Stögra Antriebstechnik GmbH

Machtlfinger Straße 24
D-81379 München

Tel.: (089)15904000
Fax.: (089)15904009

SERS 01, SERS 06 und SERS 12

Version V

CANopen

Stepping motor power amplifier board with position control and CANopen interface

CANopen specific additions to the 'SERS installation and programming manual' for SERS with RS232 interface

edition December 2002

We reserve the right to make engineering changes, refinements and improvements.
Mechanical and electrical ratings and dimensions are, therefore,
subject to change without notice.
no liability whatsoever is accepted.

Index

1. General notes

1.1 Short overview

- the stepper motor control SERS...CAN is a 1-axis position controls with CANopen interface to drive 2-phases-stepper motors.
- all parameters, the allowed writ able identifiers and their default values are included in the file "**stoegra.eds**".
- the common (e.g. communication) parameters with index numbers until 0x1000 are defined according to the "CiA Draft Standard 301" (Application Layer and Communication Profile) from the "CAN in Automation e. V."
- Additionally the standard parameters of the "CANopen Device Profile for Drives and Motion Control" in the "CiA Draft Standard Proposal DSP-402" from Index 0x6000 are implemented.
- The implemented functions of the SERS...CAN are identically to the position control SERS with RS232 interface (for description see the standard SERS installation and programming manual)
- The CANopen-interface on the SERS is galvanically isolated.

1.2 CANopen Implementation in the SERS

- the used CAN device "SAJ1000" is filtering at the CANbus all messages which are for the own address, for not creating a buffer overflow in case of a high data intensity on the bus. Broadcast messages (NMT-telegrams) are not filtered.
- the drive interrogates the receive register at least all 2ms, and evaluates the messages.
- the input buffer of the device stores up to 64 bytes.
- there are used 11 Bit identifier.
- the drive type is "0x00040192" according to "DSP402", chapter "7.2.1".
- in the "manufacturer status" (Index 0x1002) at the high word there can be read the status class 2 (SERS-warnings P12) and in the low word the status class 1 (SERS-error P11). Resetting the status class 1 (error) can be done via the "control word" (6040) or via the manufacturer specific part.
- the drive includes 1 transmit PDO and 3 receive PDOs
- the drive accepts only a Sync-object with the COB-Id 0x80. The high bit is not relevant.
- all values are accepted as "comm cycle period" (1006), but they will be not stored or evaluated - the value read is always 2000.
- the parameter "synchronous window length" will not be evaluated and read as 0.
- Node Guarding is implemented completely. A heart beat protocol is not implemented. The drive generates an emergency telegram at power on with the length 0 (old standard), and also a Node guard boot up (new standard).
- The identity object (1018) does not have a "Vendor ID", "Product Code" and serial number yet (edition January 2001).

- "modes_of_operation" (6060) : the drive supports 1 "Profile Position Mode" and 6 "Homing Mode".
- "position_actual_value" (6064) can not be written (in contrary to the CAN standard)
- "software_position_limit" (607d) : the limits are not checked when writing the position command value, but only during movement.
- "feed_constant" (6092) allows only the value 1 for "shaft_revolutions" (subindex 2).
- "digital_inputs" (60fd) :
  - Bit 0 is the limit switch minus
  - Bit 1 is the limit switch plus
  - Bit 2 is the home switch
  - Bit 16 until Bit 23 are the digital inputs I1 until I8
  - Bit 24 is the stop switch
- "digital_outputs" (60fe) : Bit 16 until Bit 19 are the 4 outputs O1 until O4 of the SERS
- All parameters of the "SERS" (in case they do not match the standard CAN profile) can be written/read also via the manufacturer specific index range (0x2000 until 0x5fff). The file "stoegra.eds" includes the assignments index numbers → SERS parameter numbers.

1.3 Modifications in comparison to the SERS installation and programming manual

In the following documentation the “SERS installation and programming manual“ (for SERS with RS232 interface) is named **SERS-manual**.

Following limitations and changes for the SERS…CAN version apply in comparison to the SERS with RS232 interface and the corresponding SERS-manual:

1. The baudrate can be adjusted via the 8-pole DIP-switch 1 (see page 11 and SERS-manual page 11 and page 15) with the bits 1 until 3 of the switch as follows:

Baudrate (KBaud)	20	50	100	125	250	500	800	1000
Bit 1	OFF	ON	OFF	ON	OFF	ON	OFF	ON
Bit 2	OFF	OFF	ON	ON	OFF	OFF	ON	ON
Bit 3	OFF	OFF	OFF	OFF	ON	ON	ON	ON

2. The connections of the 9-pole D-Sub-connector is as follows:
   Pin 2 : CAN_Low
   Pin 7 : CAN_High
   Pin 3 : GND common ground

3. 7-segment-display
   "1" The node is in "preoperational" state – indication after power on
   "2" CAN is initialised and the node is in "stopped" state
   "3" The node is in "operational" state
   "4" No CAN interface found on SERS (internal interface error)

   All other indications on the 7-segment-display are as described in the SERS manual. Error messages, warnings and the ON-state (“5“) overwrite the indications "1" until "4".

