safety

1.1 Safety advices

Note:

This manual description is only to be used in connection with the hardware manual DS and the software manual NDrive!!



Before installation or commissioning begins, this manual must be thoroughly read and understood by the skilled technical staff involved. If any uncertainty arises, the manufacturer or dealer should be contacted

1.2 Regulations and guidelines

The devices and their associated components can only be installed and switched on where the local regulations and technical standards have been strictly adhered to.

EU Guidelines	2004/108/EG, 2006/95/EG, 2006/42/EG EN 60204-1, EN292, EN50178, EN60439-1, EN61800-3, ECE-R100 ISO 6469, ISO 26262, ISO 16750, ISO 20653, ISO12100
IEC/UL:	IEC 61508, IEC364, IEC664, UL508C, UL840
VDE Regulation/TÜV Regulation:	VDE100, VDE110, VDE160
Regulations of the statutory accident insurance and prevention institution:	VGB4

The user must ensure that in the event of:

- device failure
- incorrect operation
- loss of regulation or control



the axis will be safely de-activated.

It must also be ensured that the vehicles, machines, equipment, or vehicles are fitted with device independent monitoring and safety features.



Unearthed systems (e.g. vehicles) must be protected by means of independent insulation monitors.

There must be no danger to people or property!!!

Assembly
- should only be carried out when all voltages have been removed and the units are secured
- should only be carried out by suitably trained personnel

Installation
- should only be carried out when all voltages have been removed and the units are secured
- should only be carried out by suitably trained personnel for electrics
- should only be carried out in accordance with health and safety guidelines

Adjustments and programming
- should only be carried out by suitably trained personnel with knowledge in electronic drives and their software
- should only be carried out in accordance with the programming advice
- should only be carried out in accordance with safety guidelines
- should only be carried out if the path monitoring systems are active for limited travel distances.

CE
When mounting the units into vehicles, machines, and installations the proper operation of the units may not be started until it is ensured that the machine, the installation, or the vehicle comply with the regulations of the EC machinery directive 2006/42/EG, the EMC guideline 2004/108/EG, and the guideline ECE-R100.
On the described installation and test conditions (see chapter ‘CE notes’) it is adhered to the EC guideline 2004/108/EG including the EMC standards EN61000-2 and EN61000-4.
A manufacturer's declaration can be requested.
The manufacturer of the machine or installation is responsible for observing the threshold values demanded by the EMC laws.

QS
Test results are archived with the device serial number by the manufacturer for a period of 5 years.
The test protocols can be asked for.

2 General information

2.1 Logic functions

Originally the serial data bus system CAN (Controller Area Network) was developed for the automobile industry. Since then, the CAN-BUS is used for a wide range of applications in the plant and mechanical engineering. CAN is internationally standardized as ISO11898. CAN meets the particularly high safety requirements of highly available machines and medical equipment. High transmission rates and favorable connection costs are the advantages of the CAN-BUS.

During the CAN data transmission no stations are addressed but the content of a message is marked by a network-wide clear identifier. The identifier also determines the priority of the message.

A high system and configuration flexibility is achieved due to the content-related addressing. Thus, it is very easy to add further equipment to the network.

In all digital UNITEK devices the CAN-BUS interface is installed as **Slave.**
It is intended for being connected to a CAN-BUS master.
The interface is opto-decoupled.
The primary supply is effected internally via DC/DC converters.

The UNITEK CAN-BUS can transmit the following functions:

Examples from master (CNC/SPS/ to slave (DRIVE-DS) (receiving)

Logic functions	Command values	Parameters	
Enable	Torque command value	Control parameters	
Reference run	Speed command value	Settings	
Start, Stop	Position command value		
	Current limits		

Examples from slave (DRIVE-DS) to master (CNC/SPS) (sending, transmitting)

Logic functions	Actual values	Parameters	Signals
RUN	Actual torque value	Control parameter	State signal
ENABLE	Actual speed value	Settings	Error signal
POS	Actual position value		
Limit switch			

The addresses (REGID) are indicated in the parameter survey (see NDrive Manual),
e.g. speed command value (SPEED_CMD) = REGID 0x31 <value in hex>.

