

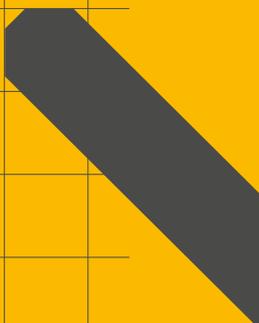
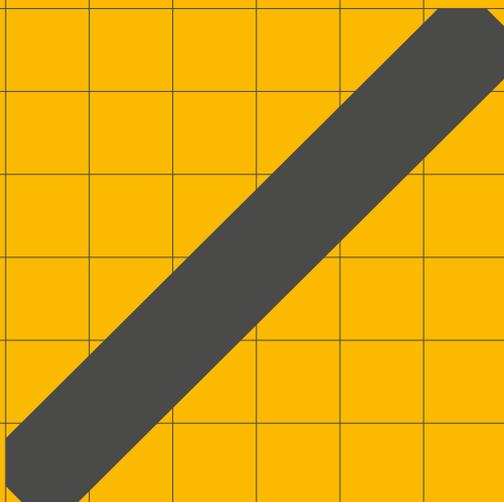
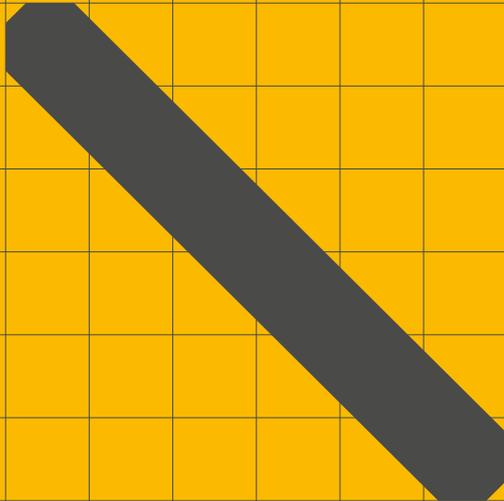
EWELLIX

MAKERS IN MOTION

INSTALLATIONS-, BEDIENUNGS- UND
INSTANDHALTUNGSANLEITUNG

Schnittstelle RS232 SCU Steuerung





Inhalt

1.0 Einführung	4
1.1 Inhalt	4
1.1.1 Gültigkeitsbereich	4
1.1.2 Zielgruppe	4
1.2 Darstellungskonventionen	5
1.2.1 Sicherheitshinweise	5
1.2.2 Sonstige Hinweise	5
1.2.3 Codebeispiele	5
1.2.4 Querverweise	5
1.2.5 Referenzierung von Bilddetails	5
2.0 Sicherheit	6
2.1 Allgemeine Sicherheitshinweise	6
3.0 Technische Übersicht	7
3.1 Anschlusskabel	7
3.2 Physical Layer	7
3.3 Data Link Layer	7
3.4 Network Layer	7
3.5 Transport Layer	8
3.5.1 Telegramm Struktur	8
4.0 Kommunikationsprotokoll	9
4.1 Befehlssatz	9
4.2 Kommunikations-Error und Acknowledge Codes	11
4.3 Verwendete Abkürzungen	11
4.4 Datenverzeichnis	12
4.5 Funktionsverzeichnis	15
4.6 SCU Error Code	16
4.7 Ansteuerung der Antriebe	17
4.7.1 Funktionsdefinition	17
4.7.2 Einstellen der Bewegungsparameter	17
4.8 Auslesen von Information	17
4.8.1 Positionsdaten	17
5.0 Kommunikationsbeispiele	18
5.1 Beispiel: Position anfahren und zurücklesen der aktuellen Position mit SCP11 Parametrisierung	18
6.0 Codebeispiele	20
6.1 Berechnung der Checksumme	20
7.0 Struktur Definitionen	23

WARNING

Bitte Lesen Sie die Anleitung vor Inbetriebnahme oder Wartung der Antriebe. Werden die Hinweise nicht befolgt, kann dies zu Fehlern am Antrieb, zu Verletzungen, Tod oder Beschädigungen führen.

1.0 Einführung

Dieses Kapitel enthält Angaben zu Aufbau und Gestaltung der Bedienungsanleitung.

Es erleichtert Ihnen die Handhabung der Bedienungsanleitung und ermöglicht den raschen Zugriff auf die gewünschten Informationen.

1.1 Inhalt

Diese Bedienungsanleitung enthält die Beschreibung der Seriellen Schnittstelle RS232 der SCU Steuerung. Beachten Sie, dass die RS232 Schnittstelle eine Option der SCU Steuerung ist und gemäss Typenschlüssel bestellt werden muss..

1.1.1 Gültigkeitsbereich

Die Angaben in dieser Bedienungsanleitung beziehen sich auf die serielle Schnittstelle der SCU Steuerung mit folgender Identifizierung:

- Hersteller: Ewellix
- Produktname: SCU Steuerung mit serieller Schnittstelle RS232
- Typenbezeichnung: SCUxx-xxxxx1-xxxx
- n Baujahr: ab 2007 mit Firmwave-Version V2B0
- CE-Kennzeichnung: gemäss technischer Dokumentation

1.1.2 Zielgruppe

Diese Anleitung richtet sich an den Entwicklungsingenieur, der die erforderlichen Fach-kenntnisse für die Entwicklung der Ansteuerungssoftware zum Betrieb dieses Produkts besitzt..

1.2 Darstellungs-konventionen

Wir verwenden in dieser Bedienungsanleitung einige Abkürzungen und Markierungen zur Kennzeichnung von Textabschnitten oder Hinweisen..

1.2.1 Sicherheitshinweise

WARNUNG:

Sicherheitshinweis zur Unterrichtung über Gefahr für irreparable Sach- und Personenschäden, die aufgrund der Gefahrenanalyse verblieben sind. Mit Hinweis auf Schutzmassnahmen und eventuell erforderliche Spezialausbildung und persönliche Schutzausrüstung.

Diese Hinweise sind wie folgt ausgezeichnet:

⚠️ WARNUNG

Quelle der Gefahr wird genannt. Beschreibung möglicher Konsequenzen!

- Handlungsanweisung zur Abwendung der Gefahr.

VORSICHT:

Sicherheitshinweis zur Unterrichtung über die Restgefahren, die aufgrund

der nicht vollständigen Wirksamkeit der Schutzmassnahmen für Sach- und Personenschäden verblieben sind; Hinweis auf eventuell erforderliche Spezialausbildung und persönliche Schutzausrüstung.

Diese Hinweise sind wie folgt ausgezeichnet:

⚠️ CAUTION

Quelle der Gefahr wird genannt. Beschreibung möglicher Konsequenzen!

