

## Gerätemodelle für Terminalversorgung der Ionenquellen:

ISAU - Ion Source special Application Unit

ISDP - Ion Source Diagnose Preßluft

ISVT - Ion Source Vakuum Turbopumpen

ISEE - Ion Source Extraktions Elektroden

ISMO - Ion Source Motorsteuerung

P. Kainberger

*Dieses Papier enthält die Beschreibung der, auf der Titelseite genannten, Gerätemodelle und den Entwurf der Gerätesoftware für diese Geräte. Alle Teilkomponenten der Terminalversorgung, die physikalisch an einer Interfacekarte angeschlossen sind, werden in den 4 Gerätemodellen, mit jeweils unterschiedlichen Properties, als insgesamt 9 logische Geräteeinheiten abgebildet.*

<b>Änderungsprotokoll</b>			
Datum	GM-Version	Name	Kommentar
16. Jan. 95	–	P. Kainberger	erste Version
20. März 95	–	P. Kainberger	diverse Ergänzungen
18. Mai 95	–	P. Kainberger	Ux3IP1B heißt jetzt Ux3IP1X
16. Dez. 96	–	P. Kainberger	Statusbits für Preßluft und Turbopumpen geändert.
02. Feb. 00	–	U. Krause	Motorsteuerung aufgenommen
Juli 2000	–	M. Kühn	Überarbeitete und erweiterte T <sub>E</sub> X-Version, die sowohl in PostScript als auch in HTML konvertiert werden kann.

# Inhaltsverzeichnis

<b>I</b>	<b>Das Gerätemodell</b>	<b>5</b>
<b>1</b>	<b>Die Hardware des Gerätes</b>	<b>5</b>
<b>2</b>	<b>Die Schnittstelle zum Gerät</b>	<b>5</b>
2.1	Funktionscodes der Interfacekarte . . . . .	6
2.2	Terminalversorgung-Hardware . . . . .	7
2.2.1	Extraktionselektroden . . . . .	7
2.2.2	Turbopumpen . . . . .	8
2.2.3	Cup . . . . .	8
2.2.4	Vorvakuumpumpen . . . . .	8
2.2.5	Quellenidentifikation . . . . .	8
2.2.6	Medienversorgung . . . . .	8
2.3	Interlock Interrupt . . . . .	8
2.4	Data Request (DRQ) Interrupts . . . . .	8
2.5	Data Ready (DRD) Interrupts . . . . .	8
2.6	Umfang eines logischen Gerätes . . . . .	8
2.7	Definition der Bits des Hardwarestatus . . . . .	9
2.8	Konfigurationsabfrage . . . . .	10
<b>3</b>	<b>Die Bedienung des Gerätes</b>	<b>11</b>
3.1	Aufgaben im Normalbetrieb . . . . .	11
3.1.1	Extraktionselektroden . . . . .	11
3.1.2	Turbopumpen . . . . .	11
3.1.3	Cup . . . . .	11
3.1.4	Vorvakuumpumpen . . . . .	11
3.1.5	Quellenidentifikation . . . . .	11
3.1.6	Medienversorgung . . . . .	11
3.2	Einordnung in das Timing . . . . .	11
3.3	Festlegung von Startwerten . . . . .	11
3.3.1	Kaltstarts . . . . .	11
3.3.2	Warmstarts . . . . .	11
3.4	Ableitung des Hardwarewarning-Bits aus dem Gerätestatus . . . . .	12
3.5	Verhalten bei Störungen . . . . .	12
3.5.1	Ausfall der Kommunikation EC – Gerät . . . . .	12
<b>4</b>	<b>Die Repräsentation des Gerätes</b>	<b>12</b>
4.1	Kennzeichnung des Gerätemodells . . . . .	12
4.2	Die Master-Properties . . . . .	12
4.2.1	POWER . . . . .	13
4.2.2	STATUS . . . . .	13
4.2.3	INIT . . . . .	15
4.2.4	RESET . . . . .	15
4.2.5	VERSION . . . . .	15
4.2.6	INFOSTAT . . . . .	16
4.2.7	CONSTANT . . . . .	17
4.2.8	GAPRADIS . . . . .	18
4.2.9	GAPRADII . . . . .	18
4.2.10	GAPLONGS . . . . .	19
4.2.11	GAPLONGI . . . . .	19