2. Assignment SERS parameter to the CANopen profile

2.1. Standard CANopen parameter

CAN standard		SERS parameter
Position actual value	6064	P51
Target_position	607a	P47
Software_position_limit	607d	
Min_position_limit		P1040
Max_position_limit		P1041
Profile_velocity	6081	P91 / V
Profile_acceleration	6083	P138 / A
Quick_stop_deceleration	6085	P1030
Motion_profile_type	6086	P1032
Gear_ratio	6091	
Motor_revolutions		P121
Shaft_revolutions		P122
Feed_constant	6092	P123
Homing_speeds	6099	
Speed_during_search_for_switch		P41
Speed_during_search_for_zero		P1003

2.2. Scaling

The scaling is defined in the manufacturer specific parameter as follows:

- scaling position data	: SDO Index 204c (SERS parameter P76)
- scaling velocity data	: SDO Index 202c (SERS parameter P44)
- scaling acceleration data	: SDO Index 20a0 (SERS parameter P160)

All position, velocity and acceleration values are scaled depending on these parameters above.

2.3. Positioning modes

The SERS..CAN drive supports mode 1 "Profile Position Mode" and 6 "Homing Mode". These modes are documented in the "CiA Draft Standard Proposal DSP-402", "CANopen Device Profile for Drives and Motion Control".

In homing mode there is possible following "homing_method" (Index 0x6098):

1: Homing on the Negative Limit Switch
2: Homing on the Positive Limit Switch
3: Homing on the Positive Home Switch
5: Homing on the Negative Home Switch
17: Homing on the Negative Limit Switch, without Index (electrical zero phase)
18: Homing on the Positive Limit Switch, without Index
19: Homing on the Positive Home Switch, without Index
20: Homing on the Negative Home Switch, without Index

2.4. PDO data content and transmission types

The SERS...CAN drive supports 3 receive and 1 transmit PDO.

The PDOs are evaluated 1 time each 2ms cycle.

29 bit identifier are not allowed.

The COB-IDs of the PDO may receive only standard values:

1. Receive PDO: 0x200+NodeID
2. Receive PDO: 0x300+NodeID
3. Receive PDO: 0x400+NodeID

Transmit PDO: 0x180+NodeID

The transmit PDO may be set not valid (high bit of COB-ID is 1).

At the receive PDO the transmission type will be saved but not evaluated.

At the transmit PDO all transmission types are implemented except "REMOTE_UPDATE_SYNC" (252).

At the transmission types "ACYCLIC_SYNC" (0), "MANUFACTUROR_SPECIFIC" (254) and "DEVICE_PROFILE_DEFINED" (255) the PDO will be send in case of a data change.

Synchronous PDO are sent according to the set Sync-numbers, but only with an accuracy of 2ms.

Remote requests of the PDO is possible at the transmission Type "REMOTE" (253).

Inhibit times are evaluated.

The mapping of the PDO is fixed, and can be read only. Following adjustments are made:

1. Receive PDO: control word (6040), 16 Bit
2. Receive PDO: control word (6040), modes_of_operation (6060), 8 Bit
3. Receive PDO: control word (6040), target_position (607a), 32 Bit

Transmit PDO:
Status word (6041), 16 Bit
position_actual_value (6064), 32 Bit
ENDSCHALTER_DIGIN (2420), 16 Bit, the limit switches are the high byte and Digin (digital inputs I1 until I8 of the SERS) are the low byte

2.5. NMT command-specifier

These telegrams ma be sent to one node or to all nodes simultaneously, to change the node status.

START_NODE	1
STOP_NODE	2
ENTER_PREOPERATIONAL	128
RESET_NODE	129
RESET_COMM	130

2.6. Node status

This status will be returned by the SERS…CAN drive with each requested node guarding telegram. The high bit is toggled.

BOOTUP	0
STOPPED	4
OPERATIONAL	5
PREOPERATIONAL	127

2.7. Abort Codes

In case a SDO transfer was terminated not correctly, then the reason for this transfer interruption can be read in the "abort code".