- Handlungsanweisung zur Abwendung der Gefahr..

1.2.2 Sonstige Hinweise

Hinweise auf wichtige und/oder nützliche Zusatzinformationen die bei den Servicearbeiten zu beachten sind.

Diese Hinweise sind wie folgt ausgezeichnet:

Hinweis:

Hinweistext wird genannt.

1.2.3 Codebeispiele

Die in dieser Anleitung aufgeführten Codebeispiele sind in C++ programmiert, sie dienen der Veranschaulichung.

Die Codebeispiele sind durch die übliche Formatierung für Software hervorgehoben:

```
unsigned short HelloWorld()
{
  //@todo
}
```

1.2.4 Querverweise

Querverweise auf Abschnitte an anderer Stelle der Bedienungsanleitung sind in Klammern gesetzt. Sie enthalten den entsprechenden Überschrifttext und die Seitenzahl.

Querverweise sind wie folgt ausgezeichnet:

(L→ 1.2.4 Querverweise, Seite 5).

1.2.5 Referenzierung von Bilddetails

Details in Abbildungen sind der Reihe nach im Uhrzeigersinn durchbuchstabiert und im Text entsprechend referenziert.

2.0 Sicherheit

Sicherheitshinweise in dieser Anleitung werden nach Ihrer Anwendbarkeit wie folgt

unterschieden.

- **Allgemeine Sicherheitshinweise**

Diese Sicherheitshinweise gelten generell und sind beim Austausch jeder Baugruppe zu beachten. Sie sind im Abschnitt Allgemeine Sicherheitshinweise genannt.

- **Spezielle Sicherheitshinweise**

Diese Sicherheitshinweise sind nur bei einigen Baugruppen relevant. Sie sind in der Beschreibung des Austauschs der betreffenden Baugruppe genannt.

2.1 Allgemeine Sicherheitshinweise

Beachten Sie bei allen Servicearbeiten die folgenden Sicherheitshinweise:

⚠️ WARNUNG

Servicearbeiten an eingeschaltetem Gerät.
Elektrischer Schlag!!

- Schalten Sie das Gerät vor Servicearbeiten aus und ziehen Sie den Netzstecker.

⚠️ WARNUNG

Servicearbeiten an eingeschaltetem Gerät.
Elektrischer Schlag!

- Achten Sie auf korrekte Zugentlastung und Kabelführung beim Einbau der Baugruppen..

⚠️ VORSICHT

Unbeabsichtigtes Bewegen des Behandlungstisches.
Beschädigung freigelegter Geräteteile!

- Schliessen Sie vor Beginn der Servicearbeiten alle Feststellbremsen..

⚠️ VORSICHT

Verwenden von ungeeignetem Werkzeug oder Material.
Beschädigung / Fehlfunktion des Geräts!

- Benutzen Sie ausschliesslich Originalteile und das angegebene Spezialwerkzeug.

3.0 Technische Übersicht

In diesem Kapitel sind die grundsätzlichen technischen Eigenschaften der seriellen Schnittstelle aufgeführt.

HINWEIS

Verfügt der Remote-Teilnehmer über kein medizinisches Netzteil (Sicherheit nach EN60601-1), muss für einen ordnungsgemässen Betrieb der RS232-Schnittstelle die Endanwendung geerdet sein.

3.1 Anschlusskabel

Empfohlenes Anschlusskabel: ZKA-160658-3000

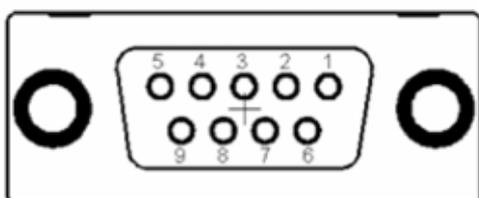
Abb. 1



3.2 Physical Layer

- Elektrische Charakteristik gemäss RS232 Definition
- Halb duplex
- Bidirektional
- Baudrate: Bei Standard Steuerungen ist die Baudrate auf 38400 eingestellt.
Bei kundenspezifischen Steuerungen kann die Baudrate auf die folgenden Werte eingestellt werden: 9600, 19200, 38400,
- Steckkontakt: 9-Pol SUB-D (female)
- Die Steuerleitungen werden nicht benutzt, jedoch müssen DTR und RTS permanent aktiv eingeschaltet sein, weil daraus die Speisung des RS232 Konverters in der Steuerung erzeugt wird. Anstelle des DTR und RTS Signals kann eine separate Spannungsquelle von 5.5...15VDC/30mA angeschlossen werden (+ auf Pin 4 DTR oder Pin 7 RTS und – auf Pin 5 GND)
- Anschlussbelegung:

Abb. 2



- 2. RxD
- 3. TxD
- 4. DTR
- 5. GND
- 7. RTS

3.3 Data Link Layer

- Ein Startbit
- 8 Datenbits (LSB first)
- Ein Stoppbit
- Kein Paritybit
- Kein Handshake

3.4 Network Layer

- Die Steuerung arbeitet als Slave und antwortet auf die Anfragen des Masters (z.B. PC-Programm)
- Der Slave antwortet auf jede Anfrage des Masters
- Maximale Verzögerungszeit auf die Anfrage: 2000 ms
- Maximale Verzögerung zwischen den einzelnen Bytes des Telegramms: 1000 ms
- Wenn die Steuerung mit Akku betrieben wird und in der Parametrisierung <Low Power> = Enabled eingestellt ist und die Steuerung sich im Low Power Mode befindet, kann die Steuerung in den Remote Mode gebracht werden, indem die Leitung RxD während min 100 ms. im Zustand „Space“ (Level > +3 V) gehalten wird (ab FW V2B1).
- Wenn die Steuerung mit Akku betrieben wird und in der Parametrisierung <Low Power> = Enabled eingestellt ist und der Remote Mode aktiv ist aber kein Kommando ansteht, geht die Steuerung nicht in den Low Power Mode. Wird die Kommunikation in diesem Zustand abgebrochen, geht die Steuerung in den Low Power Mode (ab FW V2B1).

