



## **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**

Schrittmotorleistungsverstärkerkarte mit  
Positioniersteuerung und CANopen Schnittstelle

### **CANopen spezifische Ergänzungen zum Handbuch zur Inbetriebnahme und Programmierung der SERS mit RS232 Schnittstelle**

Ausgabe Dezember 2002

Änderungen, die der Verbesserung dienen, bleiben vorbehalten.

Bei der Erstellung von Texten und Bildern wurde mit höchster Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Für fehlerhafte Angaben und deren Folgen können wir keine Haftung übernehmen.

## Inhaltsverzeichnis

	Seite
1. Allgemeine Hinweise	3
1.1 Kurzübersicht	3
1.2 CANopen Implementation der SERS	3
1.3 Änderungen im Vergleich zum SERS-Handbuch	4
2. Zuordnung SERS Parameter zum CANopen Profil	5
2.1 Genormte CANopen Parameter	5
2.2 Wichtungen	5
2.3 Positionier Modi	5
2.4 PDO Dateninhalt und Transmission Types	6
2.5 NMT Command-Specifier	6
2.6 Knotenstatus	7
2.7 Abort Codes	7
2.8 Emergency Dateninhalt	8
2.8.1 Emergency Error Code	8
2.8.2 Error Register	9
2.8.3 Fehlernummern "errno"	9
3. Abschlußwiderstand 120 Ohm	12

# 1. Allgemeine Hinweise

## 1.1 Kurzübersicht

- Die Schrittmotor-Ansteuereinheit SERS...CAN ist eine 1-Achsen Positioniersteuerung mit CANopen Schnittstelle zum Ansteuern von 2-Phasen-Schrittmotoren.
- Alle Parameter, die erlaubten Zugriffe und deren Defaulteinstellungen sind im File "**stoegra.eds**" beschrieben.
- Die allgemeinen (z.B. Kommunikations-) Parameter mit den Indexnummern bis 0x1000 sind entsprechend "CiA Draft Standard 301" (Application Layer and Communication Profile) vom "CAN in Automation e. V." definiert.
- Zusätzlich sind die im "CiA Draft Standard Proposal DSP-402" definierten genormten Parameter des "CANopen Device Profile for Drives and Motion Control" ab Index 0x6000 implementiert.
- Die implementierten Funktionen der SERS...CAN sind identisch mit der Positioniersteuerung SERS mit RS232 Schnittstelle (beschrieben im SERS Handbuch zur Inbetriebnahme und Programmierung).
- Die CANopen-Schnittstelle auf der SERS ist galvanisch getrennt.

## 1.2 CANopen Implementation der SERS

- Der verwendete CAN Baustein "SAJ1000" filtert aus allen Meldungen des CANbus die für die eigene Adresse bestimmten aus, damit eine hohe Buslast nicht zu einem Pufferueberlauf fuehrt. Broadcast Meldungen (NMT-Telegramme) werden nicht gefiltert.
- Der Antrieb fragt das Empfangsregister mindestens alle 2ms ab, und wertet die Meldungen aus.
- Der Eingangspuffer des Bausteines speichert bis zu 64 Bytes.
- Es werden 11 Bit Identifier verwendet.
- Der Antrieb hat gemäß "DSP402", Kapitel "7.2.1" den Typ "0x00040192".
- Im "manufactor status" (Index 0x1002) sind im Highword die Zustandsklasse 2 (SERS-Warnungen) und im Lowword die Zustandsklasse 1 (SERS-Fehler) lesbar. Gelöscht wird die Zustandsklasse 1 über das "controlword" (6040) oder über den herstellerspezifischen Teil.
- Der Antrieb besitzt 1 Sende- und 3 Empfangs-PDO. s.u.
- Der Antrieb akzeptiert nur ein Syncobjekt mit der COB-Id 0x80. Das Highbit ist nicht relevant.
- Als "comm cycle period" (1006) werden alle Werte akzeptiert, aber nicht gespeichert oder ausgewertet. Gelesen wird immer der Wert 2000.
- Der Parameter "synchronous window length" wird nicht ausgewertet und als 0 gelesen.
- Node Guarding ist im vollem Funktionsumfang implementiert. Ein Heartbeat Protokoll ist nicht implementiert. Der Antrieb generiert beim Einschalten sowohl ein Emergency Telegramm der Länge 0 (alte Norm), als auch ein Nodeguard Bootup (neu).

