



PLA - Preßluftantriebe

Gerätemodell und Softwareentwurf

P. Kainberger

Dieses Papier enthält die Beschreibung des Gerätemodells 'PLA - Preßluftantriebe' und den Entwurf der Gerätesoftware für dieses Gerät.

An den Beschleunigerkomponenten und Strahlführungen werden Preßluftantriebe benutzt, um Strahldiagnoseelemente (Cups, Profilgitter, Leuchttargets, ...) in den Strahlweg einzufahren.

Änderungsprotokoll

Datum	GM-Version	Name	Kommentar
09. Sep. 94	PLA.07	P. Kainberger	Basisversion
21. Sep. 94	PLA.07	P. Kainberger	diverse Ergänzungen
30. Nov. 06.	PLA	K. Herlo	Portieren dieser Dokumentation nach Linux.
18. Dez. 09		U. Krause	Angepasst an Steuerung mit ohne SD μ P

Inhaltsverzeichnis

I	Das Gerätemodell	5
1	Die Aufgabe des Gerätes	5
2	Die Hardware des Gerätes	5
3	Die Schnittstelle zum Gerät	5
3.1	Funktionscodes der Interfacekarte	5
3.2	Aufbau der Kommunikation mit der Steuerelektronik	6
3.2.1	Datenformat Schreibzugriff	6
3.2.2	Lesezugriff	7
3.3	Interlock Interrupt	8
3.4	Data Request (DRQ) Interrupts	8
3.5	Data Ready (DRD) Interrupts	8
3.6	Umfang eines logischen Gerätes	8
3.7	Definition der Bits des Hardwarestatus	8
3.8	Konfigurationsabfrage	9
4	Die Bedienung des Gerätes	9
4.1	Aufgaben im Normalbetrieb	9
4.1.1	Antrieb Ein/Ausfahren	9
4.1.2	Verriegelung eines Antriebs	9
4.2	Zeitkritische Anforderungen	10
4.3	Einordnung in das Timing	10
4.4	Festlegung von Startwerten	10
4.4.1	Kaltstarts	10
4.4.2	Warmstarts	10
4.5	Handbetrieb	10
4.6	Ableitung des Hardwarefehler-Bits aus dem Gerätestatus	10
5	Die Repräsentation des Gerätes	10
5.1	Kennzeichnung des Gerätemodells	11
5.2	Die Master-Properties	11
5.2.1	POWER	11
5.2.2	STATUS	11
5.2.3	INIT	12
5.2.4	RESET	12
5.2.5	VERSION	12
5.2.6	INFOSTAT	12
5.2.7	CONSTANT	14
5.2.8	POSITS	14
5.2.9	POSITI	14
5.2.10	BLOCK	14
5.3	Die Slave-Properties	14
5.3.1	ACTIV	15
5.3.2	EQMERROR	15
5.3.3	COPYSET	16
II	Der Entwurf der Software	17

6	Softwareentwurf	17
7	Lokale Datenbasis	17
7.1	Tabelle der Konstanten	17
8	Dualport RAM	17
9	USRs - User Service Routinen	17
9.1	Obligatorische USRs	17
9.1.1	N_Init	17
9.1.2	N_Reset	17
9.1.3	R_Status	17
9.1.4	R_Power	17
9.1.5	W_Power	17
9.1.6	R_Active	17
9.1.7	W_Active	17
9.1.8	W_CopySet	17
9.1.9	R_EQMErr	17
9.1.10	R_Version	17
9.1.11	R_InfoStat	17
9.2	Gerätespezifische USRs	17
9.2.1	R_Constant	17
9.2.2	W_PositS	18
9.2.3	R_PositS	18
9.2.4	R_PositI	18
9.2.5	W_Block	18
9.2.6	R_Block	18
10	EQMs - Equipment Module	18
10.1	Interne Zustände	18
10.1.1	Bedeutung der internen Zustände	18
10.1.2	Übergänge zwischen den Zuständen	18
10.2	Periodisch konnektierte EQMs	18
10.2.1	Check_Move_EQM	18
10.2.2	Update_Config_EQM	19
10.3	Kommandogetriggerte EQMs	19
10.3.1	Dev_Init_EQM	19
10.3.2	Dev_Reset_EQM	19
10.3.3	Status_EQM	19
10.3.4	Power_EQM	19
10.3.5	Move_EQM	19
10.4	EQMs für die Diagnose vor Ort	19
10.4.1	Display_DPR_EQM	19
10.4.2	Display_DevErr_EQM	20
10.5	Sonstige EQMs	20
10.5.1	Startup_EQM	20
10.6	Globale Routinen	20
10.6.1	Read_and_Update_Status	20
10.6.2	Do_Intr_Service_Prep	20
11	Besonderheiten	20
	Literatur	21