3.5 Transport Layer

3.5.1 Telegramm Struktur

Request: <X><Y>[<P1><P2>...<Pn>]<C1><C2>

Reply: <X><Y>[<P1><P2>...<Pn>]<C1><C2>

Request: <X><Y>[<P1><P2>...<Pn>]<C1><C2>

Reply: <X><Y>[<P1><P2>...<Pn>]<C1><C2>

[] : optional

<X> : Major Kommando Nummer (1Byte)

<Y> : Minor Kommando Nummer (1Byte)

<P1>...<Pn> : Parameter Bytes (Intel Little Endian Format, LS Byte... MS Byte)

<C1> : Low Byte der 16 Bit Telegramm Checksumme

<C2> : High Byte of 16 Bit Telegramm Checksumme

Die Checksumme wird durch den standardisierten Algorithmus CCITT CRC-16 berechnet. Das Polynom des Algorithmus ist $CRC16 = x^{16} + x^{12} + x^5 + 1$. Der Startwert ist 0.

Jede Antwort beinhaltet ein ACK Byte, welches den Status des Geräts enthält. Manche Antworten enthalten den Parameter ctp in P1/P2. Dieser definiert die Anzahl der folgenden

Daten Bytes. Jede Antwort, die mehr als 1 Datenbyte enthält, verwendet ein ctp für die Definition der Datenlänge.

Ein Telegramm kann folgendermassen beschrieben werden:

Request: <X><Y>[<P1><P2>...<Pn>]<C1><C2>

Reply: <X><Y><ACK>[<ctp1><ctp2>=n<P><P4>...<Pn+2>]<C1><C2>

4.0 Kommunikationsprotokoll

4.1 Befehlssatz

Die folgenden Kommandos sind nach Netz ein oder im Batteriebetrieb verfügbar:

- Mit dem RO-Kommando wird die Remotefunktionalität aktiviert.
- Mit dem RA-Kommando wird die Remotefunktionalität abgebrochen.
- Um die Remotefunktionalität aufrecht zu erhalten, muss das RC-Kommando zyklisch wiederkehrend mindestens alle 1000 ms aufgerufen werden. Jedes weitere Remote-Kommando (RG, RT, RC, RE, RS, ausgenommen RO, RA) muss zyklisch wiederkehrend mindestens alle 500 ms aufgerufen werden.
- Die RG-, RT-, RC-, RE- und RS-Kommandos sind nur verfügbar, wenn die Remotefunktionalität aktiviert ist

Die Steuerleitungen DTR und RTS müssen permanent aktiv eingeschaltet sein, damit der RS232 Konverter gespeist wird und eine Kommunikation mit der SCU möglich ist.

Table 1

Cmd <X>Y>	Anfrage parameter			Antwort parameter							Beschreibung		
	P1	P2	P3	P4	P5	Pn	P1	P2	P3	P4		Pn	
RG	Remote data get	data_ID [0]	data_ID [1]	-	-	-	-	ACK, **E	ctp1	ctp2	1. Datenbyte	n-2. Datenbyte	Datentransfer von SCU P2 to Pn in Antwort nur wenn P1 = ACK Datentransfer zu SCU. Nur data_ids 3xxx sind erlaubt (RemoteData!)
RT	Remote data transfer	ctp1	ctp2	data_ID [1]	1. Datenbyte	n-4. Datenbyte	ACK, **E	-	-	-	-	-	
RC	Remote cyclic	ctp1	ctp2	Index von cyclicObj	1. Byte Schreib-Obj cyclicObj	2. Byte Schreib-Obj cyclicObj	n-3. Byte Schreib-Obj cyclicObj	ACK, **E	ctp1	ctp2	1. Byte Lesedaten von cyclicObj	n-3. Byte Lesedaten von cyclicObj	Muss mindestens alle 500 ms gesendet werden, damit die SCU im Remote-Mode verbleibt (WDT). Mit P3 = -1 werden keine Daten transferiert, andernfalls ist P3 der Index eines cyclicObj, welches die Anfrage/Antwort Daten definiert P3=0: cyclicObj mit dataID=3001. P2 to Pn ist in der Antwort nur wenn P1 = ACK
RE	Remote execute function	fnc_ID	para_ID [0]	para_ID [1]	-	-	ACK, **E	-	-	-	-	-	Ausführung einer Funktion. P1 ist der Index im Funktions-Verzeichnis. P2/3 sind zusätzliche Funktions-parameter, beachten. Beachte Definition von fnc_ID and para_ID.
RS	Remote stop function	fnc_ID	para_ID [0]	-	-	-	ACK, **E	-	-	-	-	-	Stoppt eine Funktion. P1 ist der Index im Funktions-Verzeichnis. P2/3 sind zusätzliche Funktions-parameter, beachten. Beachte Definition von fnc_ID and para_ID.
RO	Remote mode open	Safety ID	para_ID [0]	-	-	-	ACK, **E	-	-	-	-	-	Safety_ID = 0: Bei einem Kommunikations-Time-out (0,5 s), werden alle Bewegungen abgebrochen bzw. keine Bewegungen gestartet. Bei einem Kommunikations-Time-out stehen nur die RC-, RA- und RO-Kommandos zur Verfügung. Safety_ID = 1: Bei einem Kommunikations-Time-out (0,5 s), werden alle Bewegungen abgebrochen. Safety_ID = 2: Bei einem Kommunikations-Time-out (0,5 s), werden nur Remote-Bewegungen abgebrochen. (ab FWV2B3)
RA	Remote mode abort	-	-	-	-	-	ACK, **E	-	-	-	-	-	Setzt die SCU in normal Mode (ohne Reset)

4.2 Kommunikations-Error und Acknowledge Codes

Table 2

Code	Hax	Dec	Name	Beschreibung
ACK	06	6	Command acknowledged	Anfrage akzeptiert
CSE	80	128	Checksum error	Fehler in der Telegram Checksumme
PDE	81	129	Parameter data error	Fehler in den Telegram Datenbytes
PCE	82	130	Parameter count error	Fälscher Zählerstand der Telegram Datenbytes
ICE	83	131	Invalid command error	Unbekannter Kommando-Code
PE	84	132	Permission error	Kommando nicht möglich im aktuellen SCU Modus/Zustand

4.3 Verwendete Abkürzungen

Table 3

Code	Wert	Beschreibung
ctp	Dyn	Anzahl folgende Telegram Bytes
data_ID	Dyn	Index in Datenverzeichnis (siehe Tabelle Datenverzeichnis)
fnc_id	Dyn	Index in Funktionsverzeichnis (siehe Tabelle Funktionsverzeichnis)
para_id	Dyn	Zusätzlicher Parameter abhängig von der Funktion (siehe Tabelle Funktionsverzeichnis)

4.4 Datenverzeichnis

Die spezifischen Einstellungen und der Status der Steuerung können über das Datenverzeichnis abgefragt werden (RG-Kommando). Es können sowohl einzelne Werte als auch ganze Blöcke abgefragt werden. Die Werte mit Collection Index 3 000 können mit dem RT Kommando beschrieben werden

