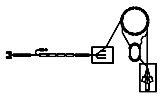


PPOS - Pressluft Positionierung

Gerätemodell und Softwareentwurf

G. Riehl, L. Hechler

GSI, Gesellschaft für Schwerionenforschung mbH
Planckstraße 1, D-64291 Darmstadt
Tel.: 06159/71-0

Issued: 4. Jul. 2018
Key: F-PPOS-02
Printed: July 9, 2018

Dieses Papier enthält die Beschreibung des Gerätemodells 'PPos - Pressluft Positionierung' und den Entwurf der Gerätesoftware für dieses Gerät.

Änderungsprotokoll			
Datum	GM-Version	Name	Kommentar
18. Jul. 97	–	G. Riehl	aufbauend auf 'Step' begonnen
8. Mai 01	–	G. Riehl	Erweiterung f. ESR Devices
6. Jul. 01	–	G. Riehl	Dokumentation zu Nodal und Spezialitäten
30. Okt. 01	–	M. Kühn	Überarbeitete und erweiterte TEX"=Version, die sowohl in PostScript als auch in HTML konvertiert werden kann.
1. Jul. 03	–	G. Riehl	PROP DRIVECNT implementiert
6. Jul. 18	–	L. Hechler	Aktualisierungen, u.a. EVTPOSS2, EVTPOSI2

Contents

List of Figures

Part I
Das Gerätemodell

1 Die Aufgabe des Gerätes

In der GSI gibt es an vielen Stellen Objekte, die über kürzere Distanzen möglichst genau bewegt werden müssen. Zur Lösung dieser Aufgabe werden bevorzugt Schrittmotoren eingesetzt, die aber den Nachteil geringer Geschwindigleit haben. Die hier beschriebene Preßluft getriebene Positionierung verspricht wesentlich höhere Fahrgeschwindigkeiten mit der Möglichkeit die Positionierungen eventgesteuert innerhalb eines SIS-Zyklus durchzuführen.

Analog zu den Schrittmotor-Antrieben gilt auch für diesen Linearantrieb:

- Alle Bewegungen nach rechts, nach oben oder in Strahlrichtung sind positiv.
- Alle Bewegungen nach links, nach unten oder gegen die Strahlrichtung sind negativ.

Bei Drehbewegungen gilt :

- Alle Bewegungen im Uhrzeigersinn sind positiv.
- Alle Bewegungen gegen den Uhrzeigersinn sind negativ.

Bei diesem Gerät werden innerhalb der VME-Ebene keine Geräte-Paare o.ä. berücksichtigt, es gibt keine 'Super-Usrs'.

2 Die Hardware des Gerätes

Pressluft-Antrieb sowie die Positions-Messung werden von einer SPS gesteuert. Diese SPS steht über eine serielle Schnittstelle mit einer speziellen Interface-Karte , dem MIL-Bus und der SE in Verbindung. Zugriff auf den Antrieb erfolgt ausschliesslich über die SPS und die von dieser zur Verfügung gestellten Kommandos.
Ausnahmen: Druck-Kontrolle, Endlagenschalter, ..., diese können über die IFK und deren Status ausgelesen werden.

3 Die Schnittstelle zum Gerät

3.1 Funktionscodes der Interfacekarte

Die für die Geräteansteuerung definierten Funktionscodes sind in den folgenden zwei Tabellen aufgelistet. Als Modus ist angegeben, ob Daten von der Interfacekarte gelesen werden, ob Daten zu der Interfacekarte geschrieben werden, oder ob nur eine Funktion ausgeführt wird. Die erste Tabelle beschreibt die MIL-Codes, mit der Daten zur/von der SPS geschrieben/gelesen werden können. Die zweite Tabelle beschreibt die (ASCCI-) Codes, mit der die SPS selbst angesteuert wird.

Funktionscode		Modus	Bedeutung
Name	Hex		
IFB_RdStat_Int	C9	Lesen	Ifk-Status lesen
IFB_Rdstat	C0	Lesen	Gerätestatus lesen
IFB_Soll_1	06	Schreiben	Ascii-Charakter zur SPS schreiben
IFB_Ist_1	81	Lesen	Ascii-Charakter von SPS lesen

IFB_Rdstat

Status des Antriebes lesen.
Das vom Gerät gelieferte Statuswort hat folgenden Aufbau :

Bit	Name	Bedeutung	
		High (1)	Low (0)
0	Endlage rechts/oben/außen	erreicht	nicht erreicht
1	Endlage links/unten/innen	erreicht	nicht erreicht
2	Druckwächter Öffner	Druck 0	Druck ok.
3	Druckwächter Schließer	Druck ok	Druck 0
4	Kabelbruch	ok	defekt
5	reserved		
⋮	⋮		
15	reserved		

3.2 Hardware des Linearantriebs

siehe dazu Seite 7 !

3.3 Interlock-Interrupt

Diese Interruptleitung wird zur ... benutzt.

3.4 Data Request (DRQ) Interrupts

Diese Interruptleitung wird zur ... benutzt.

3.5 Data Ready (DRD) Interrupts

Diese Interruptleitung wird zur ... benutzt.

3.6 Umfang eines logischen Gerätes

Ein Linear-Antrieb ist ein logisches Gerät mit einer Interface-Karte.

3.7 Definition der Bits des Hardwarestatus

Das Gerät liefert die oben beschriebenen 5 Bit Statusinformation.
Zur Bedeutung der Bits des Status siehe Abschnitt 3.1 auf Seite 8.
Das zur VAX gelieferte Statuswort wird auf der SE aus dem Gerätestatus und Software Statii generiert.
Die von der Hardware gelieferten zwei Bits des Druckwächters werden zu einem Bit zusammengefasst.
Die Bits 0...7 sind die systemweiten generierten Softwarestatusbits (in engl. derived status bits).
Die Statusbits im Einzelnen sind in der folgenden Tabelle zusammengefaßt.

