

IPS - Interactive Peripheral Systems

Gerätemodell und Softwareentwurf

P. Kainberger

Dieses Papier enthält die Beschreibung des Gerätemodells 'IPS - Interactive-Peripheral-Systems' und den Entwurf der Gerätesoftware für dieses Gerät.

Im IP-System sind Bedienelemente (Potis und Taster) der Konsolen integriert, die dem Operateur insbesondere beim Optimieren der Strahlführung (Fädeln) ein interaktives bedienen von Beschleunigerkomponenten (hauptsächlich Magnetnetzgeräte) mit hohem Datendurchsatz (quasi analog) ermöglichen.

Änderungsprotokoll			
Datum	GM-Version	Name	Kommentar
03. März 94	IPS_03	P. Kainberger	erste Version (aus Rahmen)
13. Apr. 94	IPS_03	P. Kainberger	Fehlerkorrekturen

Inhaltsverzeichnis

I	Das Gerätemodell	7
1	Die Aufgabe des Gerätes	7
2	Die Hardware des Gerätes	7
3	Die Schnittstelle zum Gerät	7
3.1	Funktionscodes der Interfacekarte	7
3.2	Hardware der IPS-Komponenten	8
3.3	Data Request (DRQ) Interrupts	10
3.4	Data Ready (DRD) Interrupts	10
3.5	Umfang eines logischen Gerätes	10
3.6	Definition der Bits des Hardwarestatus	10
3.7	Konfigurationsabfrage	10
4	Die Bedienung des Gerätes	10
4.1	Aufgaben im Normalbetrieb	10
4.1.1	Inkrementalgeber	11
4.1.2	Taster	11
4.1.3	Helligkeitsschalter	11
4.2	Einordnung in das Timing	11
4.3	Festlegung von Startwerten	11
4.3.1	Kaltstarts	11
4.3.2	Warmstarts	11
4.4	Verhalten bei Störungen	12
4.4.1	Ausfall der Kommunikation EC – Gerät	12
4.5	Bedienungsfehler vom Operating	12
5	Die Repräsentation des Gerätes	12
5.1	Kennzeichnung des Gerätemodells	12
5.2	Die Master-Properties	12
5.2.1	POWER	12
5.2.2	STATUS	13
5.2.3	INIT	13
5.2.4	RESET	13
5.2.5	VERSION	13
5.2.6	INFOSTAT	13
5.2.7	CONFIGUR	15
5.2.8	ACTIVATE	15
5.2.9	DISPLAY	16
5.2.10	VALUE	16
5.2.11	ALLOCATE	16
5.2.12	ALLOCATE	17
5.2.13	POLLRATE	17
5.2.14	INCFAC	17
5.3	Die Slave-Properties	17
5.3.1	ACTIV	18
5.3.2	EQMERROR	18
5.3.3	COPYSET	19

II	Der Entwurf der Software	21
6	Softwareentwurf	21
7	Lokale Datenbasis	21
8	Dualport RAM	21
9	USRs - User Service Routinen	21
9.1	Obligatorische USRs	21
9.1.1	N_Init	21
9.1.2	N_Reset	21
9.1.3	R_Status	21
9.1.4	R_Power	21
9.1.5	W_Power	21
9.1.6	R_Active	21
9.1.7	W_Active	21
9.1.8	W_CopySet	21
9.1.9	R_EQMErr	21
9.1.10	R_Version	21
9.1.11	R_InfoStat	21
9.2	Gerätespezifische USRs	21
9.2.1	R_Configur	21
9.2.2	R_Activate	21
9.2.3	W_Activate	22
9.2.4	W_Display	22
9.2.5	R_Value	22
9.2.6	R_Allocate	22
9.2.7	W_Allocate	22
9.2.8	R_Pollrate	22
9.2.9	W_Pollrate	22
9.2.10	R_IncFact	22
9.2.11	W_IncFact	22
9.3	Globale Routinen	22
9.3.1	set_owner	22
9.3.2	reset_owner	22
9.3.3	check_owner	23
9.3.4	no_owner	23
10	EQMs - Equipment Module	23
10.1	Interne Zustände	23
10.1.1	Bedeutung der internen Zustände	23
10.1.2	Übergänge zwischen den Zuständen	23
10.2	Periodisch konnektierte EQMs	23
10.2.1	Update_Brightness_EQM	23
10.2.2	Update_Devices_EQM	24
10.2.3	Update_Config_EQM	24
10.3	An externe Interrupts konnektierte EQMs	24
10.3.1	DRD_EQM	24
10.3.2	DRQ_EQM	24
10.4	Kommandogetriggerte EQMs	24
10.4.1	Dev_Init_EQM	24
10.4.2	Dev_Reset_EQM	24
10.4.3	Status_EQM	24

10.4.4	Active_EQM	24
10.4.5	Power_EQM	24
10.4.6	Activate_EQM	24
10.4.7	Write_EQM	24
10.5	EQMs für die Diagnose vor Ort	24
10.5.1	Display_DPR_EQM	24
10.5.2	Display_DevErr_EQM	25
10.5.3	Display_Grp_EQM	25
10.6	Sonstige EQMs	25
10.6.1	Startup_EQM	25
10.7	Globale Routinen	25
10.7.1	Read_and_Update_Status	25
10.7.2	Do_Intr_Service_Prep	25
10.7.3	TstBit	25
10.7.4	Get_Value	25
10.7.5	Set_Value	26
10.7.6	Dev_Adr	26
10.7.7	dv\$box_reset	26
10.7.8	dv\$brightswitch	26
10.7.9	dv\$interrupt	26
10.7.10	dv\$devices	26
10.7.11	dv\$activate	26
10.7.12	dv\$deactivate	26
10.7.13	dv\$set_brightness	26
10.7.14	dv\$read	26
10.7.15	dv\$write	26
10.7.16	set_brightness	26
10.7.17	init_box	26
10.7.18	update_devices	26
11	Besonderheiten	27
	Index	29