Primary collection index	Secondary collection index	Daten index	Name	Datentyp	Kommentar
0000		0001	Firmware info Name Version CS	STRING	Size = 31 byte
		0002	Configuration info Name Version CS	STRING	Size = 36 byte
0010		0011	Actual_Position Actuator 1	INT32	Einheit: Encoder Flanken (Count)
		0012	Actual_Position Actuator 2	INT32	
		0013	Actual_Position Actuator 3	INT32	
		0014	Actual_Position Actuator 4	INT32	
		0015	Actual_Position Actuator 5	INT32	
		0016	Actual_Position Actuator 6	INT32	
0020			Actual_State_Binary Inputs 1...4	UINT8	Logischer Level Bit 0: Binär Eingang 1 (0 = nicht aktiv / 1 = aktiv) Bit 1: Binär Eingang 2 (0 = nicht aktiv / 1 = aktiv) Bit 2: Binär Eingang 3 (0 = nicht aktiv / 1 = aktiv) Bit 3: Binär Eingang 4 (0 = nicht aktiv / 1 = aktiv) Eingang Level Bit 4: Binär Eingang 1 (0 = nicht aktiv / 1 = aktiv) Bit 5: Binär Eingang 2 (0 = nicht aktiv / 1 = aktiv) Bit 6: Binär Eingang 3 (0 = nicht aktiv / 1 = aktiv) Bit 7: Binär Eingang 4 (0 = nicht aktiv / 1 = aktiv)
0030		0031	Actual_State_Analogue_Input_1	UINT16	Wert: 0...600 Auflösung 0.01V Bereich: 0...6.00V
		0032	Actual_State_Analogue_Input_2	UINT16	
		0033	Actual_State_Analogue_Input_3	UINT16	
		0034	Actual_State_Analogue_Input_4	UINT16	
0040			Actual_State_Keys	UINT32	Bit 0: K1 ... Bit 19: K20 Bit 20 ... Bit 31 nicht benutzt (0 = offen / 1 = geschlossen)
0060		0061	Number_cycle_off_on_off_Relay_in A1	UINT32	
		0062	Number_cycle_off_on_off_Relay_in A2	UINT32	
		0063	Number_cycle_off_on_off_Relay_in A3	UINT32	
		0064	Number_cycle_off_on_off_Relay_in A4	UINT32	
		0065	Number_cycle_off_on_off_Relay_in A5	UINT32	
		0066	Number_cycle_off_on_off_Relay_in A6	UINT32	

Primary collection index	Secondary collection index	Daten index	Name	Datentyp	Kommentar
0000	0070	0071	Number_cycle_off_on_off_Relay_out A1	UINT32	
		0072	Number_cycle_off_on_off_Relay_out A2	UINT32	
		0073	Number_cycle_off_on_off_Relay_out A3	UINT32	
		0074	Number_cycle_off_on_off_Relay_out A4	UINT32	
		0075	Number_cycle_off_on_off_Relay_out A5	UINT32	
		0076	Number_cycle_off_on_off_Relay_out A6	UINT32	
		0080	0080	0081	Number_Actuator error A1
0082	Number_Actuator error A2			UINT32	
0083	Number_Actuator error A3			UINT32	
0084	Number_Actuator error A4			UINT32	
0085	Number_Actuator error A5			UINT32	
0086	Number_Actuator error A6			UINT32	
008F	Number_Total_Over_Current			UINT32	
0090	0090	0091	Cumulated_Stroke A1	UINT32	Einheit: Encoder Flanken
		0092	Cumulated_Stroke A2	UINT32	
		0093	Cumulated_Stroke A3	UINT32	
		0094	Cumulated_Stroke A4	UINT32	
		0095	Cumulated_Stroke A5	UINT32	
		0096	Cumulated_Stroke A6	UINT32	
00A0	00A0	00A1	Current A1	UINT16	Wert: 0...1000 Einheit: Festkomma 0.1A Bereich: 0..100A
		00A2	Current A2	UINT16	
		00A3	Current A3	UINT16	
		00A4	Current A4	UINT16	
		00A5	Current A5	UINT16	
		00A6	Current A6	UINT16	
00B0	00B0	00B1	Max_Current A1	UINT16	Wert: 0...1000 Einheit: Festkomma 0.1A Bereich: 0..100A
		00B2	Max_Current A2	UINT16	
		00B3	Max_Current A3	UINT16	
		00B4	Max_Current A4	UINT16	
		00B5	Max_Current A5	UINT16	
		00B6	Max_Current A6	UINT16	
		00BF	Max_Total_Current	UINT16	
		00C0	Max_Temp_Rectifier_FET	UINT8	
00D0	00D0	00C1	Number_Over_Temp_Rectifier_FET	UINT32	
		00D1	Error_Code 1 (last recent)	UINT32	Aufbau siehe Kapitel 4.6 SCU error code
		00D2	Error_Code 2 (History 1)	UINT32	
		00D3	Error_Code 3 (History 2)	UINT32	
		00D4	Error_Code 4 (History 3)	UINT32	
		00D5	Error_Code 5 (History 4)	UINT32	
00E0	00E0	00E1	Actuator status 2 A1	UINT8	
		00E2	Actuator status 2 A2	UINT8	
		00E3	Actuator status 2 A3	UINT8	
		00E4	Actuator status 2 A4	UINT8	
		00E5	Actuator status 2 A5	UINT8	
		00E6	Actuator status 2 A6	UINT8	
00F0	00F0	00F1	Speed A1	UINT16	Wenn speed select relativ: Einheit: % Bereich: 0..100 Wenn speed select absolut: Einheit: Encoder Flanken/ s Bereich: 0..1 000
		00F2	Speed A2	UINT16	
		00F3	Speed A3	UINT16	
		00F4	Speed A4	UINT16	
		00F5	Speed A5	UINT16	
		00F6	Speed A6	UINT16	