Bit	Name	Bedeutung	
		High (1)	Low (0)
0	Power	on	off
1	Remote/Local	Remote	Local
2	reserved		
3	reserved		
4	Emergency	no	yes
5	Interlock	no	yes
6	HW Error	no	yes
7	SW Error	no	yes
8	Endlage rechts/oben/außen	erreicht	nicht erreicht
9	Endlage links/unten/innen	erreicht	nicht erreicht
10	Druckwächter	Druck ok	Druck 0
11	reserved		
12	Kabelbruch	ok	defekt
13	Aktion Motor	steht	fährt
14...31	reserved		

4 Die Bedienung des Gerätes

4.1 Aufgaben im Normalbetrieb

4.1.1 Antrieb fahren

Der Antrieb kann auf Position und auf Endlage (außen oder innen) gefahren werden. Die Überwachung eines fahrenden Antriebs (lesen der aktuellen Position), wird von der SE übernommen, d.h. die SE wartet bis ein Fahrkommando abgeschlossen ist.

4.1.2 Positionsüberwachung

Für den Antrieb muss die aktuelle Position periodisch (Periode 1 s) überwacht werden.

4.1.3 Gerätestatusüberwachung

Der Status wird periodisch übewrwacht.

4.1.4 Einschalten

Die Steuerung kann nicht über das Kontrollsystem eingeschaltet werden.

4.1.5 Ausschalten

Die Steuerung kann nicht über das Kontrollsystem ausgeschaltet werden.

4.2 Genauigkeitsanforderungen

Alle Genauigkeitsanforderungen werden durch die verwendete Hardware erfüllt.

4.3 Zeitkritische Anforderungen

...

4.4 Einordnung in das Timing

Das Gerät nimmt an der PPM teil und zwar sollen 2 Events berücksichtigt werden:

- Event 1 zum Anfahren von Sollposition 1
- Event 2 zum Anfahren von Sollposition 2

Die Konnektierungen sind in der folgenden Tabelle zusammengefaßt.

Aktion	EQM	Event
Position 1 anfahren	EvtMove1_EQM	Evt_UDet_In (162)
Position 2 anfahren	EvtMove2_EQM	Evt_UDet_Out (163)

Table 2: Standard-Eventkonnektierungen für den Linear-Antrieb

4.5 Festlegung von Startwerten

4.5.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt :

- Reset der Ifk und der SPS, Wiederherstellung der Kommunikation mit der SPS, SPS auf Rechner schalten und Referenzfahrt durchführen
- Das Status_EQM wird als periodischer Auftrag (Periode 1 s) konnektiert
- Alle Sollwerte werden auf 0 gesetzt.
- Die Stati werden gelesen.
- Die Ist-Positionen werden gelesen.
- Die Standard-Eventkonnektierungen werden gesetzt (siehe Tabelle 2 auf Seite 10).

- Alle Geräte werden für alle Beschleuniger auf inaktiv gesetzt.
- Die SE wird in den Eventmode-Betrieb geschaltet (nur bei Kaltstart der SE).

4.5.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Reset der Ifk und der SPS, Wiederherstellung der Kommunikation mit der SPS, SPS auf Rechner schalten und Referenzfahrt durchführen
- Das Status_EQM wird als periodischer Auftrag (Periode 1 s) konnektiert.
- Die Stati werden gelesen.
- Die Ist-Positionen werden gelesen.

4.6 Handbetrieb

??

4.7 Ableitung des Hardwarefehler-Bits aus dem Gerätestatus

Ein Hardwarefehler (angezeigt im Hardwarefehler-Bit des Status) liegt vor, wenn eines der folgenden Bits des Hardwarestatus *nicht* den angegebenen Wert (nicht OK) anzeigt.

Bit	Name	Wert
8,9	Beide Endlagen aktiv	1
10,11	Druckwächter	1,0
12	Kabelbruch	0

4.8 Verhalten bei Störungen

4.8.1 Geräteinterlock

Die Steuerung liefert keinen Geräteinterlock.

4.8.2 Event-Sequenzfehler

Event-Sequenzfehler sind nicht möglich.

4.8.3 Event-Overrun

Event-Overrunfehler werden gemeldet.

4.8.4 Emergency-Event

Emergency-Events müssen nicht berücksichtigt werden.

4.8.5 Ausfall der Kommunikation EC – Gerät

Der Ausfall der Kommunikation zwischen EC und Gerät führt zu Timeouts, Checksum-Fehlern oder ähnlichen Fehlermeldungen. Über die Protokollstatistik des EC kann ermittelt werden *wie oft welche* Fehler aufgetreten sind (Menüpunkt 30 im NODAL-Programm COMP_TEST). Läßt sich das Kommunikationsproblem nicht durch einen Reset beseitigen, so sind in den meisten Fällen ausführlichere Untersuchungen notwendig.

4.9 Bedienungsfehler vom Operating

Bei jedem Fahrkommando wird auf der EC-Ebene überprüft, ob der Antrieb schon/noch fährt. Ist dies der Fall, so wird das Kommando mit einer entsprechenden Fehlermeldung abgewiesen.

4.10 Sonstige Anforderungen

Es ist möglich (aber unwahrscheinlich), daß sich der Status eines Antriebs ändert, ohne daß vom Kontrollsystem aus eine Aktion ausgelöst wurde (z.B. Manipulation *von Hand*, Kabeldefekt am Endschalter, ...). Durch die *normale* Statusüberwachung werden solche Änderungen erfaßt.

5 Die Repräsentation des Gerätes

5.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **PPOS**.
Die Gerätemodellnummer ist 54_{dez}.