Teil I

Das Gerätermodell

1 Die Aufgabe des Gerätes

Aufgabe der IPS-Komponenten (Interactive-Peripheral-System) ist es mit hoher Geschwindigkeit Eingaben eines Benutzers entgegenzunehmen und über das Kontrollsystem an ein Operatingprogramm zur Verarbeitung weiterzureichen. Ebenso müssen Daten und Informationen des Operatingprogramms zur Visualisierung von Prozeß- oder Geräte-Zuständen an den IPS-Komponenten dargestellt werden.

2 Die Hardware des Gerätes

Die verschiedenen IPS-Komponenten sind in Gruppen von 0 bis 7 aufgeteilt, wobei nach dem jetzigen Stand nur noch die Gruppen 0 (Inkrementalgeber) und 1 (Tastenmodul) benutzt werden.

Poti (Inkrementalgeber): Der Inkrementalgeber bietet die Möglichkeit, Parameter *quasi analog* vorzugeben. Er besitzt zwei acht-stellige alphanumerische Anzeigen, die beliebig verwendbar sind. Z.B. für die Darstellung der Gerätenomenklatur und eines gerätespezifischen Parameters (Sollwert, Istwert, ...). Angezeigt werden können alle Buchstaben, Ziffern und Sonderzeichen im ASCII-Code.

Der Inkrementalgeber liefert 128 Inkremente pro Umdrehung, die regelmäßig vom Kontrollrechner (VME-SE) abgefragt werden müssen. Ein Interruptbetrieb ist nicht möglich.

Taster: Das Tastenmodul besitzt wie der Inkrementalgeber zwei alphanumerische Anzeigen, die genauso benutzt werden können. Wenn die Taste gedrückt wurde, erzeugt das Tastenmodul einen Interrupt, der vom Kontrollrechner entsprechend ausgewertet und weitergereicht werden muß. Wird die Taste wieder losgelassen, erzeugt das Modul einen weiteren (vom ersten unterscheidbaren) Interrupt. Zur Zeit wird vom Kontrollrechner nur der zweite (für *Taste losgelassen*) Interrupt verwertet. Bei Tastenmodulen ist kein regelmäßiges Abfragen (*polling*) durch den Kontrollrechner nötig.

3 Die Schnittstelle zum Gerät

Alle IPS-Komponenten sind über eine Interfacekarte (FG 380 102) mit Adreß- und Datenbus und einem sog. *GSI-PIGGY* angeschlossen.

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		
ifb_data_write	06	Schreiben	Datenbus setzen
ifb_addr_write	11	Schreiben	Adreßbus setzen
ifb_reset	14	Funktion	Reset aller Komponenten
ifb_data_read	80	Lesen	Datenbus lesen

Funktionscode		Modus	Bedeutung
Name	Hex		
ifb_piggy_id	8E	Lesen	Identifikationsnummer der IFK lesen
ifb_rdstat	C0	Lesen	Status lesen

ifb_data_write

Über den Datenbus werden 8 Bit Daten (Lowbyte des über den MIL-Bus empfangenen 16 Bit Wertes) an die adressierte Komponente geschickt.

ifb_addr_write

Am Adreßbus wird eine 16 Bit Adresse zur Selektion einer Komponente oder einer Funktion angelegt.

ifb_reset

Alle angeschlossenen Komponenten werden in einen definierten Ausgangszustand gebracht.

ifb_data_read

Vom Datenbus werden die selektierten Daten gelesen.

ifb_piggy_id

Die Identifikationsnummer des Interfacekarten-Piggy wird gelesen. Daran läßt sich erkennen, ob es sich um IPS-Komponenten handelt oder nicht.

ifb_rdstat

Der Status wird gelesen.

3.2 Hardware der IPS-Komponenten

Über den Adreßbus können die einzelnen IPS-Komponenten und ihre Funktionen selektiert werden.

Adreßbusbelegung für Poti									
Funktion		Gerätenummer							
		0	1	2	3	4	5	6	7
Write Display oben	Read Lowbyte	FD20	FD24	FD28	FD2C	FD30	FD34	FD38	FD3C
Write Display unten	Read Highbyte	FD21	FD25	FD29	FD2D	FD31	FD35	FD39	FD3D
Write Command		FD22	FD26	FD2A	FD2E	FD32	FD36	FD3A	FD3E
Reserve		FD23	FD27	FD2B	FD2F	FD33	FD37	FD3B	FD3F
Identifikation		FD19							
Request		FD1A							
Helligkeitsschalter		FD1F							

In folgender Weise könne die einzelnen Funktionen zur Ausführung gebracht werden:

Write-Display: Nach Selektion der Displayadresse (oben/unten) des gewünschten Gerätes, können 8 Zeichen (ASCII) geschrieben werden (über den Datenbus). Diese werden automatisch von links nach rechts zur Darstellung gebracht.