Primary collection index	Secondary collection index	Daten index	Name	Datentyp	Kommentar
0000	0100		Battery Mains	UINT8	Bit 0 0/1: Netz nicht -/angeschlossen Bit 1 0/1: Batterie nicht-/angeschlossen Bit 2 0/1: Laderegler aus-/eingeschaltet Bit 3 0/1: Ladevorgang deaktiv / aktiv
	0110		Binary Output Status	UINT8	Bit 0 0/1: Binary Output 1 aus/ein Bit 1 0/1: Binary Output 1 aus/ein
	0120		LED HS	UINT8	Bit 0 0/1: LED1 Handschalter aus/ein Bit 1 0/1: LED2 Handschalter aus/ein
	0130		LED LB	UINT8	Bit 0 0/1: LED1 Lockingbox aus/ein Bit 1 0/1: LED2 Lockingbox aus/ein ... Bit 7 0/1: LED8 Lockingbox aus/ein
		0140	Buzzer	UINT8	Bit 0 0/1: Buzzer aus/ein
		0150	Sensor Supply	UINT8	Bit 0 0/1: Sensor Supply aus/ein
		0162	Lock Status	UINT16	Bit 0 0/1: Funktion 0 entsperrt / gesperrt Bit 1 0/1: Funktion 1 entsperrt / gesperrt ... Bit 9 0/1: Funktion 10 entsperrt / gesperrt
		0164	Battery voltage	UINT16	Einheit: Festkomma 0,1V Bereich: 0... 40,0 V
		0165	Locking Box detected	UINT8	0..2 locking box
		0166	User	UINT8	User 1..4
	0170	0171	Actuator Status 1 A 1	UINT8	Bit 0 0/1: Antrieb nicht vorhanden/ Antrieb vorhanden
		0172	Actuator Status 1 A 2	UINT8	Bit 1 0/1: signal limit_in_out nicht-/aktiv
		0173	Actuator Status 1 A 3	UINT8	Bit 2 0/1: signal switch 1 nicht-/aktiv
		0174	Actuator Status 1 A 4	UINT8	Bit 3 0/1: signal switch 2 nicht-/aktiv Bit 4 0/1: motion nicht-/aktiv
		0175	Actuator Status 1 A 5	UINT8	Bit 5 0/1: in position nicht-/erreicht
		0176	Actuator Status 1 A 6	UINT8	Bit 6 0/1: out position nicht-/erreicht Bit 7 0/1 Stroke nicht-/ aufgenommen
Configuration data items					
1000	1010	1011-1016	Conversion factor A 1-6	FLOAT	
User position data items					
2000		2001	UserPositionData A 1	STRUCT	Struktur Definition gemäss Kapitel 7.0 Struktur Definitionen, Seite 23: ACTUATOR_POSITIONS
		2002	UserPositionData A 2	STRUCT	
		2003	UserPositionData A 3	STRUCT	
		2004	UserPositionData A 4	STRUCT	
		2005	UserPositionData A 5	STRUCT	
		2006	UserPositionData A 6	STRUCT	
Remote data items					
Gespeichert in flüchtigem Speicher. Initialisiert nach Reset mit voreingestellten Werten.					
3000		3001	CyclicObj 1	UINT16[12]	Mit der CyclicObj Definition können die transferierten Daten zu und von der SCU festgelegt werden, die mit jedem RC Kommando übertragen werden. Die Daten Indizes, welche in den ersten 6 Bytes (para[0..5] gesetzt sind, definieren die Daten, die zur SCU gesendet werden, (Schreib- daten) und die Daten Indizes, welche in den letzten 6 Bytes (para[6..11]) gesetzt sind, definieren die Daten, die von der SCU zurückgesendet werden (Lesedaten). Ein Daten Index von -1 steht für kein Daten- transfer. Sämtliche Daten können von derSCU gelesen werden, aber nur die Daten mit den Indizes 3xxx können geschrieben werden. Voreinstellung: -1
		3002	CyclicObj 2	UINT16[12]	
		3003	CyclicObj 3	UINT16[12]	
		3004	CyclicObj 4	UINT16[12]	
		3005	CyclicObj 5	UINT16[12]	
	3010	3011-301A	Remote Speed F1-10	UINT16	Voreinstellung: Function speed aus Configuration. Wenn speed select relativ: Einheit: % Bereich: 0..100 Wenn speed select absolut: Einheit: Encoder Flanken/ s Bereich: 0..1 000
	3020	3021-3026	Remote Position A1-6	INT32	Voreinstellung: Memory 1 / User 1 Position von UserPositionData (DynamicConfiguration) Einheit: Encoder Flanken

4.5 Funktionsverzeichnis

Table 5

Func-ID	Wert (dez.)	Verwendet von Kommando	Beschreibung	Para_ID[x]
F1...F10	0...9	RE, RS	Motion Function (abhängig von Parametrisierung)	Para_ID[0] gemäss Tab 4-1 Para_ID[1] = -1
F11	10	RE, RS	Buzzer (ab FW V2B1)	Para_ID[0] = -1 Para_ID[1] = -1
F17	16	RE, RS	Binary Output 1 (ab FW V2B1)	Para_ID[0] = -1 Para_ID[1] = -1
F18	17	RE, RS	Binary Output 2 (ab FW V2B1)	Para_ID[0] = -1 Para_ID[1] = -1
F20	19	RE, RS	Emergency stop (ab FW V2B1)	Para_ID[0] = -1 Para_ID[1] = -1
F21	20	RE, RS	Operating unit Led1 (ab FW V2B1)	Para_ID[0] = -1 Para_ID[1] = -1
F22	21	RE, RS	Operating unit Led2 (ab FW V2B1)	Para_ID[0] = -1 Para_ID[1] = -1

Parameter depends on function:

Table 6

Used for func_ID	Para_ID[1]	Wert (dez.)	Beschreibung
F1-F10 (nur bei RE Kommando)	motion_direction	0-9	0: undefinierte Richtung (no motion) 1: Move to position In 2: Move to position Out 3: Move to position Mem1 4: Move to position Mem2 5: Move to position Mem3 6: Move to position Mem4 7: Move to position Intermediate In 8: Move to position Intermediate Out 9: Move to Remote Position
F1-F10 F1-F10 (nur bei wird nicht)	motion_stop RS Kommando	0-1	0: Fast Start/Stop (Start/Stop Rampe berücksichtigt) 1: Soft Start/Stop (Start/Stop Rampe wird berücksichtigt)

Die ungenutzten Parameter in der Telegram Struktur sollten auf -1 (unused_para) gesetzt sein.

Die Funktion von F1 – F10 ist in der Parametrisierung der Steuerung festgelegt. Einer Funktion können eins bis sechs Antriebe zugewiesen sein. Bei mehr als einem zu einer Funktion zugewiesenen Antrieb können die Antriebe untereinander koordiniert sein:

- Simultanlauf gleiche Richtung oder Gegenrichtung (gleichzeitiges Starten / Stoppen, aber keine Positionssynchronisation)
 - Synchronisierter Gleichlauf in gleiche Richtung oder Gegenrichtung (geregelter Positionssynchronisation)
- Der zweite Fall kann auch mit einer konstanten Differenz zwischen den Antrieben parametrisiert sein.