3 CAN BUS connections

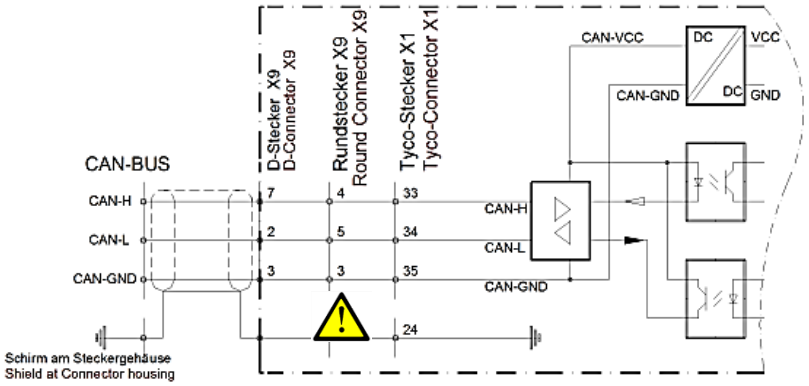
3.1 Connections

The CAN-BUS is the digital connection to the CNC control (CAN master).
The programming and operation is effected via the CAN-BUS by means of the control panel.
Interface acc. to ISO 11898-2.

Connection hardware:

Characteristic impedance	120 Ω
Conductor resistance (loop)	160 Ω/km
Operating capacity (800 Hz)	<60 nF/km

Input circuit



Connector pin assignment: see device manual
Cable colours (recommended)
LiYCY 4x0.25 + shield

CAN-GND	white
CAN-H	green
CAN-L	yellow

(Note: colors may different)

Fig. 1-1

CAN BUS isolated / CAN Gnd to common potential

CAN-BUS connection with several digital motor controllers DS- (slave) (example):
For other device types please observe the connector pin assignment (device manual)

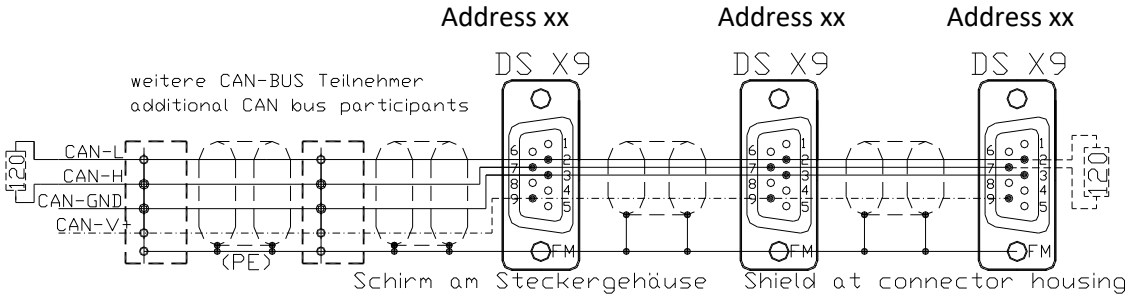


Fig. 4-2

Termination resistance

The line connection resistance (R = 120 Ω) must be installed across the first and the last BUS participants between CAN-H and CAN-L.

Power supply

The power supply of the CAN-BUS is internally provided via a DC/DC converter.

CAN BUS setting

The transmission addresses for receiving and sending and the transmission rate are entered via the parameter field ‘CAN-Setup’ of the pc program NDrive.