4.2.12	REVOLUTI . . . . .	19
4.2.13	POSITS . . . . .	19
4.2.14	POSITI . . . . .	19
4.2.15	SOURCEID . . . . .	19
4.2.16	MOTPOSS . . . . .	19
4.2.17	MOTPOSI . . . . .	20
4.3	Die Slave-Properties . . . . .	20
4.3.1	ACTIV . . . . .	20
4.3.2	EQMERROR . . . . .	20
4.3.3	COPYSET . . . . .	21
<b>II</b>	<b>Der Entwurf der Software</b>	<b>23</b>
<b>5</b>	<b>Softwareentwurf</b>	<b>23</b>
<b>6</b>	<b>Lokale Datenbasis</b>	<b>23</b>
6.1	Tabelle der Konstanten . . . . .	23
<b>7</b>	<b>USRs - User Service Routinen</b>	<b>23</b>
<b>8</b>	<b>EQMs - Equipment Module</b>	<b>23</b>
	<b>Index</b>	<b>25</b>

## Teil I

# Das Gerätemodell

## 1 Die Hardware des Gerätes

Unter dem Begriff *Terminalversorgung der Ionenquellen* sind folgende Geräte und Eigenschaften des Quellenterminals zusammengefaßt.

**Extraktionselektroden:** Bedient wird eine Blende, die über DC-Motoren sowohl radial als auch longitudinal in Bezug auf den Strahl gefahren werden kann.

**Turbopumpen:** Zur Vakuumerzeugung.

**Vorvakuumpumpen:** Zur Vakuumerzeugung.

**Pressluft für eine Faradaycup:** Die Cup kann nur raus oder rein gefahren und die aktuelle Position gelesen werden.

**Quellenidentifikation:** Durch Aufstecken eines Kodiersteckers auf eine Buchse, wird eine Kennung der eingebauten Quelle zur Verfügung gestellt.

**Medienversorgung:** Der Status verschiedener Medien (Kühlanlage, Power Rack) wird hier zusammengeführt. Ebenso ist hier der Netzschalter für die Terminalversorgung angesiedelt.

Neben der Terminalversorgung, die als Teilkomponenten eines Gerätes realisiert sind, werden über die gleiche Gerätesoftware auch eigenständige Geräte bedient:

**Motorsteuerung:** Antrieb eines Stellmotors.

Sollwerte bzw. Kommandos gibt es für Faraday-Cup und Pumpenstand. Istwerte und Status gibt es für Faraday-Cup, Kühlanlage und Pumpenstand.

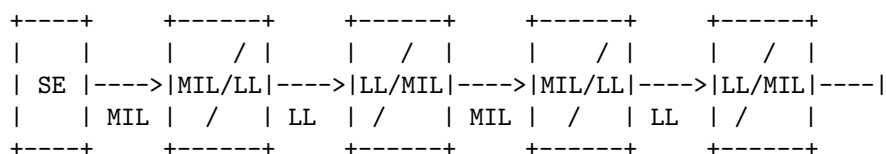
## 2 Die Schnittstelle zum Gerät

Alle Komponenten der Terminalversorgung sind an einer einzigen Interfacekarte angeschlossen. Die weiteren Geräte (derzeit nur die Motorsteuerung – Stand Feb. 2000) werden über jeweils eine eigene Interfacekarte angesteuert.

Wegen elektromagnetischer Umweltverschmutzung auf dem Terminal und dem Power Rack müssen alle Sollwerte periodisch erneuert werden. Die Rate wurde auf 200ms festgelegt.