4.6 SCU Error Code

Bit in error field	Ursache	Auftretensbedingung	Verhalten	Table 7
Bit 1	CRC Fehler bei ROM Test. Fehlerhaftes ROM	–	Bewegungen werden gestoppt und Steuerung führt Reset durch.	
Bit 2	RAM Test ist fehlgeschlagen. Fehlerhaftes RAM.	–	Bewegungen werden gestoppt und Steuerung führt Reset durch.	
Bit 3	CPU Test ist fehlgeschlagen. Fehlerhafte CPU.	–	Bewegungen werden gestoppt und Steuerung führt Reset durch.	
Bit 4	STACK Ueberlauf detektiert.	–	Bewegungen werden gestoppt (Fast Stopp) und Steuerung führt Reset durch.	
Bit 5	Wathdog Reset	–	Bewegungen werden gestoppt (Fast Stopp) und Steuerung führt Reset durch.	
Bit 6	Handschaltertest fehlgeschlagen. Kurzschluss in Handschalter detektiert.	Nur wenn Handschalter als "safe" parametrisiert.	Bewegungen werden gestoppt (Fast Stopp)	
Bit 7	Test binäre Eingänge fehlgeschlagen. Kurzschluss zwischen binären Eingängen detektiert.	Nur wenn binäre Eingänge als safe und no analogue input parametrisiert.	Bewegungen werden gestoppt (Fast Stopp)	
Bit 8	Relais und FET Test fehlgeschlagen. Fehlerhaftes Relais oder FET.	Test wird bei Bewegungs- start durchgeführt.	Bewegung wird nicht ausgeführt.	
Bit 9	–	–	–	
Bit 10	Kommunikation mit MoveEnable Controller fehlgeschlagen. Keine Antwort von MoveEnable Controller.	–	Bewegungen werden gestoppt (Fast Stopp)	
Bit 11	MoveEnable Ausgang Ueberprüfung fehlgeschlagen. Der Ausgang vom MoveEnable Controller ist falsch.	–	Bewegungen werden gestoppt (Fast Stopp)	
Bit 12	Übertemperatur an Gleichrichter oder FET detektiert.	–	Bewegungen werden gestoppt (Fast Stopp)	
Bit 13	Abschaltung aufgrund Tiefentladung der Batterie	–	Bewegungen werden gestoppt (Fast stop). Steuerung schaltet sich aus.	
Bit 14	Summenstrom wurde überschritten.	Wenn Bewegung im Gang.	Bewegungen werden gestoppt (Fast stop). Bit wird mit nächster Bewegung zurückgesetzt.	
Bit 15: Antrieb 1 Bit 16: Antrieb 2 Bit 17: Antrieb 3 Bit 18: Antrieb 4 Bit 19: Antrieb 5 Bit 20: Antrieb 6	Fehler des Antriebs	Peak current Short circuit current Sensor monitor Over current (wenn nicht limit position) Time out (wenn nicht limit position)	Antrieb wird gestoppt (Fast stop). Bit wird mit nächster Bewegung zurückgesetzt.	
Bit 21	Positionsdifferenz zwischen den Antrieben zu gross (Synchronisierter Parallellauf)	Nur wenn synchro- nisierter Parallellauf parametrisiert.	Bewegung wird nicht gestartet oder wenn Bewegung im Gang wird die Bewegung gestoppt (Fast stop). Bit wird mit nächster Bewegung zurückgesetzt.	
Bit 22	Remote Kommunikation Time out	–	Abhängig von Safety ID	
Bit 23	–	–	–	
Bit 24	Lockingbox I ² C Kommunikation Fehler	Nur wenn locking box safe parametrisiert	Bewegungen werden nicht ausgeführt resp. gestoppt	
Bit 25	RAM Kopie von den EEPROM Konfigurationsdaten weist inkorrekte CRC auf	–	Bewegungen werden nicht ausgeführt resp. gestoppt	
Bit 26	RAM Kopie von den EEPROM Userdaten weist inkorrekte CRC auf	–	Bewegungen werden nicht ausgeführt resp. gestoppt	
Bit 27	EEPROM Lockingboxdaten weist inkorrekte CRC auf	–	Bewegungen werden nicht ausgeführt resp. gestoppt	
Bit 28	RAM Kopie von den EEPROM dynamischen Daten weist inkorrekte CRC auf	–	Bewegungen werden nicht ausgeführt resp. gestoppt	
Bit 29	RAM Kopie von den EEPROM Kalibrationsdaten weist inkorrekte CRC auf	–	Bewegungen werden nicht ausgeführt resp. gestoppt	
Bit 30	RAM Kopie von den EEPROM HW Settings weist inkorrekte CRC auf	–	Bewegungen werden nicht ausgeführt resp. gestoppt	
Bit 31	IO Test	Wird ausgeführt, wenn keine Bewegung aktiv ist.	Bewegungen werden nicht ausgeführt	
Bit 32	IDF Betriebssystem Fehler	–	Bewegungen werden nicht ausgeführt resp. gestoppt	

4.7 Ansteuerung der Antriebe

Die Ansteuerung der einzelnen Antriebe erfolgt über die Funktionen F1-F10. Mittels RE Kommando wird eine Funktion aktiviert und damit ein oder mehrere Antriebe gestartet. Jeder RE-Befehl muss mit RS gestoppt werden, selbst wenn der Antrieb durch das Erreichen der Endlage angehalten wird.

4.7.1 Funktionsdefinition

Die Funktionsdefinition entnehmen Sie bitte der Parametrisierungsdokumentation der Steuerung.

4.7.2 Einstellen der

Bewegungsparameter

Die Bewegungsparameter Geschwindigkeit und Zielposition können über die Indexe 3011 bis 301A bzw. 3021 bis 3026 vorgegeben werden. Die Geschwindigkeit gilt für die gewählte Funktion, die Zielposition ist auf die einzelnen Antriebe bezogen. Der REBefehl mit Parameter 9 startet die Bewegung.

Die Geschwindigkeit ist in Prozent (0-100%) oder in Inkrementen anzugeben. Dies hängt von der Parametrisierung der Steuerung ab. Bei Standard Steuerungen ist die Geschwindigkeit in Prozent eingestellt. Die untere Schwelle, bei der ein Antrieb in Bewegung versetzt wird, ist abhängig von Antriebsart und Belastung. Die Geschwindigkeit kann während der Bewegung verändert werden. Die Steuerung passt die Geschwindigkeit unter Einhaltung der Soft Start Rampe an.