5.2 Die Master-Properties

Master-Properties							
Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
POWER	R/W	0	–	1	BitSet16	1	0
STATUS	R	0	–	1	BitSet32	1	0
INIT	N	0	–	0	–	–	–
RESET	N	0	–	0	–	–	–
VERSION	RA	0	–	48	BitSet8	1	0
INFOSTAT	RA	0	–	25	BitSet32	1	0
CMDPOSS	R/W	0	–	1	Integer16	m	-4
CMDPOSI	R	1	Integer16	1	Integer16	m	-4
POSABSI	R	1	Integer16	1	Integer16	m	-4
CONSTANT	RA	0	–	3	RealF	1	0
ENDLAGE	W	0	–	1	BitSet16	1	0
SPEED	R/W	0	–	1	Integer16	m	0
SPSCMD	RA	81	BitSet8	81	BitSet8	1	0
SPSSTR	RA	1	String	1	String	1	0
SPSDATA	RA	0	–	2593	BitSet8	1	0

Master-Properties							
Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
DPRDATA	RA	0	–	26	BitSet16	1	0
DRIVECNT	R/W	0	–	1	Integer32	1	0

5.2.1 POWER

Bedeutung: Gibt an, ob der Leistungsteil des Gerätes ein- oder ausgeschaltet ist. Die Power des Gerätes kann NICHT geschaltet werden.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Null heißt, das Gerät ist eingeschaltet. Eins heißt, das Gerät ist ausgeschaltet.

5.2.2 STATUS

Bedeutung: Auslesen des 32-Bit-Gerätestatus.

Parameter: Keine.

Daten: Das 32-Bit-Statuswort. Die Bits entsprechen den Statusbits, wie sie in Abschnitt 3.7 auf Seite 9 erklärt sind.

5.2.3 INIT

Bedeutung: Initialisierung des Gerätes (Kaltstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.5.1 auf Seite 10.

Parameter: Keine.

Daten: Keine.

5.2.4 RESET

Bedeutung: Reset des Gerätes (Warmstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.5.2 auf Seite 11.

Parameter: Keine.

Daten: Keine.

5.2.5 VERSION

Bedeutung: Lesen der Versionskennung der Gerätesoftware.

Parameter: Keine.

Daten: Versionskennung als ASCII-String, pro Datum ein ASCII-Zeichen.

Bytes	Inhalt
1...12	Version der USRs
13...24	Version der EQMs
25...36	Version des Standard-MIL-Treibers
37...48	Variante der EQMs

5.2.6 INFOSTAT

Bedeutung: Diese Property liefert einige wichtige Geräteinformationen in einem Zugriff. Die Informationen werden direkt aus dem Dualport-RAM gelesen, also ohne den expliziten Aufruf eines EQMs, und sind daher in der Abarbeitung nicht abhängig von Kommandoevents.

Parameter: Keine.

Daten: Die 25 Langworte enthalten im Einzelnen:

1: Gerätestatus (wie in der Property STATUS)

2: Gibt in den oberen 16 Bits an, welcher virtuelle Beschleuniger aktiv gesetzt ist (ein Bit pro Beschleuniger). Das niederwertigste Bit (Bit 16) gibt den Beschleuniger 15 an, das Bit 31 den Beschleuniger 0. Die unteren 16 Bit sind nicht verwendet. Dabei bedeutet Null, daß der Beschleuniger inaktiv ist und Eins, daß der Beschleuniger aktiv ist.

3: Master-Fehler. Hier ist derjenige Master-Gerätefehlercode mit dem schwersten Fehlergrad eingetragen. Bei mehreren Fehlern mit dem gleichen Fehlergrad wird der erste eingetragen, der gefunden wurde.

4: Slave Fehler für virtuellen Beschleuniger 0. Entsprechend dem Master-Fehler wird hier der nach dem Fehlergrad schwerste Slave-Gerätefehlercode für den Beschleuniger 0 eingetragen.

5: Entsprechend Punkt 4, aber für virtuellen Beschleuniger 1.

⋮

19: Entsprechend Punkt 4, aber für virtuellen Beschleuniger 15.

20: EC-Mode. In den oberen 16 Bit des Langwortes steht der von der Gerätesoftware eingestellte Default-EC-Mode, in den unteren 16 Bit der aktuelle EC-Mode. Folgende Modi sind definiert:

- **0:** *not set*
- **1:** *Preset_Command* Der ECM hat das Umschalten in Command-Mode vorbereitet aber noch nicht beendet.
- **2:** *Command* Der ECM läuft im Command-Mode.

3: *Preset_Event* Der ECM hat das Umschalten in Event-Mode vorbereitet aber noch nicht beendet.

4: *Event* Der ECM läuft im Event-Mode.

21: EC-Performance-Mode. In den oberen 16 Bit des Langwortes steht der von der Gerätesoftware eingestellte Default-Performance-Mode, in den unteren 16 Bit der aktuelle Performance-Mode. Folgende Modi sind definiert:

0: *not set*

1: *Display* Der ECM läuft im Display-Mode.

2: *Preset_Turbo* Der ECM hat das Umschalten in den Turbo-Mode vorbereitet aber noch nicht beendet.

3: *Turbo* Der ECM läuft im Turbo-Mode.

22: HW_Warning_Maske. Die 32 Bits geben an aus welchen Bits im Gerätestatus das HW-Warning-Bit im Status abgeleitet wird.

23: Reserviert für Erweiterungen.

⋮

25: Reserviert für Erweiterungen.

5.2.7 CMDPOSS

Bedeutung: Bringt einen Antrieb per Kommando auf eine absolute Soll-Position.

Parameter: keine

Daten: Positionsangabe in 1/100 mm.

5.2.8 CMDPOSI

Bedeutung: Liefert von einem Antrieb die angefahrenen, absolute Ist-Position nach dem letzten Positions-Kommando. Die aktuelle Position des Antriebes kann mit der Property POSABSI ermittelt werden.