Read-Low/High-Byte: Nach Selektion der Readadresse (low/high-byte) des gewünschten Gerätes, kann die Anzahl der Inkremente, die sich seit dem letzten Lesen angesammelt haben, gelesen werden. Mit dem Lesen des Highbyte wird der interne Inkrementzähler automatisch gelöscht.

Bei Inkrementalgebern können nach Selektion der *Write Command*-Funktion über den Adreßbus folgende Kommandos durch Schreiben einer 8 Bit Funktionsnummer auf den Datenbus (in Klammer angegeben) zur Ausführung gebracht werden:

Reset-Display-Counter (80_{Hex}): Vor dem Beschreiben der Displays muß mit diesem Kommando der Display-Counter, der intern dafür sorgt, daß die Zeichen von links nach rechts zur Darstellung kommen, zurückgesetzt (d.h. Startposition ganz links) werden.

Send-Signal (40_{Hex}): Mit diesem Kommando kann die Sendelampe für 0.2s zum Leuchten gebracht werden. Mit dieser Lampe zeigt der Kontrollrechner an, daß eine Eingabe empfangen und bearbeitet wurde.

Aktiv (01_{Hex}): Pulsgeber für die Inkrementalgeber enablen. Gleichzeitig wird auch die *Aktiv*-Lampe eingeschaltet.

Passiv (02_{Hex}): Pulsgeber für die Inkrementalgeber disablen. Gleichzeitig wird auch die *Aktiv*-Lampe ausgeschaltet.

Adreßbusbelegung für Taster									
Funktion		Gerätenummer							
		0	1	2	3	4	5	6	7
Write Display oben	Read Lowbyte	FD40	FD44	FD48	FD4C	FD50	FD54	FD58	FD5C
Write Display unten	Read Highbyte	FD41	FD45	FD49	FD4D	FD51	FD55	FD59	FD5D
Write Command		FD42	FD46	FD4A	FD4E	FD52	FD56	FD5A	FD5E
Reserve		FD43	FD47	FD4B	FD4F	FD53	FD57	FD5B	FD5F
Identifikation		FD1B							
Interr.Request <i>EIN</i>		FD1C							
Interr.Request <i>AUS</i>		FD1D							
Helligkeitsschalter		FD1F							

In folgender Weise könne die einzelnen Funktionen zur Ausführung gebracht werden:

Write-Display: Nach Selektion der Displayadresse (oben/unten) des gewünschten Gerätes, können 8 Zeichen (ASCII) geschrieben werden (über den Datenbus). Diese werden automatisch von links nach rechts zur Darstellung gebracht.

Bei Tastenmodulen können nach Selektion der *Write Command*-Funktion über den Adreßbus folgende Kommandos durch Schreiben einer 8 Bit Funktionsnummer auf den Datenbus (in Klammer angegeben) zur Ausführung gebracht werden:

Reset-Display-Counter (80_{Hex}): Vor dem Beschreiben der Displays muß mit diesem Kommando der Display-Counter, der intern dafür sorgt, daß die Zeichen von links nach rechts zur Darstellung kommen, zurückgesetzt (d.h. Startposition ganz links) werden.

Send-Signal (40_{Hex}): Mit diesem Kommando kann die Sendelampe für 0.2s zum Leuchten gebracht werden. Mit dieser Lampe zeigt der Kontrollrechner an, daß eine Eingabe empfangen und bearbeitet wurde.

Aktiv (01_{Hex}): Taster-EIN Funktion enablen. Damit löst der Taster einen Interrupt aus, wenn er gedrückt wurde. Gleichzeitig wird auch die *Aktiv*-Lampe eingeschaltet.

Passiv (02_{Hex}): Taster-EIN Funktion disable. Gleichzeitig wird auch die *Aktiv*-Lampe ausgeschaltet.

Aktiv II (04_{Hex}): Taster-AUS Funktion enable. Damit löst der Taster einen Interrupt aus, wenn er losgelassen wurde. Gleichzeitig wird auch die *Aktiv*-Lampe eingeschaltet.

Passiv II (08_{Hex}): Taster-AUS Funktion disable. Gleichzeitig wird auch die *Aktiv*-Lampe ausgeschaltet.

Clear EIN (10_{Hex}): Der Interrupt-Request für Taster-EIN wird gelöscht.

Clear AUS (20_{Hex}): Der Interrupt-Request für Taster-AUS wird gelöscht.

3.3 Data Request (DRQ) Interrupts

Ein DRQ-Interrupt wird erzeugt, wenn ein Benutzer einen der Reset-Knöpfe gedrückt hat. Danach sollte der Kontrollrechner einen Konfigurationsupdate und einen Reset an allen Komponenten durchführen.

3.4 Data Ready (DRD) Interrupts

IPS-Tastenmodule melden per DRD-Interrupt, daß Eingaben eines Benutzers erfolgt sind und vom Kontrollrechner gelesen und ausgewertet werden müssen.

3.5 Umfang eines logischen Gerätes

Maximal zwei Interfacekarten repräsentieren ein logisches Gerät. Dabei gilt, daß die zweite Interfacekarte zur ersten einen Adreßabstand von genau 80_{Hex} (MSB im Adreßbyte) aufweisen muß (z.B. 01_{Hex} und 81_{Hex}, 20_{Hex} und A0_{Hex}, ...). Vom Kontrollrechner aus werden alle Komponenten an beiden Interfacekarten der ersten Interfacekarte zugeordnet.