4.8 Auslesen von Information

Über den RG Befehl können Betriebszustände und Informationen aus der Steuerung ausgelesen werden. Werte können sowohl einzeln als auch Blockweise abgefragt werden.

4.8.1 Positionsdaten

Die Indexe 0011 bis 0016 geben die aktuelle Position zurück. Der Gruppierungsindex 0010 liefert die Position aller 6 möglichen Antriebe zurück. Aus den Werten der Endpositionen und der Hublänge kann die Position im mm berechnet werden.

5.0 5. Kommunikationsbeispiele

5.1 Beispiel: Position anfahren und zurücklesen der aktuellen Position mit SCP11 Parametrisierung

Bei der SCP11 Parametrisierung sind alle Antriebe im Einzellauf eingestellt. Antrieb 1 ist auf Funktion 1, Antrieb 2 Funktion auf 2 und so weiter zugewiesen. Dadurch kann mit den Funktionen 1-6 die Antriebe einzeln gesteuert werden.

Ablauf:

- Kommunikation Mode öffnen mit RO (Safety ID)
- Remote Position von Antrieb 1 setzen
- Bewegung Antrieb 1 starten
- Status Antrieb 1 lesen. Prüfung ob Bewegung aktiv.
- Rücklesen der aktuellen Position von Antrieb 1
- Während des gesamten Ablaufs muss mindestens alle 500ms ein zyklisch wiederkehrende Kommunikation mit RC Kommando stattfinden.
Die RC Kommunikation arbeitet als Watchdog. Falls die RC Kommunikation ausfällt, stoppt die SCU alle in Bewegung befindlichen Antriebe und deaktiviert den Remote Modus.
- Bevor das erste Kommando zur SCU verschickt wird, muss ebenfalls eine RC Kommunikation stattgefunden haben (Aktivierung des Remote Modus)
- Kommunikation Mode schliessen mit RA

Tabelle 8

Periodische RC Kommunikation ohne Datentransfer in diesem Fall (ohne CyclObj)

Cmd	Name	Anfrage parameter						Antwort parameter			
		P1	P2	P3	P4	P5	Pn	P1	P2	P3	P4
RO	Remote Mode open	00	-	-	-	-	-	ACK, **E	-	-	-

-- Safe communication mode openn

Cmd	Name	Anfrage parameter						Antwort parameter						
		P1	P2	P3	P4	P5	Pn	P1	P2	P3	P4	P5	P6	P7
RC	Remote cyclic	01	00	-1	-	-	-	ACK, **E	-	-	-	-	-	-

Tabelle 9

Setzen der Remote Geschwindigkeit von Antrieb 1 auf Wert 100h mit Kommando RT (Datenindex 3011)

Cmd	Name	Anfrage parameter						Antwort parameter				
		P1	P2	P3	P4	P5	P6	P1	P2	P3	P4	Pn
RT	Remote data transfer	04	00	11	30	01	00	ACK, **E	-	-	-	-

Tabelle 10

Bewegung auf Remote Position mit Antrieb 1 ohne Start/Stop. Starten mit Kommando RE (Funktionsindex 0)

Cmd	Name	Anfrage parameter						Antwort parameter				
		P1	P2	P3	P4	P5	Pn	P1	P2	P3	P4	Pn
RE	Remote execute function	00	09	-1	-	-	-	ACK, **E	-	-	-	-

Tabelle 11

Status Antrieb 1 abfragen mit Kommando RG (Datenindex 0171)

Cmd	Name	Anfrage parameter						Antwort parameter						
		P1	P2	P3	P4	P5	Pn	P1	P2	P3	P4	P5	P6	P7
RG	Remote data get	71	01	-	-	-	-	ACK, **E	01	00	status			

Bit 4 in Status ist gesetzt solange die Bewegung aktiv ist.

Tabelle 12

Aktuelle Position von Antrieb 1 abfragen mit Kommando RG (Datenindex 0011)

Cmd	Name	Anfrage parameter						Antwort parameter						
		P1	P2	P3	P4	P5	Pn	P1	P2	P3	P4	P5	P6	P7
RG	Remote data get	11	00	-	-	-	-	ACK, **E	04	00	1. Datenbyte	2. Datenbyte	3. Datenbyte	4. Datenbyte

Cmd	Name	Anfrage parameter						Antwort parameter				
		P1	P2	P3	P4	P5	Pn	P1	P2	P3	P4	
RA	Remote Mode abord	-	-	-1	-	-	-	ACK, **E	-	-	-	-

Close communication mode

6.0 Codebeispiele

6.1 Berechnung der Checksumme

Die Checksumme wird aus dem standardisierten CCITT CRC16 Algorithmus ermittelt. Das Polynom ist $CRC16 = x^{16} + x^{12} + x^5 + 1$, der Startwert ist 0.

Die Berechnung der CRC-Checksumme ist rechenintensiv. Um diesen Aufwand zu verringern, wird idealerweise mit einer CRC Tabelle gearbeitet.