- Für das Identity Objekt (1018) wurden noch keine "Vendor ID", "Product Code" und Seriennummer vergeben (Stand Januar 2001).
- "modes\_of\_operation" (6060) : Der Antrieb unterstützt 1 "Profile Position Mode" und 6 "Homing Mode".
- "position\_actual\_value" (6064) ist entgegen der Norm auch schreibbar.
- "software\_position\_limit" (607d) : Die Grenzen werden noch nicht bei der Wegvorgabe, sondern erst während dem Fahren kontrolliert.
- "feed\_constant" (6092) erlaubt fuer "shaft\_revolutions" (Subindex 2) nur den Wert 1.
- "digital\_inputs" (60fd) :
  - Bit 0 ist der Endschalter Minus
  - Bit 1 ist der Endschalter Plus
  - Bit 2 ist der Referenzschalter
  - Bit 16 bis Bit 23 sind die digitalen Eingänge I1 bis I8
  - Bit 24 ist der Stop Schalter
- "digital\_outputs" (60fe) : Bit 16 bis Bit 19 sind die 4 Ausgänge O1 bis O4
- Alle Parameter der "SERS" sind, falls sie nicht in das genormte Profil passen, auch über den herstellerepezifischen Indexbereich (0x2000 bis 0x5fff) erreichbar. Die Zuordnung ist im "stoegra.eds" File dokumentiert.

### 1.3 Änderungen im Vergleich zum SERS-Handbuch

Im Folgenden wird das "Handbuch zur Inbetriebnahme und Programmierung der SERS mit RS232 Schnittstelle" als **SERS-Handbuch** bezeichnet.

Folgende Einschränkungen bzw. Änderungen im Vergleich zur SERS mit RS232 Schnittstelle und dem entsprechendem SERS-Handbuch gelten:

1. Die Baudrate am 8-poligen DIP-Schalter 1 (siehe Seite 11 und SERS-Handbuch Seite 11 und Seite 14) kann mit den Schalterbits 1 bis 3 folgendermaßen eingestellt werden:

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. Die Belegung des 9-poligen D-Sub-Steckers ist folgendermaßen:

Pin 2 : CAN\_Low

Pin 7 : CAN\_High

Pin 3 : GND Gemeinsamer Rückleiter

3. 7-Segmentanzeige

"1" Knoten ist im Zustand "preoperational" - Anzeige nach dem Einschalten

"2" CAN ist initialisiert und der Knoten im Zustand "stopped"

"3" Der Knoten ist im Zustand "operational"

"4" kein CAN vorhanden (interner Fehler)

Alle anderen Anzeigen sind wie im SERS Handbuch beschrieben.

Fehlermeldungen und Warnungen überschreiben die Anzeigen "1" bis "4".

## 2. Zuordnung SERS Parameter zum CANopen Profil

### 2.1. Genormte CANopen Parameter

Genormt		entspricht herstellerspezifisch
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. Wichtungen

Die Wichtungen werden in folgenden herstellerspezifischen Parametern definiert:

- Wichtung Lagedaten : SDO Index 204c (SERS Parameter P76)
- Wichtung Geschwindigkeitsdaten : SDO Index 202c (SERS Parameter P44)
- Wichtung Beschleunigungsdaten : SDO Index 20a0 (SERS Parameter P160)

Alle Lage-, Geschwindigkeits- und Beschleunigungswerte sind abhängig von diesen Parametern gewichtet.