3.6 Definition der Bits des Hardwarestatus

IPS-Komponenten liefern keine, im Sinne des Kontrollsystems, relevanten Statusinformationen.

3.7 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.

4 Die Bedienung des Gerätes

4.1 Aufgaben im Normalbetrieb

Bevor eine IPS-Komponente von einem Operatingprogramm aus benutzt werden kann, muß die Komponente *allokiert* werden. Dabei wird in den Datenstrukturen der Geräteverwaltung die VAX-Knotennummer und die Process-id (beides steht im Ethernet-header) als Owner-id eingetragen. Interaktionen mit der IPS-Komponente darf nur dieser eingetragene Owner durchführen, alle anderen Zugriffe werden abgewiesen. Nach Beendigung aller Aktivitäten muß das Operatingprogramm (also der Owner) die allokierten Geräte durch *Deallokierung* wieder freigeben.

4.1.1 Inkrementalgeber

In regelmäßigen Abständen (90 ms) müssen vom Kontrollrechner alle *aktiven* Inkrementalgeber abgefragt werden (Adr. *Read-Low-Byte* und *Read-High-Byte*, wobei mit dem Lesen des Highbyte automatisch beide Bytes in der Hardware gelöscht werden). Bei Änderungen muß ein Interrupt zum G μ P ausgelöst werden.

4.1.2 Taster

Ein aktives Tastenmodul meldet per DRD-interrupt, wenn die Taste gedrückt worden ist. Der Kontrollrechner muß dann einen Interrupt zum G μ P auslösen.

4.1.3 Helligkeitsschalter

In regelmäßigen Abständen (500 ms) muß vom Kontrollrechner der Helligkeitsschalter abgefragt (Adr. *Helligkeitsschalter*) und bei Änderungen die Einstellung der Helligkeit für alle Komponenten in folgender Weise geändert werden:

1. Komponente (Poti oder Taste) *inactive* schalten
2. Helligkeit für oberes Display setzen (Adr. *Write-Display-Oben*)
3. Helligkeit für unteres Display setzen (Adr. *Write-Display-Unten*)
4. Komponente wieder auf *active* schalten, wenn sie vorher auch *active* war

4.2 Einordnung in das Timing

IPS-Geräte arbeiten ausschließlich im *Commandmode* und benötigen daher kein Timing.

4.3 Festlegung von Startwerten

4.3.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset und ein Konfigurationsupdate durchgeführt.
- Alle Datenstrukturen werden initialisiert.
- Ein periodischer Auftrag (Periode 500 ms) zum update des Helligkeitsschalters wird gestartet.
- DRD- und DRQ- Interrupt werden am Gerät enabled.

4.3.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset und ein Konfigurationsupdate durchgeführt.
- Die Datenstrukturen für Istwerte werden initialisiert.
- Falls nicht schon geschehen, wird ein periodischer Auftrag (Periode 500 ms) zum update des Helligkeitsschalters gestartet.
- Falls noch nicht geschehen, wird für alle aktiven Potis eines Gerätes ein periodischer Auftrag (Periode 90 ms) gestartet.
- DRD- und DRQ- Interrupt werden am Gerät enabled.

4.4 Verhalten bei Störungen

4.4.1 Ausfall der Kommunikation EC – Gerät

Der Ausfall der Kommunikation zwischen EC und Gerät führt zu einem Timeout und beim nächsten turnusmäßigen Konfigurationsupdate wird das entsprechende Gerät als *offline* gemeldet.

4.5 Bedienungsfehler vom Operating

Alle von der Operating-Ebene kommenden Daten müssen auf Zulässigkeit geprüft und im Fehlerfall mit einer entsprechenden Fehlermeldung abgewiesen werden.

5 Die Repräsentation des Gerätes

Ein Gerät erscheint zur Benutzerebene im Umfang des in Abschnitt ?? definierten logischen Gerätes. Die eigentlichen IPS-Komponenten werden als Teil dieses Gerätes angesprochen. Die Identifizierung erfolgt über die Gruppennummer (0: Inkrementalgeber, 1: Tastenmodul).

5.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **IPS**.

Die Gerätemodellnummer ist 05_{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	–	36	BitSet8	1	0
INFOSTAT	RA	0	–	25	BitSet32	1	0
CONFIGUR	RA	0	–	288	BitSet16	1	0
ACTIVATE	RA/WA	2	BitSet16	2	BitSet16	1	0
DISPLAY	WA	3	BitSet16	8	BitSet8	1	0
VALUE	RA	1	BitSet16	32	Integer32	1	0
ALLOCATE	RA	1	BitSet16	16	BitSet16	1	0
ALLOCATE	W	2	BitSet16	1	BitSet16	1	0
POLLRATE	R/W	0	–	1	Integer32	s	-2
INCFACT	R/W	0	–	1	RealF	1	0

5.2.1 POWER

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

Parameter: Keine.

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

5.2.2 STATUS

Bedeutung: Auslesen des 32bit Gerätestatus.

Parameter: Keine.

Daten: Das 32bit Statuswort.

5.2.3 INIT

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

Parameter: Keine.

Daten: Keine.