TOGGLE_BIT_NOT_ALTERED	0x05030000
SDO_PROTOCOL_TIMED_OUT	0x05040000
COMMAND_SPECIFIER_UNKNOWN	0x05040001
INVALID_BLOCK_SIZE	0x05040002
INVALID_SEQUENCE_NUMBER	0x05040003
CRC_ERROR	0x05040004
OUT_OF_MEMORY	0x05040005
UNSUPPORTED_ACCESS_TO_OBJECT	0x06010000
ATTEMPT_TO_READ_A_WRITE_ONLY	0x06010001
ATTEMPT_TO_WRITE_A_READ_ONLY	0x06010002
OBJECT_DOES_NOT_EXIST	0x06020000
OBJECT_CANNOT_BE_MAPPED_TO_PDO	0x06040041
PDO_LENGTH_EXCEEDED	0x06040042
PARAMETER_INCOMPATIBILITY	0x06040043
INTERNAL_INCOMPATIBILITY	0x06040047
ACCESS_HARDWARE_ERROR	0x06060000
DATA_TYPE_LENGTH_WRONG	0x06070010
DATA_TYPE_LENGTH_TOO_HIGH	0x06070012
DATA_TYPE_LENGTH_TOO_LOW	0x06070013
SUBINDEX_DOES_NOT_EXIST	0x06090011
PARAMETER_VALUE_RANGE_EXCEEDED	0x06090030
PARAMETER_VALUE_TOO_HIGH	0x06090031
PARAMETER_VALUE_TOO_LOW	0x06090032
MAX_IS_LESS_THAN_MIN	0x06090036
GENERAL_ERROR	0x08000000
DATA_NOT_STORED_OR_TRANSFERED	0x08000020
DATA_NOT_STORED_LOCAL_CONTROL	0x08000021
DATA_NOT_STORED_WRONG_STATE	0x08000022
OBJECT_DICT_CREATION_FAIL	0x08000023

2.8. Emergency data content

Emergency telegrams are sent by the SERS…CAN drive without having been requested in case of any change of the error situation. That the telegram will be sent also in case the drive turns to a status of no error.
The first 2 bytes content the "Error Code". The next Byte is the “Error Register“, which can be read also via Index 1001.
Then the customer specific contents follow: 1 byte "errno", 2 bytes "state class 1" (SERS-errors = SERS parameter P11) and 2 bytes "state class 2" (SERS-warnings = parameter P12).

2.8.1. Emergency Error Code (further codes in DSP402)

NULL	0x0000	
GENERIC	0x1000	
CURRENT	0x2000	
CURRENT_INPUT	0x2100	//Current, device input side
SHORT_CIRCUIT	0x2110	//Current, device input side
CURRENT_DEVICE	0x2200	//Current inside the device
CURRENT_OUTPUT	0x2300	//Current, device output side
VOLTAGE	0x3000	
MAINS_VOLTAGE	0x3100	
VOLTAGE_INSIDE	0x3200	//Voltage inside the device
OUTPUT_VOLTAGE	0x3300	
TEMPERATURE	0x4000	
AMBIENT_TEMPERATURE	0x4100	
DEVICE_TEMPERATURE	0x4200	
DEVICE_HARDWARE	0x5000	
DEVICE_SOFTWARE	0x6000	
INTERNAL_SOFTWARE	0x6100	
USER_SOFTWARE	0x6200	
DATA_SET	0x6300	
ADDITIONAL_MODULES	0x7000	
MONITORING	0x8000	
COMMUNICATION	0x8100	
CAN_OVERRUN	0x8110	//CAN Overrun (Objects lost)
CAN_PASSIVE	0x8120	//CAN in Error Passive Mode
LIFE_GUARD	0x8130	//Life Guard Error or Heartbeat Error
RECOVERED_FROM_BUS_OFF	0x8140	
PROTOCOL	0x8200	
PDO_NOT_PROCESSED	0x8210	//PDO not processed due to length error
PDO_LENGTH_EXCEEDED	0x8220	
EXTERNAL	0x9000	
ADDITIONAL_FUNCTIONS	0xf000	
DEVICE_SPECIFIC	0xff00	

2.8.2. Error Register

GENERIC	0x01	//general error
CURRENT	0x02	//short circuit power amplifier or motor
VOLTAGE	0x04	//error under voltage
TEMPERATURE	0x08	//error over temperature
COMMUNICATION	0x10	//communication error (overrun, error state)
PROFILE_SPECIFIC	0x20	//device profile specific
RESERVED	0x40	//reserved (always 0)
MANUFACTOR_SPEC	0x80	

2.8.3. Error numbers "errno"

1: to big
2: to small
3: not valid
4: invalid output
5: EEPROM storage full
6: EEPROM acknowledge timeout
7: EEPROM no acknowledge
8: EEPROM no page begin
9: run decimal constant to small
10: decimal constant to big
11: unknown if event
12: admission refused
13: Parameter not existing
14: adc expected
15: end of text expected
16: input text only in pgm mode
17: text to long
18: [decimal constant pgm psave] expected
19: * permitted only at P1 or z
20: data or z expected
21: command expected
22: programming mode not active
23: if expected
24: if event expected
25: goto or gosub or GT or GS expected
26: goto or gosub expected
27: goto expected
28: goto decimal constant expected
29: gosub expected
30: gosub decimal constant expected
31: [return RT run rs rf] expected
32: return expected
33: [decimal constant list ls lf] expected
34: = or ? expected
35: [decimal constant on off] expected
36: decimal constant or n expected
37: decimal constant expected