### 2.3. Positionier Modi

Der Antrieb unterstützt 1 "Profile Position Mode" und 6 "Homing Mode".

Diese sind im "CiA Draft Standard Proposal DSP-402", "CANopen Device Profile for Drives and Motion Control" beschrieben.

Im Homing Modus sind folgende "homing\_method" (Index 0x6098) möglich:

- 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, ohne Index (Nullphase)
- 18: Homing on the Positive Limit Switch, ohne Index
- 19: Homing on the Positive Home Switch, ohne Index
- 20: Homing on the Negative Home Switch, ohne Index

## 2.4. PDO Dateninhalt und Transmission Types

Der Antrieb unterstützt 3 Empfangs- und 1 Sende-PDO.

Die PDOs werden 1 Mal pro 2ms Zyklus ausgewertet.

### 29 Bit Identifier sind nicht erlaubt.

Die COB-IDs der PDO können nur die Standardwerte annehmen:

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

Transmit PDO: 0x180+NodeID

Das Transmit PDO kann ungültig gesetzt werden (Highbit der COB-ID ist 1).

Bei den Receive PDO wird die Transmission Type gespeichert, aber nicht ausgewertet.

Bei dem Transmit PDO sind alle Transmission Types außer "REMOTE\_UPDATE\_SYNC" (252) implementiert.

Bei den Transmission Types "ACYCLIC\_SYNC" (0), "MANUFACTUROR\_SPECIFIC" (254) und "DEVICE\_PROFILE\_DEFINED" (255) wird das PDO jeweils bei einer Datenveränderung gesendet.

Synchrone PDO werden entsprechend der eingestellten Sync-Anzahl gesendet, aber nur mit einer Genauigkeit von 2ms.

Remote Anforderung des PDO ist bei der Transmission Type "REMOTE" (253) möglich.

Inhibitzeiten werden ausgewertet.

Das Mapping der PDO ist festgelegt, und kann nur ausgelesen werden. Folgende Einstellungen sind getroffen:

1. Receive PDO: controlword (6040), 16 Bit
2. Receive PDO: controlword (6040), modes\_of\_operation (6060), 8 Bit
3. Receive PDO: controlword (6040), target\_position (607a), 32 Bit

Transmit PDO:

statusword (6041), 16 Bit

position\_actual\_value (6064), 32 Bit

ENDSCHALTER\_DIGIN (2420), 16 Bit, die Endschalter bilden das Highbyte und Digin (Digitale Eingänge I1 bis I8) bildet das Lowbyte

## 2.5. NMT Command-Specifier

Diese Telegramme können an einen oder alle Knoten gleichzeitig gesendet werden, um den Zustand zu verändern.

START_NODE	1
STOP_NODE	2
ENTER_PREOPERATIONAL	128
RESET_NODE	129
RESET_COMM	130

## 2.6. Knotenstatus

Dieser Status wird mit jedem angeforderten Nodeguarding Telegramm vom Antrieb zurückgesendet. Das Highbit wird getoggelt.

BOOTUP	0
STOPPED	4
OPERATIONAL	5
PREOPERATIONAL	127

## 2.7. Abort Codes

Wenn ein SDO Transfer nicht ordnungsgemäß beendet wird, so steht im "abort code" der Grund des Abbruchs.

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 Dateninhalt

Emergency Telegramme werden bei Änderung einer Fehlersituation unangefordert vom Antrieb gesendet, also auch, wenn der Antrieb fehlerfrei wird.

Die ersten 2 Bytes enthalten den "Error Code". Das nächste Byte ist das "Error Register", das auch auf Index 1001 abgefragt werden kann.

Dann folgen die herstellerspezifischen Inhalte: 1 Byte "errno", 2 Byte "Zustandsklasse 1" (SERS-Fehler) und 2 Bytes "Zustandsklasse 2" (SERS-Warnungen).