Parameter: 1 dummy siehe (siehe Seite 27)

Daten: Positionsangabe in 1/100 mm.

5.2.9 POSABSI

Bedeutung: Liefert von einem Antrieb die aktuelle, absolute Ist-Position.

Parameter: 1 dummy siehe (siehe Seite 27)

Daten: Positionsangabe in 1/100 mm.

5.2.10 ENDLAGE

Bedeutung: Der Antrieb wird an die angegebene Endlage gefahren.

Parameter: keine

Daten: Folgende Bedeutungen sind vereinbart:

0: Endlage außen,
1: Endlage innen.

5.2.11 SPEED

Bedeutung: Setzen/Lesen der Fahrgeschwindigkeit. Erlaubt sind Werte zwischen 1 und 100 als Prozentwerte der Maximalgeschwindigkeit.

Parameter: Keine

Daten: 1% - 100% der Maximalgeschwindigkeit

5.2.12 CONSTANT

Bedeutung: Lesen der Geräte-Konstanten.

Parameter: Keine

Daten:

PosFact: Umrechnungsfaktor Abs. Position (1/100 mm) in Geräteposition (0=aussen, ca. 15000=150mm innen): DevPos = (AbsPos-Endl.Aussen) * PosFact

EndlAussen: Absolute Endlage aussen in 1/100 mm

EndlInnen: Absolute Endlage innen in 1/100 mm

5.2.13 SPSSTR

Bedeutung: Direktes Schreiben eines SPS-Kommandos zum Gerät und Lesen der SPS-Antwort

Parameter: SPS-Kommandostring

Daten: SPS-Antwortstring

5.2.14 SPSCMD

Bedeutung: Direktes Schreiben eines SPS-Kommandos zum Gerät und Lesen der SPS-Antwort

Parameter: SPS-Kommandostring, 81 Bytes

Daten: SPS-Antwortstring, 81 Bytes

5.2.15 SPSDATA

Bedeutung: Lesen der SPS-Kommunikation (Kommandos und Antworten)

Parameter: Keine

Daten: 2593 Bitset8,

1: Anzahl Kommandos/Antworten,

2-2593: max. 16 mal 81 Bytes KommandString plus 81 Bytes AntwortString

5.2.16 DPRDATA

Bedeutung: Lesen aktueller DPR-Daten

Parameter: Keine

Daten: 26 Bitset16

1,2 m_sts : M_Status_Type;
3 int_sts : internal_status_type;
4 const_actual : word;
5,6 PosFact : long;
7,8 EndlAussen : long;
9 Soll_pos : word;
10 Soll_speed : word;
11 Ist_pos : word;
12 Ist_speed : word;
13 Ist_time : word;
14 EvtPoss_1 : word;
15 EvtPoss_2 : word;
16 EvtSped_1 : word;
17 EvtSped_2 : word;
18 EvtPosi_1 : word;
19 EvtPosi_2 : word;
20 EvtTime_1 : word;
21 EvtTime_2 : word;
22, 23 EvtStat_1 : long;
24, 25 EvtStat_2 : long;
26 ActIst_Pos : word;

5.2.17 DRIVECNT

Bedeutung: Liefert oder setzt von einem Antrieb die gezählten Fahrten >= 20 mm.

Parameter: -

Daten: 1 Integer32, Zählerwert

5.3 Die Slave-Properties

Slave Properties							
Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
ACTIV	R/W	0	–	1	BitSet16	1	0
COPYSET	W	0	–	1	BitSet16	1	0
EQMERROR	RA	217	BitSet32	348	BitSet32	1	0
EVTPOSS	R/W	1	Integer16	1	Integer16	m	-4
EVTPOSI	R	1	Integer16	1	Integer16	m	-4
EVTPOSS2	R/W	0	–	2	Integer16	m	-4
EVTPOSI2	R	0	–	2	Integer16	m	-4

5.3.1 ACTIV

Bedeutung: Gibt an, ob das Gerät für den zugehörigen virtuellen Beschleuniger an der Puls-zu-Puls-Modulation (kurz PPM) teilnehmen soll bzw. teilnimmt.

Auch nichtgepulste Geräte müssen diese *obligatorische* Property haben. Ungepulste Geräte liefern beim Versuch ACTIV zu schreiben einen entsprechenden Fehler und beim Versuch ACTIV zu lesen eine Information oder eine Warnung zurück.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Null heißt, das Gerät nimmt für den zugeordneten Beschleuniger *nicht* an der PPM teil bzw. soll *nicht* an der PPM teilnehmen. Eins heißt, das Gerät nimmt für den zugeordneten Beschleuniger an der PPM teil bzw. soll an der PPM teilnehmen.

5.3.2 COPYSET

Bedeutung: Kopiert alle Geräteeinstellungen (Sollwerte) eines virtuellen (‘fremden’) Beschleunigers in den zugehörigen (‘eigenen’) Beschleuniger. Die Property ist bei den Schrittmotoren wirkungslos (siehe oben) !

Parameter: Keine.

Daten: Nummer des virtuellen (‘fremden’) Beschleunigers, von dem die Einstellungen (Sollwerte) kopiert werden sollen.

5.3.3 EQMERROR

Bedeutung: Fehlermeldungen der auf der SE installierten Gerätesoftware. Es werden die aktuellen Fehlermeldungen sowohl für die Masterfehler als auch für die Slavefehler der Geräteebene geliefert. Dazu wird auch der Inhalt des Fehlerpuffers zurückgegeben, in dem die letzten aufgetretenen Fehler abgespeichert wurden.

Parameter: Keine.