Teil I

Das Gerätemodell

1 Die Aufgabe des Gerätes

Preßluftantriebe werden bei GSI benutzt, um verschiedenste Strahl Diagnoseelemente in den Strahlweg ein- und auszufahren. Dabei muß ein Preßluftantrieb nicht wissen, von welchem Typ das zu fahrende Element ist. Es wird also nicht zwischen Profiltitter- und Cup-Preßluftantrieb unterschieden.

2 Die Hardware des Gerätes

Von einem Steuereinschub können mehrere Pressluftantriebe angesprochen werden. Die Steuerung ist als Einsteck-Karten ausgelegt, von denen jede Karte zwei Antriebe bedienen kann. Bis zu 15 dieser Einsteck-Karten können in einem Steuereinschub untergebracht werden. Somit kann ein Steuereinschub bis zu 30 einzelne Pressluftantriebe bedienen.

Die Steuerelektroniken werden über einfache Interfacekarten angesprochen. Die erste Ausführung, bei denen die Ansteuerung über einen SD μ P erfolgte, sind nicht mehr im Einsatz.

3 Die Schnittstelle zum Gerät

3.1 Funktionscodes der Interfacekarte

Die für die Geräteansteuerung definierten Funktionscodes sind in der folgenden Tabelle 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 Codes und ihre Bedeutung sind in der folgenden Tabelle zusammengefasst.

Funktionscode		Modus	Bedeutung
Name	Hex		
<code>ifb_reset</code>	01	Funktion	Reset
<code>ifb_send_data</code>	06	Funktion	(<code>ifb_soll_1</code>) Datum an die Steuerelektronik schicken
<code>ifb_read_data</code>	81	Lesen	(<code>ifb_ist_1</code>) Datum aus der Steuerelektronik lesen
<code>ifb_read_status</code>	C0	Lesen	(<code>ifb_rdstat</code>) Gerätestatus lesen

Tabelle 1: Funktionscodes für den Zugriff auf die Steuerelektronik

Daten, die zur Steuerelektronik geschrieben werden, beinhalten in einem Wort die Adressierung des Antriebes und das Steuerkommando (siehe folgenden Abschnitt).

Der Vollständigkeit halber sind die Codes für den SD μ P noch aufgeführt. Diese Codes werden in der aktuellen Gerätesoftware nicht mehr verwendet.

Funktionscode		Modus	Bedeutung
Name	Hex		
<code>ifb_reset</code>	01	Funktion	Reset
<code>ifb_Pos_Rein</code>	20	Funktion	Antrieb reinfahren
<code>ifb_Pos_Raus</code>	21	Funktion	Antrieb rausfahren
<code>ifb_rdstat</code>	C0	Lesen	Gerätestatus lesen

Forts. auf nächster Seite

Forts. von letzter Seite

Funktionscode		Modus	Bedeutung
Name	Hex		

Tabelle 2: Ehemalige Funktionscodes für den SD μ P

3.2 Aufbau der Kommunikation mit der Steuerelektronik

3.2.1 Datenformat Schreibzugriff

Sollwerte an die Steuerelektronik sind in einem 16-Bit Wort kodiert, das aufgebaut ist wie folgt:

Bit	Bedeutung
0	interne Adresse des Antriebes,
:	Bereich 2..31
4	
5	Sollposition, 1: einfahren, 0: ausfahren
6	Disable Command
7	Disable Status
8	probably not used
:	
15	

Tabelle 3: Aufbau des Datenwortes bei Schreibzugriffen

Die Adresse des Pressluftantriebes liegt im Bereich 2 bis 31. Das bedeutet: Der erste Antrieb wird über die Adresse 2 angesprochen, der zweite Antrieb über die Adresse 3, usw.