Schaltsignale (Power, Reset, ...) sind in diesem Sinne keine Sollwerte, müssen also *nicht* periodisch versorgt werden. Wo es notwendig ist, wird entsprechende Hardware gebaut (z.B. Selbsthaltung nach dem Einschalten), um einen sicheren Zustand zu erreichen.

Da die Interfacekarte im ??? untergebracht ist und damit auf 50 kV Potential liegt, muß für die Verbindung zur SE eine Lichtleiterstrecke verwendet werden. Die MIL-Bus/Lichtleiter-Umsetzer, die in der Datentechnik entwickelt wurden, können mehrfach in einer Strecke eingesetzt werden:



Auch eine T-Abzweigung ist möglich:

```

+-----> MIL
| +-----+
+--| / |
|MIL/LL|-----> LL
MIL ---->| / |
+-----+

```

Dies eröffnet die Möglichkeit, Geräte eines Gerätemodells auf unterschiedlichen Potentialen mit *einer* SE anzusteuern.

## 2.1 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. In Klammern wird jeweils angegeben, für welches Gerätemodell und welche Property innerhalb des Modells der jeweilige Funktionscode benötigt wird.

Funktionscode		Modus	Bedeutung
Name	Hex		
ifb_reset	01	Funktion	Reset
ifb_power_on	02	Funktion	Netz einschalten (ISAU,POWER)
ifb_power_off	03	Funktion	Netz ausschalten (ISAU,POWER)
ifb_soll_gapradi	06	Schreiben	Sollwert radiale Lage (ISEE,GAPRADIS)
ifb_soll_gaplong	07	Schreiben	Sollwert longitudinale Lage (ISEE,GAPLONGS)
ifb_intr_mask	12	Schreiben	Interruptmaske setzen
ifb_posits_in	14	Funktion	Faradaycup Ein (ISDP,POSITS)
ifb_posits_out	15	Funktion	Faradaycup Aus (ISDP,POSITS)
ifb_ist_gapradi	81	Lesen	Istwert radiale Lage (ISEE,GAPRADIS)
ifb_ist_gaplong	82	Lesen	Istwert longitudinale Lage (ISEE,GAPLONGS)
ifb_ist_revolut_1	83	Lesen	Istwert Drehzahl Turbopumpe 1 (ISVT,REVOLUTI)
ifb_ist_revolut_2	84	Lesen	Istwert Drehzahl Turbopumpe 2 (ISVT,REVOLUTI)
ifb_ist_revolut_3	85	Lesen	Istwert Drehzahl Turbopumpe 3 (ISVT,REVOLUTI)
ifb_ist_revolut_4	86	Lesen	Istwert Drehzahl Turbopumpe 4 (ISVT,REVOLUTI)
ifb_ist_sourceid	87	Lesen	Quellenidentifikation (ISAU,SOURCEID)
ifb_rdstat	C0	Lesen	Gerätestatus, 1. Byte lesen
ifb_rdstat_1	C1	Lesen	Gerätestatus, 2. Byte lesen
ifb_rdstat_2	C2	Lesen	Gerätestatus, 3. Byte lesen
ifb_rdstat_int	C9	Lesen	Status der Interfacekarte lesen

### ifb\_reset

Das Gerät wird in einen definierten Ausgangszustand gebracht.

Bisher ist nicht bekannt, ob irgendeine angeschlossene Hardware auf *Reset* reagiert. Auf der Interfacekarte selbst soll (laut R.Hartmann) nichts passieren.

### ifb\_power\_on, ifb\_power\_off

Einschalten bzw. Ausschalten des Gerätes.

Diese Funktion ist nur für das Gerät **Medienversorgung** (UR3IM1P und UL3IM1P) im Geräte-  
modell *ISAU* ausführbar.

#### **ifb\_soll\_gapradi**

Sollwert radiale Lage der Extraktionselektroden setzen. Diese Funktion ist nur für das Gerät  
**Extraktionselektroden** (UR3IP1X und UL3IP1X) im Gerätemodell *ISEE* ausführbar.