Daten: Die Anzahl der Fehlermeldungen sei bezeichnet durch:

- m Zahl der Master-Fehlermeldungen
- s Zahl der Slave-Fehlermeldungen
- b Größe des Fehlerpuffers

Weiterhin soll gelten:

$\boldsymbol{l} = m + s$

$\boldsymbol{t} = m + s + b$

Die Daten im Einzelnen:

1 :	In den unteren beiden Bytes sind die Anzahl der Master-Fehlermeldungen m und die Anzahl der Slave-Fehlermeldungen s angegeben:

0	0	s	m

2 :	erste Master-Fehlermeldung
	⋮
$m + 1$:	letzte Master-Fehlermeldung
$m + 2$:	erste Slave-Fehlermeldung
	⋮
$l + 1$:	letzte Slave-Fehlermeldung
$l + 2$:	Länge b des Fehlerpuffers
$l + 3$:	Zahl der Einträge im Fehlerpuffer
$l + 4$:	Index des ersten freien Platzes im Fehlerpuffer (der Fehlerpuffer ist ein Ringpuffer)
$l + 5$:	Erster Speicherplatz im Fehlerpuffer
	⋮
$t + 4$:	Letzter Speicherplatz im Fehlerpuffer

5.3.4 EVTPOSS

Bedeutung: Setzen/Lesen der Sollpositionen der konnektierten Events. Zur Zeit werden 2 Events berücksichtigt und somit werden zwei Sollpositionen gesetzt. Die Nummer der Sollposition wird als Parameter übergeben.

Parameter: 1 Word, Nummer der Sollposition, derzeit 1 oder 2.

Daten: 1 Integer16, Sollposition in 1/100 mm

5.3.5 EVTPOSI

Bedeutung: Lesen der Istpositionen der konnektierten Events. Zur Zeit werden 2 Events berücksichtigt und somit können zwei Istpositionen gelesen werden. Die Nummer der Position wird als Parameter übergeben. Position entspricht der angefahrenen Position nach dem letzten empfangenen Event. Die aktuelle Position des Antriebes kann mit der Property POSABSI ermittelt werden.

Parameter: 1 Integer16, Nummer der Position, derzeit 1 oder 2.

Daten: 1 Integer16, Position in 1/100 mm-Einheiten

5.3.6 EVTPOSS2

Bedeutung: Setzen/Lesen der Sollpositionen der konnektierten Events. Zur Zeit werden 2 Events berücksichtigt und somit werden zwei Sollpositionen gesetzt.

Parameter: Keine.

Daten: 2 Integer16, Sollposition 1 und 2 in 1/100 mm

5.3.7 EVTPOSI2

Bedeutung: Lesen der Istpositionen der konnektierten Events. Zur Zeit werden 2 Events berücksichtigt und somit können zwei Istpositionen gelesen werden. Die Position entspricht der angefahrenen Position nach dem letzten empfangenen Event. Die aktuelle Position des Antriebes kann mit der Property POSABSI ermittelt werden.

Parameter: Keine.

Daten: 2 Integer16, Istposition 1 und 2 in 1/100 mm-Einheiten

Part II
Der Entwurf der Software

6 Softwareentwurf

Keine erwähnenswerten Besonderheiten.

7 Lokale Datenbasis

7.1 Tabelle der Konstanten

Elemente in der Konstantentabelle der lokalen Datenbasis. Die Elemente haben in der Reihenfolge folgende Bedeutung:

1: (dev_const1) Konstante zum Umrechnen von Koordinatensystemen bzw. zur Einbeziehung von Übersetzungen.

2: (dev_const2) Konstante zum Umrechnen von Koordinatensystemen (bisher nicht benutzt)

8 Dualport RAM

In den Datenstrukturen des Dualport RAM sind keine erwähnenswerten Besonderheiten enthalten.

9 USRs - User Service Routinen

9.1 Obligatorische USRs

Eine Beschreibung der USRs findet nur statt, wenn die Funktion vom Standard abweicht und relevante Besonderheiten aufweist.

9.1.1 N_Init

9.1.2 N_Reset

9.1.3 R_Status

9.1.4 R_Power

9.1.5 W_Power

9.1.6 R_Active

9.1.7 W_Active

9.1.8 W_CopySet

9.1.9 R_EQMErr

9.1.10 R_Version

9.1.11 R_InfoStat

9.2 Gerätespezifische USRs

Zuzüglich der obligatorischen USRs werden für die Steuerung des Schrittgerätes gerätespezifische USRs zur Umsetzung der gerätespezifischen Properties benötigt. Eine Beschreibung dieser USRs findet hier jedoch nur statt, wenn die Funktion vom Standard abweicht oder neben der einfachen Umsetzung der Propertie relevante Besonderheiten aufweist.

9.3 Globale Routinen

9.3.1 Get_Device_Parameter

Holt die Konfigurationsinformationen über ein Gerät (logische Nummer, Pointer auf Dualport RAM der entsprechenden SE) aus der Konfigurationstabelle des MOPS.

10 EQMs - Equipment Module

10.1 Interne Zustände

10.1.1 Bedeutung der internen Zustände

Für die Gerätesoftware sind folgende interne Zustände definiert:

`not_set`	Initzustand. Dieser Zustand sollte nie auftreten.
`emergency`	Ein Emergency-Event wurde empfangen. Dieser Zustand darf nur durch Rücksetzen vom Operating verlassen werden.
`error`	Während der Abarbeitung eines EQMs wurde ein Fehler erkannt.
`ready`	Das Gerät ist bereit für Aktionen. Ausgangszustand am Beginn eines virtuellen Beschleunigers.
`busy`	Das Gerät bearbeitet den letzten Fahrbefehl
`ramp`	Das Gerät arbeitet eine Positionsrampe ab

10.1.2 Übergänge zwischen den Zuständen

Erläuterung, welche Übergänge zwischen den internen Zuständen vorgesehen sind und wodurch sie ausgelöst werden sollen.
Die Zustände und die Übergänge zwischen denselben sind in Tabelle 5 zusammengefasst. Die Legende zu diesen Tabellen ist in Tabelle 6 zu finden.