Address	Short symbol	Basic value (default)	Range
Receiving address (slave)	RPD01	0x201	0x201 to 0x27F
Transmission address (slave)	TPD01	0x181	0x181 to 0x1FF

Transmission rate NBT	Setting value BTR	Line length max.	
1000 kBaud	0x4002	20 m	
500 kBaud	0x4025	70 m	
625 kBaud	0x4014	70 m	
250 kBaud	0x405c	100 m	
100 kBaud	0x4425	500 m	

4 Software

4.1 Format description

The software format is designed for an optimal communication with the CNC machine controls and CAN modules of the Labod electronic company.
This format does not correspond to CANopen.
The transmission rate (Baud rate) is programmable.
The UNITEK standard is 500 kB/s (Labod 615 kB/s).
The devices UNITEK DSxx and BAxx can be added to a CANopen network (TPDO1, RPDO1) as slave.

Numerical format
Parameter value and parameter no. as Little-Endian format (Intel format)
Bit7 to 0 / Bit15 to 8 / Bit23 to 16 / Bit31 to 24

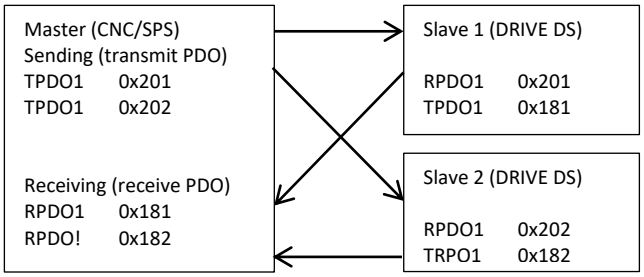
CAN format
The CAN protocol is a 3 or 5 Byte data package when received and 4 or 6 when send.
It is also possible to receive data packages of up to 8 Byte. In this case, however, it is evaluated as 5 Byte data package. The identifier is 11Bit wide. It comprises the **COB identifier**, the **RTR function** (Remote Transmission Request) and the **DLC information** (Data Length Code).
The Byte 0 of the data field is for the REGID index (parameter no.).
The second to the fifth Byte (Byte 1 to Byte 4) contains the data of the REGID index (parameter value).

Range	**Head**			**Data field**				
	COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function	11 Bit	0	Length	REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24

Master-Slave connection
In order to simplify the configuration a predefined Master/Slave connection set was specified in CANopen. For networks with one master and up to 127 slaves this assignment of COB identifiers offers each participant a simple solution for a CANopen network. Any information is solely distributed from the master. Direct communication among the slaves is not possible.

COB identifier										
Service					Node ID					
10			7		6					0
Example 0 x 181										
0	0	1	1	0	0	0	0	0	0	1
1			8				1			

The preferred objects (slave) are
TPDO1 (0x201 to 0x27F) and
RPDO1 (0x181 to 0x1FF).
The objects TPDO2..4 and RPDO2..4 can also be used.



Connection from master to slave

4.2 Head field

Range	Head			Data field				
	COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function	11 Bit	0	Length	REGID	B7 to 0	B15 to 8	B23 to 16	B31 to 24

4.3 COB ID bits (CAN OBJECT ID)

With CANopen the standard value (default) for TPDO1=0x181 and for RPDO1=0x201.

COB identifier											Object
Service				Node ID							
0	0	1	1	0	0	0	0	0	0	1	TPDO1
1			8				1				0x181-0x1FF
	1	0	0	0	0	0	0	0	0	1	RPD01
2			0				1				0x201-0x27F

The address can be changed by entering a direct transmission address in the digital motor controller (DSxx, BAxx) for receiving (CAN-ID-Rx 0x68) and for transmission field CAN-Setup in the NDrive.
The addresses of Tx-ID and Rx-ID can also be changed directly via the CAN (see example 1).

4.4 RTR bit (REMOTE TRANSMISSION REQUEST)

The value for RTR is always set to 0 / RTR is not used.

4.5 DLC bits (DATA LENGTH CODE)

The size of the data field is determined by the DLC bits.

Receiving:	value 0x03	corresponds to REGID plus 2 Byte (16 bit)
	value 0x05	corresponds to REGID plus 4 Byte (32 bit)
Transmission:	value 0x04	corresponds to REGID plus 2 Byte plus Dummy (16 bit)
	value 0x06	corresponds to REGID plus 4 Byte plus Dummy (32 bit)

4.6 Data field

The length of the data field for messages received in the digital motor controller is 3 or 5 Byte.
The upper data Bytes are registered when received, however, not taken into account.
The message for transmitting from the digital motor controller to the CAN-BUS is 4 or 6 Byte wide.