#### **ifb\_soll\_gaplong**

Sollwert longitudinale Lage der Extraktionselektroden setzen. Diese Funktion ist nur für das Gerät  
**Extraktionselektroden** (UR3IP1X und UL3IP1X) im Gerätemodell *ISEE* ausführbar.

#### **ifb\_posits\_in**

Faradaycup auf Endlage innen fahren (über Preßluftantrieb). Diese Funktion ist nur für das Gerät  
**Cup** (UR3DC4\_P und UL3DC4\_P) im Gerätemodell *ISDP* ausführbar.

#### **ifb\_posits\_out**

Faradaycup auf Endlage außen fahren (über Preßluftantrieb). Diese Funktion ist nur für das Gerät  
**Cup** (UR3DC4\_P und UL3DC4\_P) im Gerätemodell *ISDP* ausführbar.

#### **ifb\_ist\_gapradi**

Istwert radiale Lage der Extraktionselektroden lesen. Diese Funktion ist nur für das Gerät  
**Extraktionselektroden** (UR3IP1X und UL3IP1X) im Gerätemodell *ISEE* ausführbar.

#### **ifb\_ist\_gaplong**

Istwert longitudinale Lage der Extraktionselektroden lesen. Diese Funktion ist nur für das Gerät  
**Extraktionselektroden** (UR3IP1X und UL3IP1X) im Gerätemodell *ISEE* ausführbar.

#### **ifb\_ist\_revolut\_1,...,ifb\_ist\_revolut\_4**

Istwert Drehzahl Turbopumpe 1..4 lesen. Diese Funktion ist nur für die Geräte **Turbopumpen**  
(UR3VP1T, UR3VP2T, UR3VP3T, UR3VP4T, UL3VP1T, UL3VP2T, UL3VP3T und UL3VP4T)  
im Gerätemodell *ISVT* ausführbar.

#### **ifb\_ist\_sourceid**

Quellenidentifikation lesen. Diese Funktion ist nur für das Gerät **Quellenidentifikation**  
(UR3IQ1I und UL3IQ1I) im Gerätemodell *ISAU* ausführbar.

## **2.2 Terminalversorgung-Hardware**

### **2.2.1 Extraktionselektroden**

Die longitudinale und radiale Soll- bzw. Istposition sind über DACs (??? wieviel Bit 12, 16) bzw.  
ADCs (??? wieviel Bit 12, 16) setz- bzw. lesbar. Die 0–10V der DACs bzw. ADCs entsprechen  
0–??? mm Soll- bzw. Istposition.

### 2.2.2 Turbopumpen

Für jede der 4 Turbopumpen, kann über ein Statusbit gelesen werden, ob die Pumpe 80% ihrer maximalen Drehzahl erreicht hat oder nicht. Außerdem kann über ADCs (??? wieviel Bit 12, 16) die Drehzahl einer Turbopumpe in 1/??? gelesen werden.

### 2.2.3 Cup

Ein Faradaycup kann als Beamdump in den Strahlweg ein bzw. aus dem Strahlweg ausgefahren werden. Über 2 Bits im Status kann die aktuelle Position ermittelt werden (drin, draußen oder keine Endlage). Eine Strommessung ist bisher nicht vorgesehen.

### 2.2.4 Vorvakuumpumpen

Die Vorvakuumpumpen liefern 1 Statusbit mit der Information *ein* oder *aus* (Pumpe läuft oder läuft nicht).

### 2.2.5 Quellenidentifikation

Über einen ???-poligen Kodierstecker, der beim Einbau eines Quellenmoduls auf eine entsprechende Buchse aufgesteckt wird, kann das eingebaute Quellenmodul identifiziert werden.

Diese Identifikation hat nichts mit der Unterscheidung zwischen verschiedenen Quellentypen (CORDIS, MEVVA, ...) zu tun, sondern identifiziert nur das verwendete Quellenmodul.

### 2.2.6 Medienversorgung

Unter dem Begriff *Medienversorgung* sind die Kühlanlage und das Power Rack zusammengefaßt. Neben dem Ein/Ausschalter des Power Racks stehen noch einige Statusbits zur Verfügung.