In der VME Geräte-Software ist das Setzen eines Sollwertes über folgende Sequenz aus drei Schreib-Kommandos an die Interfacekarte implementiert:

1. Select Device. Setze die unteren 8 Bit des Schreib-Datenwortes auf die Werte:

Bit 0 ... 4: Interne Adresse des anzusteuernenden Antriebs (erster Antrieb hat die Adresse 2)

Bit 5: Sollposition. Auf den Wert 1 setzen

Bit 6: Disable command. Auf den Wert 1 setzen

Bit 7: Disable status, auf den Wert 1 setzen

Bit 8 ... 15: Bleiben auf dem Wert 0

Schreib-Datenwort mit dem Befehl `ifb_soll_1 (06hex)` schreiben

2. Select Command. Inhalt des Schreib-Datenwortes unverändert, bis auf

Bit 5: Sollposition. Herausfahren: 0, einfahren: 1

Schreib-Datenwort mit dem Befehl `ifb_soll_1 (06hex)` schreiben

3. Enable Command. Inhalt des Schreib-Datenwortes unverändert, bis auf

Bit 6: Disable command. Auf den Wert 0 setzen

Schreib-Datenwort mit dem Befehl `ifb_soll_1 (06hex)` schreiben

Es ist davon auszugehen, dass diese Sequenz bei der Kommunikation mit dem Gerät einzuhalten ist. Inwieweit von dieser Sequenz abgewichen werden kann, lässt sich nicht sagen, da auch bei den Betreuern der Steuerelektronik keine Beschreibung vorliegt, aus der die Anforderungen an die Kommunikation hervorgehen.

3.2.2 Lesezugriff

Lesezugriffe liefern ein 16-Bit Datenwort, das aufgebaut ist wie folgt:

Bit	Bedeutung
0	extern blockiert
1	intern blockiert
2	Alarm Temperatur
3	Endlage aussen, 0: Endlage erreicht
, 4	Endlage innen, 0: Endlage erreicht
5	externer Interlock
6	remote/local
7	reserved by (will be set by) software
8	not used
:	
15	

Tabelle 4: Aufbau des Datenwortes bei Lesezugriffen

Die Adresse des Pressluftantriebes liegt im Bereich 2 bis 31. Das bedeutet: Der erste Antrieb wird über die Adresse 2 angesprochen, der zweite Antrieb über die Adresse 3, usw.

In der VME Geräte-Software ist das Lesen eines Datenwortes über folgende Sequenz aus drei Schreib- / Lese-Kommandos an die Interfacekarte implementiert:

1. Select Device. Die unteren 8 Bit des Datenwortes sind zu setzen auf die Werte:

Bit 0 ... 4: Interne Adresse des anzusteuernenden Antriebs (erster Antrieb hat die Adresse 2)

Bit 5: Sollposition. Auf den Wert 1 setzen

Bit 6: Disable command. Auf den Wert 1 setzen

Bit 7: Disable status. Auf den Wert 1 setzen

Bit 8 ... 15: Auf den Wert 0 setzen

Schreib-Datenwort mit dem Befehl `ifb_soll.1 (06hex)` schreiben

2. Select Command. Inhalt des Schreib-Datenwortes unverändert, bis auf

Bit 7: Disable Status. Auf den Wert 0 setzen

Schreib-Datenwort mit dem Befehl `ifb_soll.1 (06hex)` schreiben

3. Read Data. Mit dem Befehl `ifb_ist.1 (81hex)` den Zustand für den jeweiligen Antrieb lesen. Das Format des gelesenen Datenwortes ist beschrieben in Tabelle 4 auf Seite 7.

Es ist davon auszugehen, dass diese Sequenz bei der Kommunikation mit dem Gerät einzuhalten ist. Inwieweit von dieser Sequenz abgewichen werden kann, lässt sich nicht sagen, da auch bei den Betreuern der Steuerelektronik keine Beschreibung vorliegt, aus der die Anforderungen an die Kommunikation hervorgehen.

3.3 Interlock Interrupt

Wird von Preßluftantrieben nicht generiert.

3.4 Data Request (DRQ) Interrupts

Wird von Preßluftantrieben nicht generiert.

3.5 Data Ready (DRD) Interrupts

Wird von Preßluftantrieben nicht generiert.

3.6 Umfang eines logischen Gerätes

In einem Einschub mit Preßluftelektroniken können bis zu 30 Antriebe untergebracht werden. Dabei bildet jeder einzelne Antrieb eine logische Geräteeinheit.

Abhängig vom Steckplatz seiner Elektronik im Steuereinschub hat jeder Antrieb eine eindeutige physikalische Adresse (Steckplatznummer + MIL-Adresse der Interface-Karte.