4.7 REGID

The first Byte is provided for the REGID index (parameter no.).
It is possible to determine up to 254 registers.
The most important parameter indexes are listed in the REGID list (see manual NDrive).

4.8 Data

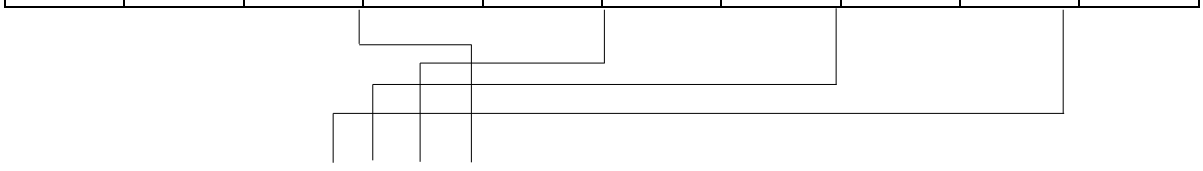
The data length is preset in the field 'DLC bits' (16 or 32 bits).
Byte 2 to Byte 5 are for the 32 bit register data (4 Byte).
Byte 2 to Byte 3 are for the 16 bit register data (2 Byte).

Example for the data field
Position command value for num 300010000

Function	**Hex value**
Transmission address for receiving	0x201
Data length 4 Byte	DLC=5
REGID for the position command value (POS_SOLL)	0x6E
Data length 4 Byte	DLC=5
Data for the position command value Num 300010000	0x11E1CA10

Data input

Byte 0		**Byte 1**		**Byte 2**		**Byte 3**		**Byte 4**	
REGID		**Data bits 7 to 0**		**Data bits 15 to 8**		**Data bits 23 to 16**		**Data bits 31 to 24**	
6	E	1	0	C	A	E	1	1	1



Data = 0x**11E1CA10** (corresponds to the num. position 300010000)

The input format is Little-Endian (Intel format)

Range	**Head**		**Data field**				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 2	**0x201**	**5**	**0x6E**	**0x10**	**0xCA**	**0xE1**	**0x11**

5 Examples

5.1 Receiving CAN BUS data

Transmission address at the DS servo	COB ID	(default = 0x201)
Data format	DLC	(3, 4, 5)
Parameter	Byte 0	(REGID – see parameter list)
Parameter content	Byte 1 to Byte 4	

Examples:

Changing the transmission address via CAN	see example 1
Disable the controller (no enable)	see example 2
Speed command value	see example 3
Position command value	see example 4
Torque command value	see example 5
Parameter value	see example 6
Write EEPROM	see example 7

5.2 Transmission of CAN data from the DSxx and BAxx servo to the CAN BUS

In general, the following is valid for the request to transmit from the DS servo:

Data field:	Byte 0 =	0x3D	Parameter transmission request
(DLC = 3)	Byte 1 =	REGID Value	Content of this REGID
	Byte 2 =	0x??	Time interval

1. Transmitting once: (see example 8)

Data field:	Byte 0 =	0x3D	Parameter transmission request
(DLC = 3)	Byte 1 =	0xA8	Content of this REGID
	Byte 2 =	0x00	Transmitting once

2. Cyclic transmission: (see example 9)

Data field:	Byte 0 =	0x3D	Parameter transmission request
(DLC = 3)	Byte 1 =	0xA8	Content of this REGID
	Byte 2 =	0x0A	Transmitting every 10ms (0 to 254ms)
Note:	*Byte 2 =*	*0xFF*	*Stop cyclic transmission*

3. Request for a status message after action: (see example 10)

Data field:	Byte 0 =	0x51	REGID for data after action
(DLC = 3)	Byte 1 =	0x10	Activation via bit 4
	Byte 2 =	0x00	*Don't care*