Tabelle der Zustandsübergänge				
von↓	nach→	emergency	error	ready
emergency	U:	--	--	RESET
	B:	--	--	--
	A:	--	--	Reset_EQM
error	U:	Evt_Emerg	--	RESET
	B:	--	--	--
	A:	Emerg_EQM	--	Reset_EQM,
ready	U:	Evt_Emerg	--	--
	B:	--	--	--
	A:	Emerg_EQM	--	--

Table 5: Zustandsübergangsdiagramm

Legende

- Die Priorität der Zustände (höchste Priorität zuerst): emergency, interlock, local, power_off und power_seq, error, ready.

  Liegen mehrere Bedingungen für verschiedene Zustände gleichzeitig vor (z. B. Netz aus und Gerät auf Handbetrieb), muss der jeweils wichtigste Zustand eingenommen werden.

- U: Auslösende Ursache.

Evt_Emerg	Pulszentrale verschickte Emergency-Event.
RESET	Reset wird per Kommando oder Knöpfchendrücken ausgelöst.

- B: Abzuprüfende Bedingung.

R	Remotebit des Status steht auf Remote.
r	Remotebit des Status steht auf Local.
P	Powerbit des Status steht auf Power on.
p	Powerbit des Status steht auf Power off.

- A: Ausführende Stelle des Zustandübergangs.

Status lesen (period.)	Beim periodischen (oder regelmäßigen) Lesen des Status.
..._EQM	Innerhalb des EQMs ..._EQM.

Table 6: Legende zu den Zustandsübergangsdiagrammen

10.2 Eventkonnektierte EQMs

10.2.1 EvtMove1_EQM

Event: EVT_UDet_In (162dez)

Aktion: Der Antrieb wird auf die Sollposition 1 gefahren, neue Istposition bestimmt und Zeit zwischen Start der Bewegung und Abschluß der Positionsmessung bestimmt.

10.2.2 EvtMove2_EQM

Event: EVT_UDet_Out (163dez)

Aktion: Der Antrieb wird auf die Sollposition 2 gefahren, neue Istposition bestimmt und Zeit zwischen Start der Bewegung und Abschluß der Positionsmessung bestimmt.

10.2.3 Emerg_EQM

Event: Evt_Emergency.

Aktion: Internen Zustand auf 'Emergency' setzen. Keine gerätespezifische Aktion ausführen.

10.3 Periodisch konnektierte EQMs

10.3.1 Status_EQM

Zeit: $1s$

Anzahl: Unendlich.

Aktion: Liest im sogenantem Blockmode den Status und die Position aller an einem SDμPangeschlossenen Antriebe. Dieses EQM wird pro angeschlossem SDμPeinmal konnektiert.

10.3.2 Update_Config_EQM

Zeit: $60s$

Anzahl: Unendlich.

Aktion: Aktualisieren der Geräteverfügbarkeit: Es wird versucht, von möglichen Geräteadressen den Status zu lesen. Erfolgt eine Reaktion, wird das Gerät als 'online' geführt.

10.4 An externe Interrupts konnektierte EQMs

10.4.1 Interlock_EQM

Interrupt: Summen-Interlock.

Aktion: Keine Aktion.

10.4.2 DRD_EQM

Interrupt: Data Ready Interrupt.

Aktion: Keine Aktion.

10.4.3 DRQ_EQM

Interrupt: Data Request Interrupt.

Aktion: Keine Aktion.

10.5 Kommandogetriggerte EQMs

10.5.1 Dev_Init_EQM

10.5.2 Dev_Reset_EQM

10.5.3 Status_EQM

10.5.4 Active_EQM

10.5.5 Power_EQM

10.5.6 Move_EQM

Parameter: Das EQM benötigt 1 Parameter.

1. Fahrmodus:

   move_pos (1): Antrieb auf absolute Position fahren.
   move_innen (5): Antrieb auf Endlage innen/aussen fahren.

Daten: 1 Datum, abhängig vom Fahrmodus:

move_pos (1): absolute Position 1/100 mm oder 0=Endlage aussen, > 0 Endlage innen

Aktion: Veranlaßt ein Fahren des Antriebs im angegebenen Modus.

10.5.7 Posi_EQM

Parameter: keine

Daten: 1 word

Ist-Position: absolute Position in 1/100 mm

Aktion: Misst die aktuelle Ist-Position des Antriebs

10.6 EQMs für die Diagnose vor Ort

10.6.1 Display_DPR_EQM

Parameter: Das EQM benötigt 2 Parameter.

1. virtueller Beschleuniger (in Hex angeben)
2. logische Gerätenummer (in Hex angeben)

Daten: Keine.

Aktion: Zeigt am Bildschirm vor Ort die wichtigsten Daten aus dem DPRAM für das gewählte Gerät und den gewählten virtuellen Beschleuniger an.

10.6.2 Display_DevErr_EQM

Parameter: Das EQM benötigt 2 Parameter.

1. virtueller Beschleuniger (in Hex angeben)
2. logische Gerätenummer (in Hex angeben)

Daten: Keine.

Aktion: Zeigt am Bildschirm vor Ort die Error-Codes aus der aus der Datenstruktur im Dualport-RAM für das gewählte Gerät und den gewählten virt. Beschleuniger an.

10.7 Sonstige EQMs

10.7.1 Startup_EQM

Installiert die Event-EQM-Konnektierung für alle virtuellen Beschleuniger (siehe hierzu auch Abschnitt 4.4 auf Seite 10).

10.8 Globale Routinen

10.8.1 Read_and_Update_Status

Liest vom Gerät den Gerätestatus.