## 2.3 Interlock Interrupt

Bisher sind keine Komponenten der Terminalversorgung bekannt, die einen Interlock Interrupt auslösen könnten.

## 2.4 Data Request (DRQ) Interrupts

Wird nicht generiert.

## 2.5 Data Ready (DRD) Interrupts

Wird nicht generiert.

## 2.6 Umfang eines logischen Gerätes

Hinter einer einzigen Interfacekarte verbergen sich mehrere Geräte, die auf insgesamt 9 logische Geräteeinheiten abgebildet werden müssen.

Das Kontrollsystem bietet dafür die Möglichkeit, mit Hilfe der Konstante *IFB\_GroupCount* (in diesem Fall  $IFB\_GroupCount = 9$ ) dem MIL-Treiber der Systemsoftware mitzuteilen, wieviele logische Geräteeinheiten aus einer Interfacekarte gebildet werden sollen.

Nach den Vereinbarungen im Kontrollsystem muß für dieses Verfahren die physikalische Adresse der Interfacekarte ein Vielfaches von *IFB\_GroupCount* sein (also 9, 18, 27, ...). Die MIL-Treiber im Kontrollsystem sorgen automatisch dafür, daß jedes Gerät mit einer eigenen physikalischen Adresse angesprochen werden kann. Damit ist möglich, jeder einzelnen Komponente eine eigene Nomenklatur zuzuweisen.



Die Adressen der einzelnen Komponenten ergeben sich aus der Grundadresse der Interfacekarte und folgendem Offset:

Nomenklatur	Beschreibung	Offset zur Grundadresse
UL3IP1X	Extraktionselektroden	0
UL3VP1T	Turbopumpe 1	1
UL3VP2T	Turbopumpe 2	2
UL3VP3T	Turbopumpe 3	3
UL3VP4T	Turbopumpe 4	4
UL3DC4_P	Preßluftantrieb f. Faradaycup	5
UL3VP1R	Vorvakuumpumpe	6
UL3IQ1I	Quellenidentifikation	7
UL3IM1P	Medienversorgung	8

## 2.7 Definition der Bits des Hardwarestatus

Die Interfacekarte der Terminalversorgung liefert 3 Bytes Statusinformation. Die Funktionscodes zum Lesen der entsprechenden Bytes sind in der folgenden Tabelle zusammengefaßt.

Statusbits	Funktionscode
8 ... 15	ifb_rdstat
16 ... 23	ifb_rdstat_1
24 ... 31	ifb_rdstat_2

Die Statusbits im Einzelnen sind in der folgenden Tabelle zusammengefaßt.

	Name	Bedeutung	
		High (1)	Low (0)
8	unbenutzt	immer ok	–
9	unbenutzt	immer ok	–
10	unbenutzt	immer ok	–
11	Cup draußen	ja	nein
12	Cup drinnen	ja	nein
13	unbenutzt	immer ok	–
14	unbenutzt	immer ok	–
15	unbenutzt	immer ok	–
16	Vorpumpen	ein	aus
17	unbenutzt	immer ok	–
18	Power Rack	ein	aus
19	Kühlwasser P1	ok	Fehler
20	Kühlwasser P2	ok	Fehler
21	Vorlauftemperatur	ok	zu hoch
22	Rücklauftemperatur	ok	zu hoch
23	Durchfluß Kühlanlage	ok	zu gering
24	Durchfluß Ionentauscher	ok	zu gering
25	Kühlanlage	ok	Summenfehler
26	Kühlanlage	ein	aus
27	Pumpe 1 80% Drehzahl	erreicht	unterschritten
28	Pumpe 2 80% Drehzahl	erreicht	unterschritten
29	Pumpe 3 80% Drehzahl	erreicht	unterschritten

Fortsetzung auf nächster Seite

Fortsetzung			
	Name	Bedeutung	
		High (1)	Low (0)
30	Pumpe 4 80% Drehzahl	erreicht	unterschritten
31	Steuerung	Rechner	Hand