Zu Beachten: *Der Offset der physikalischen Adresse zur Grundadresse unterscheidet sich von der internen Geräteadressierung der Steuerelektronik.* Die Zählung der internen Adresse beginnt mit 2. Somit hat der erste Antrieb als physikalische Adresse die Grundadresse der Interface-Karte, aber 2 als interne Adresse der Steuerelektronik.

3.7 Definition der Bits des Hardwarestatus

Das Gerät liefert 1 Byte Statusinformation. Die Funktionscodes zum Lesen der entsprechenden Bytes sind in der folgenden Tabelle zusammengefaßt.

Statusbits	Funktionscode
16 ... 23	ifb_rdstat

Da in der Gerätesoftware bei der Ermittlung des Geräte-Status zuvor die Sequenz zum Lesen eines Datenwortes ausgeführt wird, ist zu vermuten, dass das Status-Byte nun korrekt gelesen wird, wenn zuvor eine „Select Device / Select Command“ Sequenz ausgeführt wird.

Für das Lesen des Status-Bytes kann man sich z.B. an der Konfigurations-Abfrage orientieren, siehe dazu Abschnitt 3.8 auf Seite 9.

Die Statusbits im Einzelnen sind in der folgenden Tabelle zusammengefasst.

Bit	Name	Bedeutung	
		High (1)	Low (0)
0	Power	on	off
1		unused	
⋮			
6			
7			
7	Online	Antrieb vorhanden	Antrieb nicht bestückt

Zu diesem gelesenen Status-Byte werden in der Property Status weitere Bits hinzugefügt. Der Aufbau der Daten der Status-Property ist beschrieben in Abschnitt 5.2.2 aus Seite 11 beschrieben sind.

Die Bedeutung des Bit 7 (Online): Mit einer Interface-Karte können bis zu 30 Antriebe angesprochen werden. An dem Online-Bit ist erkennbar, ob der jeweilig adressierte Antrieb tatsächlich bestückt ist. Achtung, soweit die Erinnerung zutreffend ist, kann nur erkannt werden,

ob die jeweilige Einschub-Karte vorhanden ist. Mit jeder dieser Karten können zwei Antriebe angesprochen werden, von denen aber nur jeweils einer vorhanden angeschlossen sein kann.

3.8 Konfigurationsabfrage

Ein logisches Gerät ist ansprechbar und damit im Kontrollsystem vorhanden, wenn von der Interfacekarte mit dem Funktionscode $C0_{hex}$ (ifb_rdstat) ein Status gelesen werden kann und das Onlinebit im Status gesetzt ist (siehe Abschnitt 3.7 auf Seite 8).

Um dieses Bit im Status auswerten zu können, ist in der Gerätesoftware folgende Befehls-Sequenz implementiert:

1. Select Device. Die unteren 8 Bit des Datenwortes sind zu setzen auf die Werte:

- Bit 0 ... 4:** Interne Adresse des anzusteuernenden Antriebs (erster Antrieb hat die Adresse 2)
- Bit 5:** Sollposition, auf 1 setzen
- Bit 6:** Disable command, auf 1 setzen
- Bit 7:** Disable status: auf *Wert 1* setzen
- Bit 8 ... 15:** Auf den Wert 0 setzen

Schreib-Datenwort mit dem Befehl ifb_soll.1 (06_{hex}) schreiben

2. Select Command. Inhalt des Datenwortes belassen, bis auf

- Bit 7:** Disable status: auf *Wert 0* setzen

Schreib-Datenwort mit dem Befehl ifb_soll.1 (06_{hex}) schreiben

3. Read Status. Mit dem Befehl ifb_read_status ($C0_{hex}$) den Status für den jeweiligen Antrieb lesen.

Es ist davon auszugehen, dass diese Sequenz bei der Kommunikation mit dem Gerät einzuhalten ist. Inwieweit von dieser Sequenz abgewichen werden kann, lässt sich nicht sagen, da auch bei den Betreuern der Steuerelektronik keine Beschreibung vorliegt, aus der die Anforderungen an die Kommunikation hervorgehen.