### 2.8.1. Emergency Error Code (weitere 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	//allgemeiner Fehler
CURRENT	0x02	//Kurzschluß
VOLTAGE	0x04	//Unterspannungs Fehler
TEMPERATURE	0x08	//Übertemperatur
COMMUNICATION	0x10	//communication error (overrun, error state)
PROFILE_SPECIFIC	0x20	//device profile specific
RESERVED	0x40	//Reserved (always 0)
MANUFACTOR_SPEC	0x80	

## 2.8.3. Fehlernummern "errno"

- 1: zu groß
- 2: zu klein
- 3: ungültig
- 4: ungültiger Ausgang
- 5: EEPROM Speicher voll
- 6: EEPROM acknowledge timeout
- 7: EEPROM no acknowledge
- 8: EEPROM no page begin
- 9: run Dezimalkonstante zu klein
- 10: Dezimalkonstante zu groß
- 11: unbekanntes if Ereignis
- 12: Zugriff verweigert
- 13: Parameter nicht vorhanden
- 14: adc erwartet
- 15: Textende erwartet
- 16: Texteingabe nur im pgm Modus
- 17: Text zu lang
- 18: [Dezimalkonstante pgm psave] erwartet
- 19: \* nur bei P1 oder z erlaubt
- 20: Datum oder z erwartet
- 21: Anweisung erwartet
- 22: Programmiermodus nicht aktiv
- 23: if erwartet
- 24: if Ereignis erwartet
- 25: goto oder gosub oder GT oder GS erwartet
- 26: goto oder gosub erwartet
- 27: goto erwartet
- 28: goto Dezimalkonstante erwartet
- 29: gosub erwartet
- 30: gosub Dezimalkonstante erwartet
- 31: [return RT run rs rf] erwartet
- 32: return erwartet
- 33: [Dezimalkonstante list ls lf] erwartet
- 34: = oder ? erwartet
- 35: [Dezimalkonstante on off] erwartet
- 36: Dezimalkonstante oder n erwartet

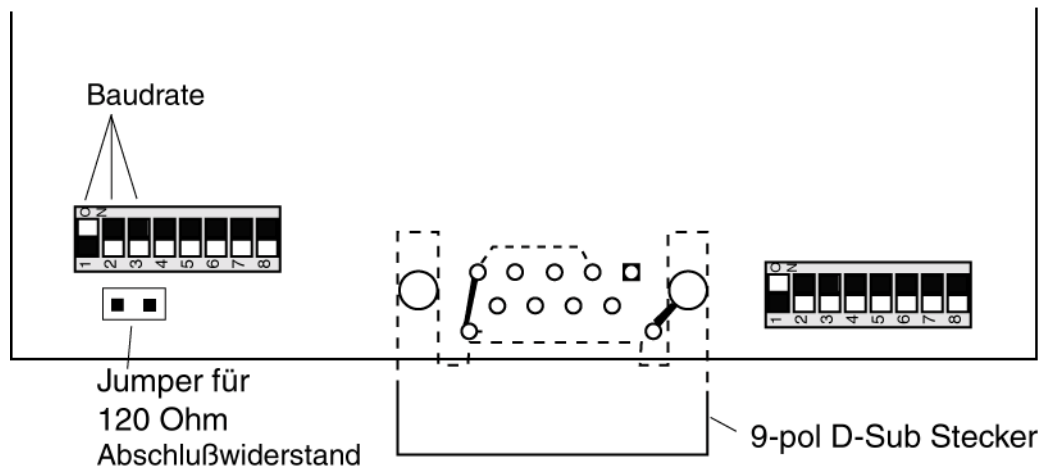
- 37: Dezimalkonstante erwartet
- 38: run erwartet
- 39: [new, neg, not] erwartet
- 40: new oder neg erwartet
- 41: list erwartet
- 42: quit erwartet
- 43: off erwartet
- 44: Programm läuft noch
- 45: pgm erwartet
- 46: Programmiermodus nicht aktiv
- 47: del erwartet
- 48: Datum erwartet
- 49: Change: nur Konstante erlaubt
- 50: Dezimalkonstante oder pos erwartet
- 51: pos erwartet
- 52: psave erwartet
- 53: [tr tron troff] erwartet
- 54: Programm laeuft nicht
- 55: troff erwartet
- 56: ver erwartet
- 57: 1, 2, 3 oder 4 erwartet
- 58: pos oder possave erwartet
- 59: lp erwartet
- 60: possave erwartet
- 61: Datum oder Parameter erwartet
- 62: Komma nicht erlaubt
- 63: not erwartet
- 64: unbekannter Zustand
- 65: Programmstart nicht möglich wenn Serviceschalter an ist
- 66: Programmstart nicht möglich, Zustandsklasse-1 Fehler
- 67: Stop Schalter ist aktiv
- 68: Stop Schalter ist offen
- 69: kein gültiges Programm vorhanden
- 70: Antrieb muß stehen
- 71: Sprungziel unbekannt
- 72: Sprungziel ungültig
- 73: Stack Überlauf
- 74: unbekannter Opcode, return vergessen ?
- 75: unbekannter fxxx Opcode
- 76: ungültiger Opcode fuer Zieladresse
- 77: unbekannter f7xx Opcode
- 78: Endschalter offen
- 79: Antrieb ist nicht freigegeben
- 80: unbekannter Positioniermodus
- 81: Antrieb muß konstant fahren oder stehen
- 82: ungültiger EEPROM Modus Wert
- 83: label bereits definiert: L
- 84: Die Wegdifferenz ist zu groß
- 85: Die neue Position ist zu groß
- 86: Die neue Position ist zu klein