Die Statusbits der Motorsteuerung haben die folgende Bedeutung:

	Name	Bedeutung	
		High (1)	Low (0)
8	unbenutzt	immer ok	–
9	remote/local	remote	local
10	unbenutzt	immer ok	–
11	unbenutzt	immer ok	–
12	unbenutzt	immer ok	–
13	unbenutzt	immer ok	–
14	Endlage innen	erreicht	nicht err.
15	Endlage aussen	erreicht	nicht err.
16	unbenutzt	immer ok	–
17	unbenutzt	immer ok	–
	...	immer ok	–
31	unbenutzt	immer ok	–

Die Beschreibung der Statusbits, wie sie von der Property `STATUS` der einzelnen Gerätemodelle geliefert werden, sind in Abschnitt 4.2.2 auf Seite 13 beschrieben.

## 2.8 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.

Da mit Hilfe der Systemeinstellung `IFB_GroupCount=9` aus einer Interfacekarte 9 logische Geräteeinheiten erzeugt werden, sind automatisch mit einer Interfacekarte (deren Adresse ein Vielfaches von `IFB_Group_Count` sein muß) alle 9 logischen Geräte vorhanden und verfügbar. Ist diese eine Interfacekarte ausgeschaltet, sind alle 9 logischen Geräte *offline* und damit nicht mehr verfügbar.

Damit die zusätzlichen Einzelgeräte (wie die Motorsteuerung) nicht zu mehrfachen Geräteeinträgen führen, wird eine gerätmodellvariantenspezifische Online-Erkennung implementiert:

- Alle physikalischen Interfacekarten-Adressen müssen vielfache des `IFB_GroupCount` (also Vielfache von 9) sein.
- Für IFB-Adressen oberhalb einer Grenze (derzeit 206) werden für jede physikalische IFB-Adresse auch die folgenden (Anzahl über `IFB_GroupCount` festgelegt) Adressen als online geführt.
- Bei IFB-Adressen bis zu dieser Grenze wird nur die physikalisch gefundene Adresse (die Vielfaches von `IFB_GroupCount` sein muss!) als online geführt.

## 3 Die Bedienung des Gerätes

### 3.1 Aufgaben im Normalbetrieb

#### 3.1.1 Extraktionselektroden

Die Extraktionselektroden, werden in ihre longitudinale und radiale Sollposition gefahren und die entsprechenden Istwerte erfaßt.

#### 3.1.2 Turbopumpen

Im Normalbetrieb wird von Turbopumpen nur der Status überwacht.

#### 3.1.3 Cup

Die Faradaycup kann nur rein- oder rausgefahren werden. Die Istposition wird über die entsprechenden Statusbits überwacht.

#### 3.1.4 Vorvakuumpumpen

Im Normalbetrieb wird von Vorvakuumpumpen nur der Status überwacht.

#### 3.1.5 Quellenidentifikation

Im Normalbetrieb wird die Quellenidentifikation nur auf Veranlassung durch das Operating gelesen.

#### 3.1.6 Medienversorgung

Im Normalbetrieb wird von der Medienversorgung nur der Status überwacht (Kühlanlage und Power Rack). Nur über die Medienversorgung kann das Power Rack ein- oder ausgeschaltet werden.

### 3.2 Einordnung in das Timing

Alle Komponenten der Terminalversorgung arbeiten ausschließlich im *Commandmode* und benötigen daher kein Timing.

### 3.3 Festlegung von Startwerten

#### 3.3.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt:

- Alle gerätespezifischen Konstanten werden initialisiert und als *invalid* gekennzeichnet.
- Es wird ein Gerätereset durchgeführt.
- Alle Sollwerte werden auf 0 gesetzt.

#### 3.3.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Die gerätespezifischen Konstanten werden als *invalid* gekennzeichnet.
- Es wird ein Gerätereset durchgeführt.