4 Die Bedienung des Gerätes

4.1 Aufgaben im Normalbetrieb

4.1.1 Antrieb Ein/Ausfahren

Für jeden Antrieb gibt es eine maximal erlaubte Fahrzeit (steht in der VME-Datenbank), die beim Ein/Ausfahren des Antriebes zu überprüfen ist. Erreicht der Antrieb die gewünschte Endlage nicht innerhalb dieser Zeit, so ist ein Fehler zu melden. Vor dem Fahren eines Antriebes ist zu prüfen, ob alle Verriegelungsmechanismen (siehe Statusbits in Abschnitt 5.2.2 auf Seite 11) *ok* anzeigen (Power, externer Interlock, externe u. interne Blockierung, Remote/Local).

4.1.2 Verriegelung eines Antriebs

Preßluftantriebe müssen softwaremäßig verriegelt werden können. Wird ein Antrieb verriegelt oder entriegelt, so ist ein entsprechender Alarm zu erzeugen.

Dabei ist jedoch zu beachten, daß die Verriegelung per Software nur vor unabsichtlicher und nicht (weil sie von jedem aufgehoben werden kann) vor absichtlicher Fehlbedienung schützen kann! Die Softwareverriegelung eines Antriebs bewirkt lediglich eine Zurückweisung der Fahrbefehle.

4.2 Zeitkritische Anforderungen

Bei Preßluftantrieben sind keine zeitkritischen Anforderungen zu berücksichtigen.

4.3 Einordnung in das Timing

Preßluftantriebe nehmen nicht an der PPM (Puls-zu-Puls-Modulation) teil und sind damit unabhängig vom Timing.

4.4 Festlegung von Startwerten

4.4.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset durchgeführt.
- Bei Init eines einzelnen Gerätes oder der ganzen SE wird auch der SD μ P reseted.
- Die aktuelle Position des Antriebs (drin oder draußen) wird als aktueller Sollwert übernommen.
- Ein Init hat auf das Gerät direkt keinen Einfluß (das Gerät wird nicht auf eine bestimmte Ausgangsposition gefahren).

4.4.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Bei Reset eines einzelnen Gerätes oder der ganzen SE wird auch der SD μ P reseted.
- Es wird ein Gerätereset durchgeführt.
- Ein Reset hat auf das Gerät direkt keinen Einfluß (das Gerät wird nicht auf eine bestimmte Ausgangsposition gefahren).

4.5 Handbetrieb

Ein Handbetrieb des Gerätes ist vorgesehen und jederzeit möglich, auch wenn das Gerät softwaremäßig blockiert ist!.

4.6 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
10	Temperaturüberwachung	1
13	externer Interlock	1

5 Die Repräsentation des Gerätes

Dieses Kapitel definiert das Gerätemodell, also wie das Gerät nach höheren Ebenen hin abgebildet wird. Es beschreibt die Schnittstelle zwischen Benutzerebene (Operatingprogrammen) und Geräteebene (Gerätehard- und -software).

Ein Gerät erscheint zur Benutzerebene im Umfang des in Abschnitt 3.6 definierten logischen Gerätes.

5.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **PLA**.

Die Gerätemodellnummer ist 10_{dez} .

5.2 Die 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
CONSTANT	R	0	–	1	BitSet16	s	1
POSITS	R/W	0	–	1	BitSet16	1	0
POSITI	R/W	0	–	1	BitSet16	1	0

5.2.1 POWER

Bedeutung: Gibt an, ob der Leistungsteil des Gerätes ein- oder ausgeschaltet ist.

Parameter: Keine.

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

5.2.2 STATUS

Bedeutung: Auslesen des 32bit Gerätestatus.

Parameter: Keine.

Daten: Das 32bit Statuswort. Das Statuswort setzt sich zusammen aus dem von der Steuer-Elektronik gelesenen Datenwort, dem gelesenen Status-Byte und einigen Bits, von der Geräte-Software gesetzt werden.

Die Bits 0 . . . 7 sind die systemweiten sogenannten 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

Forts. auf nächster Seite

Forts. von letzter Seite

Bit	Name	Bedeutung	
		High (1)	Low (0)
8	externe Blockierung	nicht blockiert	blockiert
9	interne Blockierung	nicht blockiert	blockiert
10	Temperaturüberwachung	Temperatur ok	Temperaturalarm
11	Endlage außen	erreicht	nicht erreicht
12	Endlage innen	erreicht	nicht erreicht
13	externer Interlock	kein Interlock	Interlock
14	Remote/Local	Remote	Local
15	Software-Blockierung	nicht blockiert	blockiert
16	Power	on	off
17		unused	
:		:	
31		unused	

5.2.3 INIT

Bedeutung: Initialisierung des Gerätes (Kaltstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.4.1 auf Seite 10.