5.3 Sending from the master to the CAN bus to the DS servo

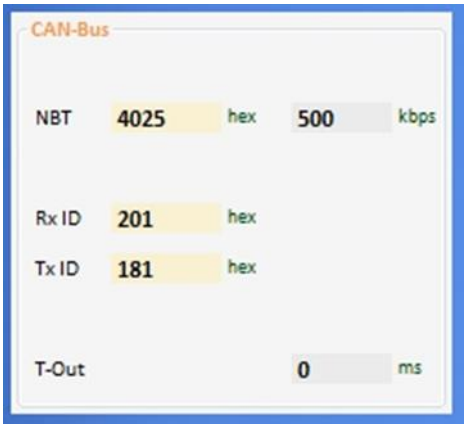
Example 1: Changing the transmission address via CAN

The address for receiving (slave) on a new DSxx, BAxx servo is 0x201 (default).
This address is to be changed in 0x210.
The REGID index for the receiving ID for the configuration of this address is 0x68 (FORE_CANIDREAD).

Function	**Hex value**
Transmission address to the Servo	0x201
Data length 2 Byte	DLC=3
REGID for CAN-Rx address	0x68
Value for a new CAN-Rx address	0x0210

Range	**Head**		**Data field**				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 1	0x201	3	0x68	0x10	0x02	---	---

Changing the transmission address in the pc program NDrive



Inputs:

NBT Transmission rate (kBaud)

Rx ID Receiving address in the DS (default 0x201)
Tx ID Transmission address from the DS (default 0x181)

T-Out Time error monitoring

Fig. 6-1

Example 2: Disable the controller (no enable) **Message to the servo**

Function	Hex value
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID for disable (MODE)	0x51
Value for the disable MODE BIT2	0x0004

Range	Head		Data field				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 2	**0x201**	**3**	**0x51**	**0x04**	**0x00**	---	---

Example 3: Speed command value **Message to the servo**

Function	Hex value
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID for the speed command value (SPEED_SOLL)	0x31
Value for 10% speed num. 3277	0x0CCD (100 % ≙ 32767)

Range	Head		Data field				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 3	**0x201**	**3**	**0x31**	**0xCD**	**0x0C**	---	---

Example 4: Position command value **Message to the servo**

Function	Hex value
Transmission address to the servo	0x201
Data length 4 Byte	DLC=6
REGID for the speed command value (POS_DEST)	0x6E
Value for position 3000000	0x2DC6C0

Range	Head		Data field				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 4	**0x201**	**5**	**0x6E**	**0xC0**	**0xC6**	**0x2D**	**0x00**

Example 5: Torque command value **Message to the servo**

Function	Hex value
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID for speed command value (TORQUE-CMD)	0x90
Value for 50% torque num 16380	0x3FFC

Range	Head		Data field				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 5	0x201	3	0x90	0xFC	0x3F	---	---

Example 6: Parameter configuration **Message to the servo**

Function	Hex value
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID for parameter acceleration (ACC ramp)	0x35
Data for 1000ms acceleration	0x03E8

Range	Head		Data field				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 6	0x201	3	0x35	0xCD	0x0C	---	---

Example 7: Writing EEPROM **Message to the servo**

Function	Hex value
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID to write EEPROM	0x84
EEPROM level 0 (EEPROM level1 = 0X0001)	0x0000

Range	Head		Data field				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 7	0x201	3	0x84	0x00	0x00	---	---

5.4 Transmission from the DS servo to the CAN BUS

All examples have the default transmission addresses (Rx ID=0x201 receiving, Tx ID=0x181 transmitting)

Example 8: Status message **One-time transmission from the servo**

In order to receive the information of a specified REGID a transmission request must be send to the servo. In the following example a one-time transmission of the REGID information is requested.