5.2.4 RESET

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

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 CONFIGUR

- Bedeutung:** Lesen der Konfigurationstabelle eines IPS-Gerätes (IPS-Box).
- Parameter:** Keine.
- Daten:** Jeder der möglichen 32 IPS-Komponenten einer IPSBox (16 Inkrementalgeber; 16 Tastenmodule) ist ein fester Bereich von 9 Elementen in dem Datenarray zugeordnet. Die insgesamt 288 16-Bit Werte bedeuten im Einzelnen:
- 1...9:** Konfigurationsinformationen über den ersten Inkrementalgeber in der Box
 - 1. Online-Status des ersten Gerätes (0: offline, 1: online)
 - 2. Aktiv-Status des Gerätes (0: inaktiv, 1: aktiv)
 - 3. Geräteidentifikationsnummer (Highbyte: Gerätegruppe (0: Inkrementalgeber, 1: Tastenmodul), Lowbyte: Gerätenummer)
 - 4. Boxnummer (Adresse der Interfacekarte)
 - 5. Funktions-Id (Nummer, die beim ACTIVATE vom Benutzer gesetzt werden kann)
 - 6. VAX-Knotennummer des Prozesses, der das Gerät allokiert hat
 - 7. Lowbyte: Name des VAX-Prozesses (1-ter Buchstabe), der das Gerät allokiert hat
Highbyte: 2-ter Buchstabe
 - 8. Lowbyte: 3-ter Buchstabe Highbyte: 4-ter Buchstabe
 - 9. Lowbyte: 5-ter Buchstabe Highbyte: 6-ter Buchstabe
 - 10...18:** Konfigurationsinformationen über den zweiten Inkrementalgeber in der Box
 - ⋮
 - 135...143:** Konfigurationsinformationen über den 16-ten Inkrementalgeber in der Box
 - 144...152:** Konfigurationsinformationen über das erste Tastenmodul in der Box
 - ⋮
 - 280...288:** Konfigurationsinformationen über das 16-te Tastenmodul in der Box

5.2.8 ACTIVATE

- Bedeutung:** Aktivieren/Deaktivieren der Ein/Ausgabefunktion einer IPS-Komponente.
- Parameter:** Die beiden 16-Bit Werte bedeuten:
- 1. Gruppennummer der Komponente (0: Inkrementalgeber, 1: Tastenmodul)
 - 2. Komponentennummer (0 ... 15)
- Daten:** Die beiden 16-Bit Werte bedeuten:

1. Aktivzustand der Komponente (0:inaktiv, 1:aktiv)
2. Funktions-Id (diese Nummer wird beim Lesen der aktuellen Daten mit der Property VALUE für jeden Wert mitgeschickt, sodaß ein Benutzerprogramm die empfangenen Änderungsdaten einfach identifizieren und zuordnen kann)

5.2.9 DISPLAY

- Bedeutung:** Schreiben einer Zeichenkette auf eines der beiden acht-stelligen alphanumerischen Displays.
- Parameter:** Die drei 16-Bit Werte bedeuten:
1. Gruppennummer der Komponente (0: Inkrementalgeber, 1: Tastenmodul)
 2. Komponentennummer (0 . . . 15)
 3. Displaynummer (1: oberes Display am Gerät, 2: unteres Display)
- Daten:** 8 ASCII-Zeichen, die am Display zur Darstellung gebracht werden.

5.2.10 VALUE

- Bedeutung:** Lesen der Datenänderungen aller Komponenten einer Gruppe (0: Inkrementalgeber, 1: Tastenmodule) innerhalb einer IPS-Box, die vom lesenden Benutzer allokiert sind.
- Parameter:** Der BitSet16-Wert gibt die Gruppennummer der Komponente (0: Inkrementalgeber, 1: Tastenmodul) an.
- Daten:** Die bis zu 32 Integer32 Werte bedeuten:
1. OE_Id (Operating Element Identification)

Bit	Bedeutung
0 . . 7	Komponentennummer
8 . . 15	Gruppennummer (0: Inkrementalgeber, 1: Tastenmodul)
16 . . 23	Boxnummer
24 . . 31	Funktions-Id (vom letzten ACTIVATE)
 2. aktueller Wert (bei Inkrementalgeber der ganzzahlige Anteil der mit dem Bewertungsfaktor multiplizierten Inkremente). Bei Tastenmodulen hat dieser Wert keine Bedeutung und ist immer 0, da bei Tastern nur mitgeteilt werden soll, daß gedrückt worden ist.

5.2.11 ALLOCATE

- Bedeutung:** Lesen welche Komponenten allokiert sind. Es werden nur die Komponentennummern geliefert, die vom Aufrufer dieser Property (also dem VAX-Prozeß) allokiert sind.
- Parameter:** Der BitSet16-Wert gibt die Gruppennummer der Komponente an (0: Inkrementalgeber, 1: Tastenmodule).

Daten: Bis zu 16 OE_Ids (Operating Element Identification), wobei die einzelnen Bits folgende Bedeutung haben:

Bit	Bedeutung
0...7	Komponentennummer
8...15	Gruppennummer (0: Inkrementalgeber, 1: Tastenmodul)
16...23	Boxnummer
24...31	Funktions-Id (vom letzten ACTIVATE)

5.2.12 ALLOCATE

Bedeutung: Allokieren/Deallokieren einer oder mehrerer IPS-Komponenten. Nur der VAX-Prozeß, der eine Komponente allokiert hat, kann die Komponente auch benutzen.