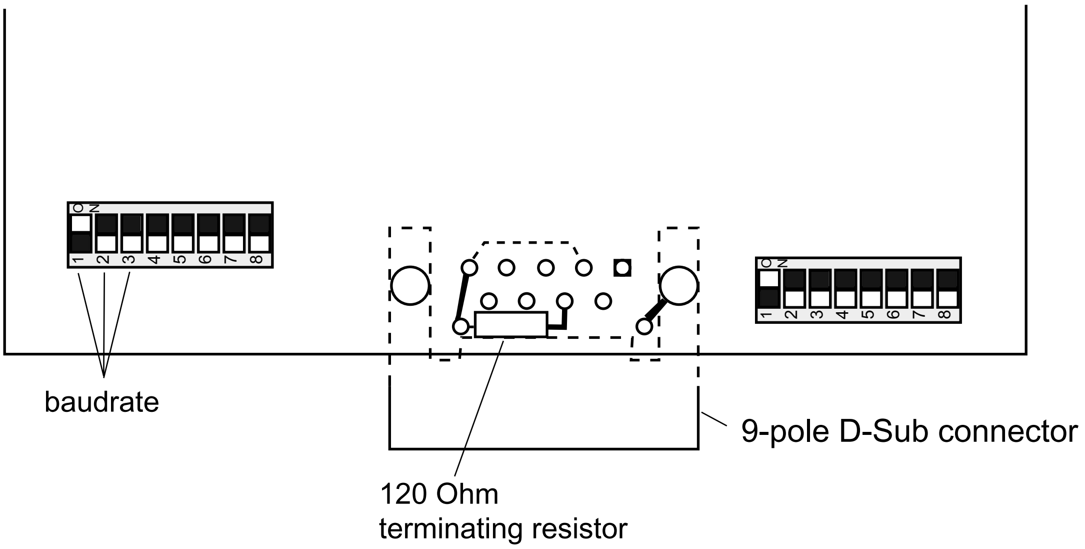
38: run expected
39: [new, neg, not] expected
40: new or neg expected
41: list expected
42: quit expected
43: off expected
44: program still running
45: pgm expected
46: Programming mode not active
47: del expected
48: data expected
49: change: only constant allowed
50: decimal constant or pos expected
51: pos expected
52: psave expected
53: [tr tron troff] expected
54: program not running
55: troff expected
56: ver expected
57: 1, 2, 3 or 4 expected
58: pos or possave expected
59: lp expected
60: possave expected
61: data or parameter expected
62: semicolon not allowed
63: not expected
64: not known status
65: program start not possible when service switch is ON
66: programmstart not possible, error P11
67: stop switch is active
68: stop switch is open
69: not valid program existing
70: drive must stand still
71: unknown destination
72: destination not valid
73: Stack overflow
74: unknown Opcode, forgot return ?
75: unknown fxxx Opcode
76: invalid Opcode for destination address
77: unknown f7xx Opcode
78: limit switch open
79: drive not enabled (OFF)
80: unknown positioning mode
81: drive must run constant or stand still
82: invalid EEPROM mode value
83: label already defined: L
84: position difference to big
85: new position to big
86: new position to small
87: new residual position to short
88: compare position 1 is to big

89: compare Position 1 is to small
90: compare Position 2 is to big
91: compare Position 2 is to small
92: new modulo value is to big
93: not writable, during drive is positioning
94: solder bridge is set wrong for this range
95: negative software limit position is bigger than positive position
96: exponent to big
97: exponent to small
98: calculation result is to big
99: calculation result is to small
100: calculation result is to big to be displayed
101: calculation result is to small to be displayed
102: division through 0
103: bus stopbit is active
104: subindex not existing
105: value can not be written
106: value can not be read
107: Polynom with backlash not allowed
108: Missing Polynom data for next section
109: Wait expected
110: <=0 expected
111: =0 expected
112: 0 expected
113: 0 or 1 expected
114: > expected
115: 1 expected
116: = expected
117: 3 expected
118: Polynom end to short for deceleration (false definition of polynom section)
119: A to small
120: A to big
121: V to small
122: V to big

3. Terminating resistor 120 Ohm

In case of more participants at the CANopen bus only the last participant may have a 120 Ohm terminating resistor between the connecting pins CAN_Low (9-pol. D-SUB pin 2) and CAN_High (9-pol. D-Sub pin 3).

On the SERS the terminating resistor can be activated via a jumper – see figure below.



Jumper set: 120 Ohm terminating resistor in CANopen Bus
Jumper not set: no terminating resistor