Message to the servo for a transmission request:

Function	**Hex value**
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID for reading data from the servo and transmission to the CAN (READ)	0x3D
REGID for status (KERN_STATUS)	0x40
Time interval (transmitting once)	0x00

Range	**Head**		**Data field**				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 8	**0x201**	**3**	**0x3D**	**0x40**	**0x00**	---	---

Retransmitted information from the servo:

Function	**Hex value**
Transmission address to the servo	0x181
Data length 2 Byte	DLC=4
REGID for status (KERN_STATUS)	0x40
Value of KERN_STATUS (0x40) are	0x0181

Range	**Head**		**Data field**				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 8	**0x181**	**4**	**0x40**	**0x81**	**0x01**	**0x****	

(Data range Byte 1 to Byte 4 in Little Endian format)

Current state of the status = 0x0181:

Bit0	Enable drive	(Ena)
Bit7	Position control	(P-N)
Bit8	Speed control	(N-I)

Example 9: Actual speed value **Cyclic transmission from the servo**

For the cyclic retransmission the register REGID_READ is programmed with a repeating time.
For the transmission repetition a cycle time (in ms) is entered in the Byte 2 in hex format (1-254ms).

Message to the servo for a transmission request:

Function	**Hex value**
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID for reading data from the servo and transmission to the CAN (READ)	0x3D
REGID for actual speed value (SPEED_IST)	0x30
For the repeating time 100ms the input in Byte 2 is	0x64
Note:	
The permanent transmission in Byte 2 can be stopped by	*0xFF*

Range	**Head**		**Data field**				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 9	**0x210**	**3**	**0x3D**	**0x30**	**0x64**	---	---

Information retransmitted from the servo within the interval of 100 ms

Function	**Hex value**
Transmission address to the servo	0x201
Data length 2 Byte	DLC=4
REGID for actual speed value (SPEED_IST)	0x30
Value of the speed command value 100% (num 32767)	0x7FFF

Range	**Head**		**Data field**				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 9	**0x190**	**4**	**0x30**	**0xFF**	**0x7F**	**0x****	---

Note:
It is possible to configure max. 8 state values which send their status cyclically.

Example 10: Status message after an event **Transmission from the servo (0x51 – Bit 4)**

Activation:
The automatic transmission is activated according to a specified configuration by setting of bit 4 in the REGID address 0x51. The device status signal (REGID 0x40) is automatically sent. In case of a modification of the device status the transmission takes place according to a configuration mask (bit mask (REGID 0x52)).

Configuration:
The configuration is active via the bit mask (REGID 0x52). The bit mask has a preset value of 0x0030. That is, in case the status bit 12 (Cal) or status bit 13 (Tol) is modified the complete status message (KERN_STATUS bit 0 to bit 15) is send to the CAN BUS.

Transmission request to the servo:

Function	**Hex value**
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID for data after an event (event trigger)	0x51
REGID for MODE BIT 4	0x10

Range	**Head**		**Data field**				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Beispiel 10	**0x201**	**3**	**0x51**	**0x10**	**0x00**	---	---

Information retransmitted from the servo:

In the example the target position of a positioning run is reached and bit 13 is set in the device status (Tol). Thus, the automatic transmission of the device status (REGID 0x40) is triggered.

Function	**Hex value**
Transmission address to the servo	0x181
Data length 4 Byte	DLC=6
REGID for status (KERN_STATUS)	0x40
Value of KERN_STATUS (0x40)	0x0181

Range	**Head**		**Data field**					
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24	b32 to 39
Example 10	**0x181**	**6**	**0x40**	**0x81**	**0x31**	**0x00**	**0x00**	**0x****

Current state of the status (KERN_STATUS) = 0x3181:

Bit0	Enable drive	(Ena)
Bit7	Position control	(P-S)
Bit8	Speed control	(S-I)
Bit12	Calibrated	(Cal)

Example 10-1: Status message after a selected event **Transmission from the servo**

The event trigger is changed to the assigned status bit via the configuration mask (REGID 0x52).