Parameter: Die beiden 16-Bit Werte bedeuten:

1. Gruppennummer der Komponente (0: Inkrementalgeber, 1: Tastenmodul)
2. Komponentennummer (0...15) oder eine der folgenden

Wildcards:

Wildcard	Bedeutung
<i>FF_{Hex}</i>	alle Komponenten einer Gruppe (maximal 16)
<i>FE_{Hex}</i>	die nächste freie Komponente in der Gruppe
<i>Fx_{Hex}</i>	die nächsten x freien Komponenten in der Gruppe

Daten: 1 BitSet16-Wert gibt an, ob die Komponente allokiert (1) oder deallokiert (0) werden soll.

5.2.13 POLLRATE

Bedeutung: Gibt an, in welchem zeitlichen Abstand die aktivierten Inkrementalgeber abgefragt (gepollt) werden bzw. werden sollen.

Parameter: Keine.

Daten: 1 Integer32 Zeitabstand in 1/100 s

5.2.14 INCFACT

Bedeutung: Gibt an, mit welchem Faktor die Inkremente der Inkrementalgeber bewertet werden bzw. werden sollen.

Parameter: Keine.

Daten: 1 RealF Bewertungsfaktor.

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	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 IPS-Geräten ohne Bedeutung.

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 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
- \vdots
- $m + 1$:** letzte Master-Fehlermeldung
- $m + 2$:** erste Slave-Fehlermeldung
- \vdots
- $l + 1$:** 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. Bei IPS-Geräten ohne Bedeutung.
- Parameter:** Keine.
- Daten:** Nummer des virtuellen ('fremden') Beschleunigers, von dem die Einstellungen (Sollwerte) kopiert werden sollen.

Teil II

Der Entwurf der Software

6 Softwareentwurf

Keine erwähnenswerten Besonderheiten.

7 Lokale Datenbasis

In der lokalen Datenbasis muß lediglich die Zuordnung *Gerätenomenklatur* zu *Geräteadresse* festgelegt werden.

8 Dualport RAM

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

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 IPSgeräte folgende gerätespezifischen USRs benötigt:

9.2.1 R_Configur

Lesen der Konfigurationstabelle eines IPS-Gerätes.

9.2.2 R_Activate

Aktivierungszustand einer IPS-Komponente lesen.

9.2.3 W_Activate

Aktivierungszustand einer IPS-Komponente setzen.

9.2.4 W_Display

String auf oberem/unterem Display ausgeben.

9.2.5 R_Value

Änderungsdaten der IPS-Komponenten eines IPS-Gerätes lesen. Solange sich keine Änderungen ergeben, werden auch keine Daten geschickt. Diese USR wird von IPSACCESS an den IPS-Interrupt des EC konnektiert.

9.2.6 R_Allocate

Allokierungszustand aller IPS-Komponenten eines IPS-Gerätes lesen. Es werden nur die Geräte geliefert, die dem anfordernden Process auch gehören (also von ihm allokiert sind).

9.2.7 W_Allocate

IPS-Komponenten allokiieren (für einen Benutzer reservieren).

9.2.8 R_Pollrate

Eingestellte Pollrate für Potiupdates lesen (in 1/100 s).

9.2.9 W_Pollrate

Pollrate für Potiupdates einstellen (in 1/100 s). Bei der nächsten Aktivierung eines Potis wird mit der neuen Pollrate der Potizustand abgefragt, bereits laufende Aktivierungen sind von der Änderung nicht betroffen.

9.2.10 R_IncFact

Bewertungsfaktor für Inkrementalgeber lesen.

9.2.11 W_IncFact

Bewertungsfaktor für Inkrementalgeber setzen. Die vom Poti gelieferten Inkremente (128 pro Umdrehung) werden vom EC aus mit diesem Faktor *bewertet* (also multipliziert) und nur der ganzzahlige Anteil des Ergebnisses wird als Inkrement an die Operating-Ebene gesendet. Die Nachkommastellen werden gespeichert.

9.3 Globale Routinen

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

9.3.1 set_owner

Entnimmt aus dem Ethernet-Header des Empfangspaketes Node-Number und Process-Id und trägt sie in die Geräteverwaltung im DPRAM des EC ein.

9.3.2 reset_owner

Initialisiert Node-Number und Process-Id in der Geräteverwaltung im DPRAM des EC.

9.3.3 check_owner

Entnimmt aus dem Ethernet-Header des Empfangspaketes Node-Number und Process-Id und vergleicht sie mit der Geräteverwaltung im DPRAM des EC.

9.3.4 no_owner

Überprüft anhand Geräteverwaltung im DPRAM des EC, ob ein IPS-Gerät einen Owner hat oder nicht.

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.
- busy Das Gerät ist für Ein/Ausgabe-Aktionen aktiviert.
- error Während der Abarbeitung eines EQMs wurde ein Fehler erkannt.

10.1.2 Übergänge zwischen den Zuständen

Die Zustände und die Übergänge zwischen denselben sind in Tabelle ?? zusammengefaßt. Die Legende zu diesen Tabellen ist in Tabelle ?? zu finden.

Tabelle der Zustandsübergänge				
von nach →		error	ready	busy
error	U:	–	RESET	–
	A:	–	Reset_EQM	–
ready	U:	timeout etc.	–	ACTIVATE
	A:	div. EQMs	–	ACTIVATE_EQM
busy	U:	timeout etc.	DEACTIVATE	–
	A:	div. EQMs	ACTIVATE_EQM	–

Tabelle 1: Zustandsübergangsdiagramm

Legende

- Die Priorität der Zustände (höchste Priorität zuerst): error, ready, busy.
- U: Auslösende Ursache.
RESET Reset wird per Kommando oder Knöpfchendrücken ausgelöst.
- A: Ausführende Stelle des Zustandübergangs.
Status lesen (period.) Beim periodischen (oder zumindest regelmäßigen) Lesen des Status.
..._EQM Innerhalb des EQMs ..._EQM.