10.8.2 Do_Intr_Service_Prep

Macht im Wesentlichen eigentlich nichts, weil der Interlockinterrupt bei Schrittmotoren ohnehin nicht verwendet wird.

11 Varianten

Dieses Gerätemodell besitzt keine Varianten.

12 Besonderheiten

12.1 Der Response-Algorithmus

Um Rechen- und Transferzeiten auf den Computern der Operating-Ebene zu sparen, wird ein Algorithmus angewandt, der bei konnektierten Aufträgen nur bei Änderungen Daten liefert. Dazu ist prinzipiell das Problem zu lösen, dass eine USR kein *eigenes* RAM besitzt, in dem sie sich von Aufruf zu Aufruf etwas (z. B. die letzten Istwerte) *merken* könnte. Allerdings liegt während des gesamten Zeitraums einer bestehenden Konnektierung das ursprüngliche Auftragspaket (Parameter und Daten) im DPRAM des Ethernet-Controllers vor.
Der *Trick*, dass die USR sich von Aufruf zu Aufruf etwas merken kann, besteht nun darin, dass bei der Konnektierung der USR genau die Anzahl der zu erwartenden Daten als Parameter (mit beliebigen Werten, also Dummies) mitgeschickt werden. Dadurch ergibt sich für die USR die Möglichkeit im Empfangsparameterteil des Auftragspaketes die letzten aktuellen Daten zu *merken* und diese beim nächsten Aufruf mit den aktuellen zu vergleichen, siehe Abbildung 12.1 auf Seite 28.
Um diesen Algorithmus nutzen zu können, muss das Operatingprogramm eine, der USR-Definition entsprechende, Anzahl von Dummy-Parametern bei der Konnektierung mitschicken. Wenn der Vergleich zwischen alten und aktuellen Werten eine Differenz zeigt oder ein Fehler aufgetreten ist, werden die aktuellen Werte gemeldet. Werden keine Parameter mitgeschickt, werden immer die aktuellen Werte zurückgemeldet.

12.2 Die zweistufige Positionierung

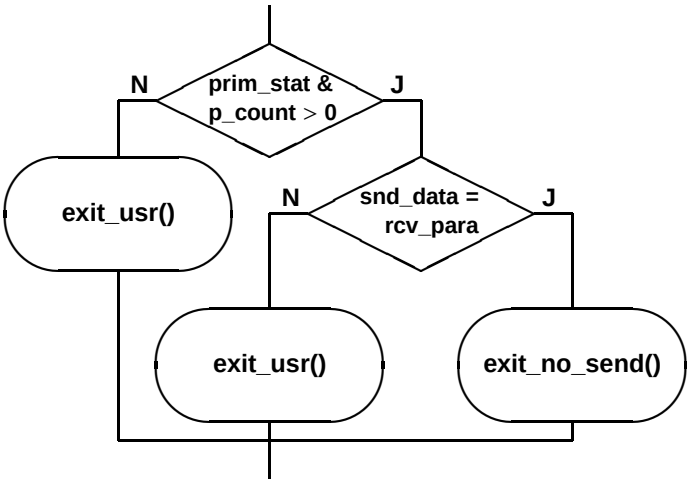
Um ein zu starkes Überschwingen des Antriebes bei der Positionierung zu verhindern, kann die Positionierung in zwei Stufen durchgeführt werden: Alle Fahraufträge mit Wegen > 10 mm werden in zwei Positionierungen aufgeteilt, wobei die erste Positionierung eine Position 5 mm VOR der endgültigen Position anfährt. Die endgültige und angezeigte Fahrzeit entspricht der Summe beider Fahrzeiten. Hierzu sind in den EQMs zwei Procedures implementiert, MoveS und TwoMoveS, die gegebenenfalls (bei Problemen) in der Software (EQMs) umgehängt werden müssen.

12.3 Positionsangaben

Es sind folgende Positionsangaben zu unterscheiden:

SPS-Position Die SPS setzt die äußerste Position (=referenzposition) = Position 0 und gibt alle weiteren Positionen nach innen in 1/100mm an

Property-Positionen Die Masseinheit ist hier ebenfalls 1/100mm. Die Position 0 ist dabei aber auf die Position des Detektors/Sonde auf Strahlachse definiert. Das Vorzeichen der Aussenposition hängt von der Einbaulage des Antriebes ab: In Strahlrichtung gesehen sind oben und rechts positiv, unten und links dagegen negative Positionen. Hier ist auf die sehr wichtige Rolle der Referenzfahrt hinzuweisen: Bei der Initialisierung des Antriebes führt die SPS eine Referenzfahrt auf den äußeren Endlagenschalter durch und setzt die erreichte Position danach auf 0. Wenn die Referenzfahrt nicht gelingt, ist die Position des Antriebes danach undefiniert !! Zur Umrechnung von SPS- zu Property-Position sind in der Datenbasis jedes Antriebes die inneren und äußeren Endlagen sowie ein Umrechnungsfaktor eingetragen.


N
prim_stat &
p_count > 0
J
exit_usr()
N
snd_data =
rcv_para
J
exit_usr()
exit_no_send()

Aktuelle Position: Dies entspricht der aktuellen (zuletzt gemessenen) Position des Antriebes und ist unabhängig davon, wie (Kommando oder Event) der Antrieb dorthin gekommen ist.

Master-Position: Dies gibt die Soll- und erreichte Ist-Position des Antriebes nach dem letzten Positionier-Kommando (manuelles Kommando vom Operating) an.

Event-Position: Dies gibt die Soll- und erreichte Ist-Position des Antriebes nach dem letzten Event-getriggerten Positionierung (derzeit Events 162dez und 163dez) an.