Tabelle 13

Codebeispiel 1: CRC Tabelle

```
static const unsigned short CRC_TABLE[256] = {
    0x0000    0x1021 0x2042 0x3063 0x4084 0x50A5 0x60C6 0x70E7
    0x8108    0x9129 0xA14A 0xB16B 0xC18C 0xD1AD 0xE1CE 0xF1EF
    0x1231    0x0210 0x3273 0x2252 0x52B5 0x4294 0x72F7 0x62D6
    0x9339    0x8318 0xB37B 0xA35A 0xD3BD 0xC39C 0xF3FF 0xE3DE
    0x2462    0x3443 0x0420 0x1401 0x64E6 0x74C7 0x44A4 0x5485
    0xA56A    0xB54B 0x8528 0x9509 0xE5EE 0xF5CF 0xC5AC 0xD58D
    0x3653    0x2672 0x1611 0x0630 0x76D7 0x66F6 0x5695 0x46B4
    0xB75B    0xA77A 0x9719 0x8738 0xF7DF 0xE7FE 0xD79D 0xC7BC
    0x48C4    0x58E5 0x6886 0x78A7 0x0840 0x1861 0x2802 0x3823
    0xC9CC    0xD9ED 0xE98E 0xF9AF 0x8948 0x9969 0xA90A 0xB92B
    0x5AF5    0x4AD4 0x7AB7 0x6A96 0x1A71 0x0A50 0x3A33 0x2A12
    0xDBFD    0xCBDC 0xFBBF 0xEB9E 0x9B79 0x8B58 0xBB3B 0xAB1A
    0x6CA6    0x7C87 0x4CE4 0x5CC5 0x2C22 0x3C03 0x0C60 0x1C41
    0xEDAE    0xFD8F 0xCDEC 0xDDCD 0xAD2A 0xBD0B 0x8D68 0x9D49
    0x7E97    0x6EB6 0x5ED5 0x4EF4 0x3E13 0x2E32 0x1E51 0x0E70
    0xFF9F    0xEFBE 0xDFDD 0xCFFC 0xBF1B 0xAF3A 0x9F59 0x8F78
    0x9188    0x81A9 0xB1CA 0xA1EB 0xD10C 0xC12D 0xF14E 0xE16F
    0x1080    0x00A1 0x30C2 0x20E3 0x5004 0x4025 0x7046 0x6067
    0x83B9    0x9398 0xA3FB 0xB3DA 0xC33D 0xD31C 0xE37F 0xF35E
    0x02B1    0x1290 0x22F3 0x32D2 0x4235 0x5214 0x6277 0x7256
    0xB5EA    0xA5CB 0x95A8 0x8589 0xF56E 0xE54F 0xD52C 0xC50D
    0x34E2    0x24C3 0x14A0 0x0481 0x7466 0x6447 0x5424 0x4405
    0xA7DB    0xB7FA 0x8799 0x97B8 0xE75F 0xF77E 0xC71D 0xD73C
    0x26D3    0x36F2 0x0691 0x16B0 0x6657 0x7676 0x4615 0x5634
    0xD94C    0xC96D 0xF90E 0xE92F 0x99C8 0x89E9 0xB98A 0xA9AB
    0x5844    0x4865 0x7806 0x6827 0x18C0 0x08E1 0x3882 0x28A3
    0xCB7D    0xDB5C 0xEB3F 0xFB1E 0x8BF9 0x9BD8 0xABBB 0xBB9A
    0x4A75    0x5A54 0x6A37 0x7A16 0x0AF1 0x1AD0 0x2AB3 0x3A92
    0xFD2E    0xED0F 0xDD6C 0xCD4D 0xBDAA 0xAD8B 0x9DE8 0x8DC9
    0x7C26    0x6C07 0x5C64 0x4C45 0x3CA2 0x2C83 0x1CE0 0x0CC1
    0xEF1F    0xFF3E 0xCF5D 0xDF7C 0xAF9B 0xBFBA 0x8FD9 0x9FF8
    0x6E17    0x7E36 0x4E55 0x5E74 0x2E93 0x3EB2 0x0ED1 0x1EF0
};
```

Codebeispiel 2 ist ein Beispiel für die Ermittlung der CRC-Checksumme aus der Tabelle. Die Returnierten 2 Byte müssen dem Kommando angehängt werden.

Tabella 14

Codebeispiel 3: Berechnung der Checksumme aus Tabelle

```
unsigned short CalculateChecksum (const unsigned char* pAdr, int len)
{
    if (len < 0)
    {
        ASSERT(FALSE);
        return 0;
    }
    unsigned short crc = 0;
    while (len--)
    {
        crc = static_cast<unsigned short>(CRC_TABLE[((crc >> 8) ^ *pAdr++) & 0xFF] ^ (crc << 8));
    }
    return crc;
}
```

Codebeispiel 4: Überprüfung der Antwort Checksumme

```
bool CheckResponseChecksum(const CArray<unsigned char>& responseData, bool
suppresstTimeoutError)
{
    CArray<unsigned char> tempData;
    unsigned char crcByte1;
    unsigned char crcByte2;
    DWORD bytesRead;
    tempData.Append(responseData);
    if (!ReadFile(m_hComm, &crcByte1, 1, &bytesRead, NULL) || (bytesRead !=1))
    {
        if(!GetLastError()) {
            // case time out
            if(!suppresstTimeoutError)
                AfxMessageBox(IDS_READ_ERROR_CRC);
        }
        else {
            Disconnect();
            AfxMessageBox(IDS_READ_ERROR);
        }
    }
    if (!ReadFile(m_hComm, &crcByte2, 1, &bytesRead, NULL) || (bytesRead !=1))
    {
        if(!GetLastError()) {
            // case time out
            if(!suppresstTimeoutError)
                AfxMessageBox(IDS_READ_ERROR_CRC);
        }
        else {
            Disconnect();
            AfxMessageBox(IDS_READ_ERROR);
        }
    }
    tempData.Add(crcByte2);
    tempData.Add(crcByte1);
    if (CalculateChecksum(tempData.GetData(), static_cast<int>(tempData.GetSize())) != 0)
    {
        AfxMessageBox(IDS_READ_ERROR_CRC_INVALID);
        return false;
    }
    else
    {
        return true;
    }
}
```

7.0 Struktur Definitionen

```
struct ACTUATOR_POSITIONSstruct {
    INT32 Position_Memory_1[USER_1];
    INT32 Position_Memory_2[USER_1];
    INT32 Position_Memory_3[USER_1];
    INT32 Position_Memory_4[USER_1];
    INT32 Position_Intermediate_In[USER_1];
    INT32 Position_Intermediate_Out[USER_1];

    INT32 Position_Memory_1[USER_2];
    INT32 Position_Memory_2[USER_2];
    INT32 Position_Memory_3[USER_2];
    INT32 Position_Memory_4[USER_2];
    INT32 Position_Intermediate_In[USER_2];
    INT32 Position_Intermediate_Out[USER_2];

    INT32 Position_Memory_1[USER_3];
    INT32 Position_Memory_2[USER_3];
    INT32 Position_Memory_3[USER_3];
    INT32 Position_Memory_4[USER_3];
    INT32 Position_Intermediate_In[USER_3];
    INT32 Position_Intermediate_Out[USER_3];

    INT32 Position_Memory_1[USER_4];
    INT32 Position_Memory_2[USER_4];
    INT32 Position_Memory_3[USER_4];
    INT32 Position_Memory_4[USER_4];
    INT32 Position_Intermediate_In[USER_4];
    INT32 Position_Intermediate_Out[USER_4];
    INT32 Position_Virtual_Limit_In;
    INT32 Position_Virtual_Limit_Out;
};
ACTUATOR_POSITIONS positions[ACTUATOR_COUNT];
```



ewellix.com

© Ewellix

All contents of this publication are the property of Ewellix, and may not be reproduced or given to third parties (even extracts) without permission. Although great care has been taken in the production of this catalog, Ewellix does not take any responsibility for damage or other loss resulting from omissions or typographical errors. The photo may differ slightly in appearance from the actual product. Due to continuous improvements being made in our products, the product's appearance and specifications are subject to change without notice.

PUB NUM TC-08032-DE-Juni 2021