Tabelle 2: Legende zu den Zustandsübergangsdiagrammen

10.2 Periodisch konnektierte EQMs

10.2.1 Update_Brightness_EQM

Zeit: 500ms

Anzahl: Unendlich

Aktion: Die Helligkeitsschalter werden gelesen und evtl. die Helligkeit der alphanumerischen Displays aktualisiert.

10.2.2 Update_Devices_EQM

- Zeit:** Wird über die Property *POLLRATE* eingestellt. Per *default* werden *90ms* verwendet.
- Anzahl:** Solange irgendein Poti in der Box *aktiv* ist.
- Aktion:** Update der *aktiven* Potis und Taster. Bei Änderungen wird ein Interrupt an den G μ P ausgelöst.

10.2.3 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 An externe Interrupts konnektierte EQMs

10.3.1 DRD_EQM

- Interrupt:** Data Ready Interrupt.
- Aktion:** Update der *aktiven* Taster und Potis. Bei Änderungen wird ein Interrupt an den G μ P ausgelöst.

10.3.2 DRQ_EQM

- Interrupt:** Data Request Interrupt.
- Aktion:** Ist die Box, die den *Request* ausgelöst hat bereits bekannt, so wird das *Reset_EQM* ausgeführt. Ist die Box noch nicht bekannt, so wird das *System-EQM Update_Config* gestartet.

10.4 Kommandogetriggerte EQMs

10.4.1 Dev_Init_EQM

10.4.2 Dev_Reset_EQM

10.4.3 Status_EQM

10.4.4 Active_EQM

10.4.5 Power_EQM

10.4.6 Activate_EQM

10.4.7 Write_EQM

10.5 EQMs für die Diagnose vor Ort

10.5.1 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.

10.5.2 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.

10.5.3 Display_Grp_EQM

Parameter: Das EQM benötigt 2 Parameter.
1. Boxnummer (Physikalische Adresse der Interfacekarte (in Hex angeben)
2. Gruppennummer (0:Poti, 1:Taster) (in Hex angeben)

Daten: Keine.
Aktion: Zeigt am Bildschirm vor Ort die wichtigsten Daten aller Komponenten der gewählten Gerätegruppe an.

10.6 Sonstige EQMs

10.6.1 Startup_EQM

Schaltet die SE in den Command-Mode.

10.7 Globale Routinen

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

10.7.1 Read_and_Update_Status

Status aktualisieren.

10.7.2 Do_Intr_Service_Prep

Interrupts *scharf* machen.

10.7.3 TstBit

Einzelne its einer Maske auswerten.

10.7.4 Get_Value

Daten von einer IPS-Komponente lesen.

10.7.5 Set_Value

Daten an eine IPS-Komponente schicken.

10.7.6 Dev_Adr

Ermitteln der Adresse einer IPS-Komponente am Adressbus der IPS-Box.

10.7.7 dv\$box_reset

Reset-Befehl an eine IPS-Box schicken.

10.7.8 dv\$brightswitch

Lesen des Helligkeitsschalters.

10.7.9 dv\$interrupt

Lesen der Interruptmasken aller IPS-Komponenten einer Box.

10.7.10 dv\$devices

Lesen der Existenzinformationen der IPS-Komponenten.

10.7.11 dv\$activate

Aktivieren einzelner IPS-Komponenten.

10.7.12 dv\$deactivate

Deaktivieren einzelner IPS-Komponenten.

10.7.13 dv\$set_brightness

Helligkeit der Displays aller Kopponenten setzen.

10.7.14 dv\$read

Daten der IPS-Komponenten lesen.

10.7.15 dv\$write

Display (oberes oder unteres) einer IPS-Komponente beschreiben.

10.7.16 set_brightness

Helligkeit für alle Komponenten einer IPS-Box aktualisieren.

10.7.17 init_box

Box incl. aller Komponenten initialisieren.

10.7.18 update_devices

Alle Komponenten einer IPS-Box aktualisieren.

11 Besonderheiten

Als Besonderheit ist zu betrachten, daß nur die ganze IPS-Box mit allen Komponenten als Gerät im Sinne des Kontrollsystems verwaltet wird. Bei Zugriffen auf einzelne Komponenten muß jeweils die Gerätegruppe (Poti oder Taster) als Parameter angegeben werden. Die Nomenklatur der IPS-Box wird aus der physikalischen Adresse der Box und der Bezeichnung *IPSBOX* zusammengesetzt. Beispielsweise hat die Box mit der Adresse $1C_{Hex}$ die Nomenklatur *IPSBOX1C*.