12.4 Positionsgenauigkeit

In der SPS sind die zulässigen Abweichungen zwischen Soll- und Ist-Position definiert, mit denen jede Positionierung durchgeführt wird. Die SPS beendet die Positionierung a) nach Erreichen der Soll-Position innerhalb der vorgegebenen Abweichung, b) nach Ablauf einer max. Positionierzeit, dann mit Fehler 'Position nicht erreicht' (Fehlertext von SE/GuP). Im Nodal-Programm wird a) die Fehlermeldung sichtbar, zum anderen ist als Positionierzeit eine Zeit 4 sek zu erkennen. Die in der Praxis erreichte Genauigkeit liegt bei ca. 0.15 mm (also im Vergleich zu Schrittmotoren ungenau !), eine Abweichung > 0.20 mm kommt vor und führt zu Fehlern.

12.5 Positionsgrenzen

Im allgemeinen werden von den Usrs alle Fahraufträge auf die in der datenbasis festgelegten Grenzen (innere und äußere Endlagen) begrenzt, siehe Property CMDPOSS. Zu Testzwecken und unter Vorbehalt verantwortlichen Umganges (!) gibt es auch die Property TSTPOSS, die KEINE überprüfung der Endlagen durchführt. ACHTUNG: Im Gegensatz zu den Schrittmotoren können bei PPOS die Endlagenschalter die Fahrt nicht unterbrechen !! UND: Da die SPS keine Positionsangaben < 0 entgegennimmt und diese Position auf die äußere Endlage definiert wird, kann der Antrieb die äußere Endlage trotzdem nicht überfahren. (Ausnahme: Referenzfahrt, dabei fährt die SPS den Antrieb ca. 10 mm über die Endlage hinaus.)

12.6 SPS Diagnosepuffer

Zwecks Kontrolle und Diagnose der V24 Kommunikation zwischen SE und SPS wird ein Kommunikations-Ringbuffer im DPR gehalten. In diesem stehen die letzten 32 Schreib-/Lese-Dialoge von SE/SPS

12.7 SPS Terminal

Zwecks Tests lassen sich einzelne SPS-Kommandos (ASCII-Strings) an die SPS schicken. Dazu sind sowohl in Gup/SE als auch im NODAL-Programm entsprechende Funktionen implementiert.

13 Nodal Programm

Nachfolgend eineige Anmerkungen zum Nodal-Programm PPOSINFO

13.1 1. Statuszeile

Nach dem Titel ”Positionierbarer Pressluftantrieb” stehen in der ersten Statuszeile

DEVICE Die Nomenklatur des angesteuerten Antriebes.

VACC Die Nummer des eingestellten virtuellen Beschleunigers.

ENDLAGEN Die äußere und innere Endlage des Antriebes. Zusätzlich wird in Klammern der Umrechnungsfaktor von SPS zu Property-Position angegeben, derzeit 1 oder -1.

13.2 Geräte-Status

Der Geräte-Status wird in 2 bz. 3 Formaten angegeben:

HEX/BINÄR Das 32-Bit Statuswort wird in Hexadezimaler und Binärer Notation angezeigt.

INTERPRETATIV Die relevanten Bits des Statuswortes werden entsprechend ihrer Bedeutung in Textform angegeben. Bei Bits, die eine Störung anzeigen, wird der Text blinkend invertiert geschrieben.

13.3 Geräte-Positionen

Die Geräte-Positionen werden alle in Einheiten von 1/100 mm angegeben.

Act-Ist Die aktuelle, zuletzt gemessenen Ist-Position des Antriebes. Wird bei jedem Refresh des Programm-Displays neu gemessen.

CommandPos Die letzte mit einem Kommando (Operating..) Position des Antriebes. Dabei werden geforderte Soll-Position, erreichte Ist-Position und die benötigte Fahrzeit in msec angegeben.

EvtPos1 Die letzte mit dem Event 1 (derzeit UDet_in, 162dez) Position des Antriebes. Dabei werden geforderte Soll-Position, erreichte Ist-Position und die benötigte Fahrzeit in msec angegeben.

EvtPos2 Die letzte mit dem Event 2 (derzeit UDet_out, 163dez) Position des Antriebes. Dabei werden geforderte Soll-Position, erreichte Ist-Position und die benötigte Fahrzeit in msec angegeben.

13.4 Nodal-Menü

Zur Gerätebedienung werden im Nodal-Menü folgende Funktionen angeboten:

EXIT Ende des Nodal Programmes...

Cmd-POS Setzen und anfahren einer neuen Position per Kommando

Endl Anfahren (per Kommando) einer Endlage (innen oder aussen) des Antriebes

NOMEN Anwahl eines Antriebes (Nomenklatur)

Terminal SPS-Terminal, es lassen sich einzelne SPS-Kommandos (ASCII-Strings) an die SPS schicken, die Antworten werden gelesen und angezeigt.

SPSData Der Kommunikations-Ringbuffer der V24 Kommunikation zwischen SE und SPS wird gelesen und angezeigt.

Init Device Initialisierung des Antriebes. Dabei wird u.a. die Baudrate zwischen Interface-Karte und SPS synchronisiert, sowie eine Referenzfahrt durchgeführt. Der Antrieb sollte danach an der äußeren Endlage stehen, der Endlagenschalter aussen aktiv sein (ON)

EventPos Setzen einer neuen Event-Position. Diese Position wird beim nächsten Eintreffen des entsprechenden Events im angegebenen Beschleuniger angefahren.

EqmError Anzeige des EQM Errorbuffers der SE.

VrtAcc/Active Setzen des virtuellen Beschleunigers und des Aktiv Zustandes des Antriebes.

Refresh Refresh des Nodal-Displays. Dabei werden auch Status und Positionen des Antriebes neu gelesen.

Index

—H—

—I—

—K—

—L—

—M—

—N—

—O—

—P—

—R—

—S—

—T—

—U—

—V—

—W—

—Z—