- 87: neuer Restweg zu kurz
- 88: Vergleich Position 1 ist zu groß
- 89: Vergleich Position 1 ist zu klein
- 90: Vergleich Position 2 ist zu groß
- 91: Vergleich Position 2 ist zu klein
- 92: Der neue Modulwert ist zu groß
- 93: nicht schreibbar, während der Antrieb positioniert
- 94: Die Lötbrücke ist für diesen Bereich falsch eingestellt.
- 95: Die negative Grenzposition ist größer als die positive
- 96: Exponent zu groß
- 97: Exponent zu klein
- 98: Das Rechenergebnis ist zu groß
- 99: Das Rechenergebnis ist zu klein
- 100: Das Rechenergebnis ist zu groß zum anzeigen
- 101: Das Rechenergebnis ist zu klein zum anzeigen
- 102: Division durch 0
- 103: Bus Stopbit ist aktiv
- 104: Subindex nicht vorhanden
- 105: Wert kann nicht geschrieben werden
- 106: Wert kann nicht gelesen werden
- 107: Polynom mit Getriebeispiel nicht erlaubt
- 108: Keine neuen Polynomdaten verfügbar
- 109: Wait erwartet
- 110:  $\leq 0$  erwartet
- 111:  $= 0$  erwartet
- 112: 0 erwartet
- 113: 0 oder 1 erwartet
- 114:  $>$  erwartet
- 115: 1 erwartet
- 116:  $=$  erwartet
- 117: 3 erwartet
- 118: Polynom Ende zu kurz zum Anhalten (fehlerhafte Definition Polynom Abschnitt)
- 119: A zu klein
- 120: A zu gross
- 121: V zu klein
- 122: V zu gross

### 3. Abschlußwiderstand 120 Ohm

Bei mehreren Teilnehmern an einem CANopen Bus darf nur der letzte Teilnehmer einen 120 Ohm Abschlußwiderstand zwischen den Anschlusspins CAN\_Low (9-pol. D-SUB Pin 2) und CAN\_High (9-pol. D-Sub Pin 3) haben.

Bei der SERS kann der Abschlusswiderstand über einen Jumper aktiviert werden.



Jumper gesetzt: 120 Ohm Abschlusswiderstand im CANopen Bus

Jumper entfernt: kein Abschlusswiderstand