Parameter: Keine.

Daten: Keine.

5.2.4 RESET

Bedeutung: Reset des Gerätes (Warmstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.4.2 auf Seite 10.

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: Pulszentralen-Identifikation:
 - 0: TIF
 - 1: SIS-PZ
 - 2: ESR-PZ
 - 3...6: undefiniert
 - 7: Software-PZ
 - 8: UNILAC, Master-PZ

- 9: UNILAC-PZ 1
- 10: UNILAC-PZ 2
- 11: UNILAC-PZ 3
- 12: UNILAC-PZ 4
- 13: UNILAC-PZ 5
- 14: UNILAC-PZ 6
- 15: UNILAC-PZ 7

24: Reserviert für Erweiterungen.

25: Reserviert für Erweiterungen.

5.2.7 CONSTANT

Bedeutung: Lesen der gerätespezifischen Konstanten.

Parameter: Keine.

Daten: Das Datum beinhaltet die maximal erlaubte Fahrzeit (in s).

5.2.8 POSITS

Bedeutung: Lesen bzw. Setzen der gewünschten Endlage eines Antriebs.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Eins heißt, der Antrieb hat Endlage *innen*. Null heißt, der Antrieb hat Endlage *außen*.

5.2.9 POSITI

Bedeutung: Lesen der aktuellen Endlage eines Antriebs.

Parameter: Keine.

Daten: Das Datum kann folgende Werte annehmen:

0	Endlage außen
1	Endlage innen
2	keine Endlage (Antrieb fährt noch, oder Hardwarefehler)

5.2.10 BLOCK

Diese Property wird nicht mehr unterstützt.

Bedeutung: Lesen bzw. Setzen der Softwareverriegelung eines Antriebs. Beim Setzen der Verriegelung wird ein *ExtDevSpecAlarm* erzeugt, in dessen Extension die Knotennummer (VAX-Node im Ethernet 16Bit) und der Prozessname (6 Character) des auftraggebenden Prozesses (VAX-Prozess) steht.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Eins heißt, der Antrieb ist verriegelt. Null heißt, der Antrieb ist *nicht* verriegelt.

5.3 Die 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	Integer32	348	Integer32	1	0

5.3.1 ACTIV

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

Bei Preßluftantrieben kann diese *obligatorische* Property nur gelesen aber nicht geschrieben werden.

Parameter: Keine.

Daten: Da Preßluftantriebe nicht an der PPM teilnehmen, wird beim Lesen immer Eins (heißt, das Gerät ist für den zugeordneten Beschleuniger aktiv) geliefert.

5.3.2 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: Hier hat nur der erste der 217 Parameter eine Bedeutung.

1: Wird bei konnektierten Aufträgen ausgewertet. 0: Es wird bei jeder Ausführung des Auftrages eine Antwort verschickt. 1: Es wird bei jeder Ausführung des Auftrages nur dann eine Antwort verschickt, wenn sich seit dem letzten Aufruf der Inhalt der Daten geändert hat.

2...217: Dummy, sie werden vom MOPS intern verwendet und können vom Benutzer beliebig gesetzt werden.

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:

$$l = m + s$$

$$t = m + s + b$$

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

5.3.3 COPYSET

Bedeutung: Kopiert alle Geräteeinstellungen (Sollwerte) eines virtuellen ('fremden') Beschleunigers in den zugehörigen ('eigenen') Beschleuniger.
Da Preßluftantriebe nicht an der PPM teilnehmen (also keine beschleunigerabhängigen Daten besitzen), hat diese *obligatorische* Property keine Auswirkungen auf die Geräteeinstellung.

Parameter: Keine.

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

Teil II

Der Entwurf der Software

6 Softwareentwurf

Kein erwähnenswerten Besonderheiten.

7 Lokale Datenbasis

7.1 Tabelle der Konstanten

Für jedes Gerät gibt es einen Wert in der Konstantentabelle der lokalen Datenbasis. Dieser Wert beinhaltet die maximale Fahrzeit eines Antriebs in s.

8 Dualport RAM

Keine Besonderheiten

9 USRs - User Service Routinen

9.1 Obligatorische USRs

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 der Preßluftantriebe folgende gerätespezifischen USRs benötigt:

9.2.1 R_Constant

Lesen der gerätespezifischen Konstanten.