Index

—A—

Abriß, 2
Activate_EQM, 24
Active_EQM, 24
Änderungsprotokoll, 2
An externe Interrupts konnektierte EQMs,
24
Aufgabe des Gerätes, 7

—B—

Bedienung des Gerätes, 11
Bedienungsfehler, 12
Besonderheiten, 27

—D—

Datenbasis, 21
Dev_Init_EQM, 24
Dev_Reset_EQM, 24
Display_DevErr_EQM, 25
Display_DPR_EQM, 24
Display_Grp_EQM, 25
DRD Interrupt, 10
DRD_EQM, 24
DRQ Interrupt, 10
DRQ_EQM, 24
Dualport RAM, 21

—E—

EQMs, 23
 An externe Interrupts konnektierte, 24
 DRD_EQM, 24
 DRQ_EQM, 24
 für die Diagnose vor Ort, 24
 Display_DevErr_EQM, 25
 Display_DPR_EQM, 24
 Display_Grp_EQM, 25
 Globale Routinen, 25
 Dev_Adr, 26
 Do_Intr_Service_Prep, 25
 dv\$activate, 26
 dv\$box_reset, 26
 dv\$brightswitch, 26
 dv\$deactivate, 26
 dv\$devices, 26
 dv\$interrupt, 26
 dv\$read, 26
 dv\$set_brightness, 26
 dv\$write, 26
 Get_Value, 25

init_box, 26
Read_and_Update_Status, 25
set_brightness, 26
Set_Value, 26
TstBit, 25
update_devices, 26
Kommandogetriggerte, 24
 Activate_EQM, 24
 Active_EQM, 24
 Dev_Init_EQM, 24
 Dev_Reset_EQM, 24
 Power_EQM, 24
 Status_EQM, 24
 Write_EQM, 24
Periodisch konnektierte, 23
 Update_Brightness_EQM, 23
 Update_Config_EQM, 24
 Update_Devices_EQM, 24
Sonstige, 25
 Startup_EQM, 25
Eventkonnektierungen, 11

—F—

Funktionscodes, 7
 ifb_addr_write, 8
 ifb_data_read, 8
 ifb_data_write, 8
 ifb_piggy_id, 8
 ifb_rdstat, 8
 ifb_reset, 8

—G—

Gerät
 Aufgabe, 7
 Bedienung, 11
 Hardware, 7
 logisches, 10
 Repräsentation, 12
 Schnittstelle, 7
Gerätemodell, 7
 Kennzeichnung, 12
 Master-Properties, 12
 Slave-Properties, 18
Globale Routinen, 22, 25

—H—

Hardware des Gerätes, 7
Hardwarestatus, 10
Helligkeitsschalter, 11

—I—

ifb_addr_write, 8
ifb_data_read, 8
ifb_data_write, 8
ifb_piggy_id, 8
ifb_rdstat, 8
ifb_reset, 8
Init, 11
Inkrementalgeber, 11
Interfacekarte, 7
Interne Zustände, 23
Interrupt
 DRD Interrupt, 10
 DRQ Interrupt, 10
IPS-Hardware, 8

—K—

Kaltstarts, 11
Kommandogetriggerte EQMs, 24
Konfigurationsabfrage, 10

—L—

logisches Gerät, 10
Lokale Datenbasis, 21

—M—

Master-Properties, 12

—N—

N_Init, 21
N_Reset, 21
Normalbetrieb, 11

—P—

Periodisch konnektierte EQMs, 23
Power_EQM, 24
Properties
 ACTIV, 18
 ACTIVATE, 16
 ALLOCATE, 17
 CONFIGUR, 15
 COPYSET, 20
 DISPLAY, 16
 EQMERROR, 18
 INCFAC, 18
 INFOS, 14
 INIT, 13
 Master-, 12
 POLLRATE, 17
 POWER, 13
 RESET, 13
 Slave-, 18

STATUS, 13
VALUE, 16
VERSION, 13

—R—

R_Activate, 21
R_Active, 21
R_Allocate, 22
R_Configur, 21
R_EQMErr, 21
R_IncFact, 22
R_InfoStat, 21
R_Pollrate, 22
R_Power, 21
R_Status, 21
R_Value, 22
R_Version, 21
Repräsentation des Gerätes, 12
Reset, 12

—S—

Schnittstelle zum Gerät, 7
Slave-Properties, 18
Softwareentwurf, 21
Sonstige EQMs, 25
Startup_EQM, 25
Startwerte, 11
Status_EQM, 24
Statusbits, 10
Störungen, 12
 Kommunikation EC – Gerät, 12

—T—

Taster, 11
Timing, 11

—U—

Update_Brightness_EQM, 23
Update_Config_EQM, 24
Update_Devices_EQM, 24
USRs, 21
 gerätespezifische, 21
 R_Activate, 21
 R_Allocate, 22
 R_Configur, 21
 R_IncFact, 22
 R_Pollrate, 22
 R_Value, 22
 W_Activate, 22
 W_Allocate, 22
 W_Display, 22
 W_IncFact, 22
 W_Pollrate, 22

Globale Routinen, 22
 check_owner, 23
 no_owner, 23
 reset_owner, 22
 set_owner, 22
obligatorische, 21
 N_Init, 21
 N_Reset, 21
 R_Active, 21
 R_EQMErr, 21
 R_InfoStat, 21
 R_Power, 21
 R_Status, 21
 R_Version, 21
 W_Active, 21
 W_CopySet, 21
 W_Power, 21

—**V**—

Varianten

 Betriebs-, 11

—**W**—

W_Activate, 22
W_Active, 21
W_Allocate, 22
W_CopySet, 21
W_Display, 22
W_IncFact, 22
W_Pollrate, 22
W_Power, 21
Warmstarts, 12
Write_EQM, 24

—**Z**—

Zustände

 Interne, 23
 Bedeutung, 23
 Übergänge, 23