For example: Configuration mask (0x52) = 0x20 corresponds to continuous current (Icns)
Configuration mask (0x52) = 0x12 corresponds to limit switch + and – (Lim+, Lim-)

Determine the trigger event with the configuration mask (0x52).

Function	**Hex value**
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID for configuration mask	0x52
REGID for status trigger selection (e.g. limit switch)	0x12

Range	**Head**		**Data field**				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 10-1	**0x201**	**3**	**0x52**	**0x12**	**0x00**	---	---

Transmission of the status after a selected status event:

The set value for the configuration mask (0x52) is 0x0012.
When a limit switch is assigned (+ or -) the complete status message (4 Byte) is send.

Function	**Hex value**
Transmission address to the servo	0x201
Data length 2 Byte	DLC=3
REGID for data after an event (event trigger)	0x51
REGID for MODE BIT 4	0x10

Range	**Head**		**Data field**				
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24
Example 10-1	**0x201**	**3**	**0x51**	**0x10**	**0x00**	---	---

Information retransmitted from the servo:

Function	**Hex value**
Transmission address to the servo	0x181
Data length 4 Byte	DLC=6
REGID for status (KERN_STATUS)	0x40
Data for KERN_STATUS (0x40)	0x0181

Range	**Head**		**Data field**					
	COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Function			REGID	b7 to 0	b15 to 8	b23 to 16	b31 to 24	b32 to 39
Example 10-1	**0x181**	**6**	**0x40**	**0x85**	**0x31**	**0x00**	**0x00**	**0x****

Current state of the status (KERN_STATUS) = 0x3185

Bit 0	Enable drive	(Ena)
Bit 2 oder Bit 3	Limit switch assigned	(Lim+ oder Lim-)
Bit 7	Position control	(P-N)
Bit 8	Speed control	(N-I)
Bit 12	Calibrated	(Cal)

6 Units

6.1 Conversion of the measuring units

For position, speed, current, and command value:
The measured values are not converted in the device.
The numerical values (num) are displayed and processed.
These values are to be observed during the data transmission (CAN-BUS, RS232) as well as for the track and oscilloscope display.

Position

Actual position value range	Resolver	Incremental encoder
Pulses/rpm Max. value +/-2147483647 (31bit-1)	65536	65536
Resolution (smallest value)	16 (65536/4096 (12Bit)	65536/Inc x4
Example Spindle drive Slope 5mm/rpm	Travel 1000 mm = 200 rpm 200 rpm = 13107200 Resolution 65536/4096 = 16	Incremental encoder 2048 puls/rpm Travel 1000 mm = 200 rpm 200 rpm = 1638400 Resolution 65536/8192 = 8

Speed

Actual speed value range	Calibration speed (Nmax)	Limitation
Max. value +/- 32767 (15Bit-1)	N max value in the parameter field Motor and speed = 32767	Limitation in the parameter field Speed within the limit
Example	N max = 2000 The speed of 2000 rpm corresponds to 32767	Limit the speed to 1500 rpm Limit = 32767/2000*1500 = 24575 num The max. speed is limited to 1500 rpm

Current

Actual current value range	I 100%	Rated current calibration I-device			Peak current DC disabled		Limitation
Max. value +/- 9Bit	mV	Num	Aeff	A=	Num	A=	
DS 205/405	550	110	5	7	160	10	Limitation in the parameter field. Motor and current. The smaller value is effective.
DS 412	800	160	12	17	230	24	
DS 420	700	140	20	28	200	40	
							Limit I con eff to 2 A. I con = 110 / 5 * 2 = 44 num. The max. continuous current is limited to 2 A.
Example (DS205/4059)							

Command values

Position command value range	Speed command value range	Current command value range		
Max. value +/- 31Bit	**Max. value +/- 15Bit**	**Max. value +/- 9Bit**		
+/- 2147483647 num	+/- 32767 num	DS205/405	rated:110	max:160
		DS 412	rated:160	max:230
		DS 420	rated: 140	max:200

Note: The analog command value (AIN1, AIN2) 10 V corresponds to 29490 (90% of the max. speed)