9.2.2 W_PositS

Setzen der Endlage (innen oder außen).

9.2.3 R_PositS

Lesen der gewünschten Endlage.

9.2.4 R_PositI

Lesen der aktuellen Endlage.

9.2.5 W_Block

Lesen des aktuellen Verriegelungszustandes (0 heißt nicht verriegelt, 1 heißt verriegelt). Dabei wird ein *ExtDevSpecAlarm* erzeugt, in dessen Extension die Knotennummer (VAX-Node im Ethernet 16Bit) und der Prozessname (6 Character) des auftraggebenden Prozesses (VAX-Prozess) steht.

9.2.6 R_Block

Lesen des aktuellen Verriegelungszustandes (0 heißt nicht verriegelt, 1 heißt verriegelt).

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.
- ready Das Gerät ist bereit für Aktionen. Ausgangszustand am Beginn eines virtuellen Beschleunigers.
- busy Antrieb fährt (rein oder raus).

10.1.2 Übergänge zwischen den Zuständen

Die Zustände und die Übergänge zwischen denselben sind in Tabelle 10 zusammengefasst. Die Legende zu diesen Tabellen ist in Tabelle 11 zu finden.

Tabelle der Zustandsübergänge			
von nach →	ready	busy	
ready	U:	–	Fahrkommando
	B:	–	RP,IL,EB,IB,SB
	A:	–	Move_EQM
busy	U:	Endlage erreicht	–
	B:	–	–
	A:	Check_Move_EQM (periodisch)	–

Tabelle 10: Zustandsübergangsdiagramm

10.2 Periodisch konnektierte EQMs

10.2.1 Check_Move_EQM

Zeit: 0.5s

Legende

- Die Priorität der Zustände (höchste Priorität zuerst): ready, busy.
- U: Auslösende Ursache.
Fahrkommando Antrieb wurde per Kommando zum rein/rausfahren veranlasst.
- B: Abzuprüfende Bedingung.
R Remotebit des Status steht auf Remote.
P Powerbit des Status steht auf Power on.
IL kein externer Interlock.
EB keine externe Blockierung.
IB keine interne Blockierung.
SB keine Softwareblockierung.
- A: Ausführende Stelle des Zustandsübergangs.
..._EQM Innerhalb des EQMs ..._EQM.

Tabelle 11: Legende zu den Zustandsübergangsdiagrammen

Anzahl: mehrmalige Ausführung (hängt von der maximal erlaubten Fahrzeit eines Antriebs ab).

Aktion: Es wird überprüft, ob der Antrieb die gewünschte Endlage erreicht hat oder die maximale Fahrzeit überschritten wurde.

10.2.2 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.

10.3 Kommandogetriggerte EQMs

10.3.1 Dev_Init_EQM

10.3.2 Dev_Reset_EQM

10.3.3 Status_EQM

10.3.4 Power_EQM

10.3.5 Move_EQM

Es wird geprüft, ob der Antrieb fahrbereit ist (Zustand `dev_ready`), das Power-Bit im Status *Power on* anzeigt, ob alle Blockierungssignale im Status *nicht blockiert* anzeigen und ob der Antrieb auf *Remote*-Bedienung steht.

Sind alle Bedingungen erfüllt, so wird das Fahrkommando an das Gerät abgesetzt und das *CheckMove_EQM* als periodischer Auftrag gestartet. Ansonsten wird ein entsprechender Fehler gemeldet.

Ist der Antrieb nicht fahrbereit (Zustand `dev_busy`), so wird geprüft, ob er in die gewünschte Richtung fährt (rein oder raus). Ist dies der Fall wird das *CheckMove_EQM* als periodischer Auftrag gestartet. Fährt der Antrieb in die entgegengesetzte Richtung, wird ein Fehler gemeldet.

10.4 EQMs für die Diagnose vor Ort

10.4.1 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.

10.4.2 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.

10.5 Sonstige EQMs

10.5.1 Startup_EQM

Schaltet die SE in den Command-Mode.

10.6 Globale Routinen

Hier werden alle Routinen aufgeführt, die im Modul EQMs global definiert sind und von verschiedenen EQMs benutzt werden.

10.6.1 Read_and_Update_Status

10.6.2 Do_Intr_Service_Prep

11 Besonderheiten

Bisher sind bei Preßluftantrieben keine erwähnenswerten Besonderheiten bekannt.

Index

- Änderungsprotokoll, 2
- Abriß, 2
- Antrieb Ein/Ausfahren, 9
- Aufgabe des Gerätes, 5

- Bedienung des Gerätes, 9
- Besonderheiten, 20
- Blockierung, 9

- Check_Move_EQM, 18

- Datenbasis, 17
- Dev_Init_EQM, 19
- Dev_Reset_EQM, 19
- Display_DevErr_EQM, 20
- Display_DPR_EQM, 19
- DRD Interrupt, 8
- DRQ Interrupt, 8
- Dualport RAM, 17

- EQMs, 18
 - für die Diagnose vor Ort, 19
 - Display_DevErr_EQM, 20
 - Display_DPR_EQM, 19
 - Globale Routinen, 20
 - Do_Intr_Service_Prep, 20
 - Read_and_Update_Status, 20
 - Kommandogetriggerte, 19
 - Dev_Init_EQM, 19
 - Dev_Reset_EQM, 19
 - Move_EQM, 19
 - Power_EQM, 19
 - Status_EQM, 19
 - Periodisch konnektierte, 18
 - Check_Move_EQM, 18
 - Update_Config_EQM, 19
 - Sonstige, 20
 - Startup_EQM, 20
- Eventkonnektierungen, 10

- Funktionscodes, 5

- Gerät
 - Aufgabe, 5
 - Bedienung, 9
 - Hardware, 5
 - logisches, 8
 - Repräsentation, 10
 - Schnittstelle, 5

- Gerätemodell, 5
 - Kennzeichnung, 11
 - Master-Properties, 11
 - Slave-Properties, 14
- Globale Routinen, 20

- Handbetrieb, 10
- Hardware des Gerätes, 5
- Hardwarefehler-Bit, 10
- Hardwarestatus, 8

- Init, 10
- Interfacekarte, 5
- Interlock, 8
- Interne Zustände, 18
- Interrupt
 - DRD Interrupt, 8
 - DRQ Interrupt, 8
 - Interlock, 8

- Kaltstarts, 10
- Kommandogetriggerte EQMs, 19
- Konfigurationsabfrage, 9

- logisches Gerät, 8
- Lokale Datenbasis, 17
 - Tabelle der Konstanten, 17

- Master-Properties, 11
- Move_EQM, 19

- N_Init, 17
- N_Reset, 17
- Normalbetrieb, 9

- Periodisch konnektierte EQMs, 18
- Power_EQM, 19
- Properties
 - ACTIV, 15
 - BLOCK, 14
 - CONSTANT, 14
 - COPYSET, 16
 - EQMERROR, 15
 - INFOSTAT, 12
 - INIT, 12
 - Master-, 11
 - POSITI, 14
 - POSITS, 14
 - POWER, 11
 - RESET, 12

- Slave-, 14
- STATUS, 11
- VERSION, 12
- R.Active, 17
- R.Block, 18
- R.Constant, 17
- R.EQMErr, 17
- R.InfoStat, 17
- R.PositI, 18
- R.PositS, 18
- R.Power, 17
- R.Status, 17
- R.Version, 17
- Repräsentation des Gerätes, 10
- Reset, 10
- Schnittstelle zum Gerät, 5
- Slave-Properties, 14
- Softwareentwurf, 17
- Softwarestatus, 11
- Sonstige EQMs, 20
- Startup_EQM, 20
- Startwerte, 10
- Status_EQM, 19
- Statusbits, 8
- Timing, 10
- Update_Config_EQM, 19
- USRs, 17
 - gerätespezifische, 17
 - R.Block, 18
 - R.Constant, 17
 - R.PositI, 18
 - R.PositS, 18
 - W.Block, 18
 - W.PositS, 18
 - obligatorische, 17
 - N.Init, 17
 - N.Reset, 17
 - R.Active, 17
 - R.EQMErr, 17
 - R.InfoStat, 17
 - R.Power, 17
 - R.Status, 17
 - R.Version, 17
 - W.Active, 17
 - W.CopySet, 17
 - W.Power, 17
- Verriegelung, 9
- W.Active, 17
- W.Block, 18
- W.CopySet, 17
- W.PositS, 18
- W.Power, 17
- Warmstarts, 10
- Zeitkritische Anforderungen, 10
- Zustände
 - Interne, 18
 - Übergänge, 18
 - Bedeutung, 18
- Varianten
 - Betriebs-, 10