### 3.4 Ableitung des Hardwarewarning-Bits aus dem Gerätestatus

Bisher sind keine Bits im Status der Geräte bekannt, die als Hardwarewarning angezeigt werden sollen.

### 3.5 Verhalten bei Störungen

#### 3.5.1 Ausfall der Kommunikation EC – Gerät

Der Ausfall der Kommunikation zwischen EC und Gerät führt zu Timeouts bei Lesezugriffen. Im Rahmen der zyklischen Konfigurationsüberwachung durch die Systemsoftware des Kontrollsystems werden alle Geräte innerhalb von 4 Minuten *offline*.

## 4 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).

Die Komponenten der Terminalversorgung erscheinen zur Benutzerebene im Umfang der in Abschnitt 2.6 definierten logischen Geräteeinheiten.

### 4.1 Kennzeichnung des Gerätemodells

Die Gerätemodelle zur Steuerung der Komponenten der Terminalversorgung haben folgende Bezeichnungen und Nummern:

Bezeichnung	Komponenten	Nummer
ISAU	Medienversorgung, Quellenidentifikation und Vorvakuumumpumpen	58
ISDP	Cup mit Preßluftantrieb	59
ISVT	Vakuum Turbopumpen	60
ISEE	Extraktionselektroden	61

### 4.2 Die Master-Properties

Master-Properties							
Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
Standard-Properties für alle Komponenten:							
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	RA	0	–	120	RealF	1	0
Extraktionselektroden:							
GAPRADIS	R/W	0	–	1	RealF	m	-3
GAPRADII	R	0	–	1	RealF	m	-3
GAPLONGS	R/W	0	–	1	RealF	m	-3
GAPLONGI	R	0	–	1	RealF	m	-3

Forts. auf nächster Seite

Forts. von letzter Seite

Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
Turbopumpen:							
REVOLUTI	R	0	-	1	RealF	1/min	3
Cup:							
POSITS	R/W	0	-	1	BitSet16	1	0
POSITI	R	0	-	1	BitSet16	1	0
Quellenidentifikation:							
SOURCEID	R	0	-	1	RealF	1	0
Motorsteuerung:							
MOTPOSS	R/W	0	-	1	RealF	1	0
MOTPOSI	R	0	-	1	RealF	1	0

#### 4.2.1 POWER

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

Nur die *Medienversorgung* (UL3IM1P u. UR3IM1P) kann ein/aus-geschaltet werden und liefert echte ein/aus-Information. Alle anderen Komponenten sind immer *ein* und können nicht ausgeschaltet werden.

**Parameter:** Keine.

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

#### 4.2.2 STATUS

**Bedeutung:** Auslesen des 32bit Gerätestatus.

**Parameter:** Keine.

**Daten:** Das 32bit Statuswort. Die Reihenfolge der Bits entspricht der in Abschnitt 2.7 auf Seite 9 und in der Tabelle 2.7 auf Seite 9 erklärten.

Die Bedeutungen für High (= 1) und Low (= 0) sind so gewählt, daß der normale Betriebszustand mit 1 gekennzeichnet ist.

Die Bits 0 . . . 7 sind die systemweiten, sogenannten generierten Softwarestatusbits (in engl. derived status bits) und gelten gleichermaßen für alle Komponenten der Terminalversorgung.

	Name	Bedeutung	
		High (1)	Low (0)
0	Power	on	off
1	Remote/Local	Remote	Local
2	reserved		
3	reserved		
Fortsetzung auf nächster Seite			

Fortsetzung			
	Name	Bedeutung	
		High (1)	Low (0)
4	Emergency	no	yes
5	Interlock	no	yes
6	HW Error	no	yes
7	SW Error	no	yes

Die Statusbits, die bei einzelnen Komponenten zusätzlich belegt sind, sind in der folgenden Tabelle zusammengefaßt. Alle nicht aufgelisteten Bits werden softwaremäßig immer 1 gesetzt. Die Bits gelten jeweils nur für den angegebenen Gerätetype und sind sonst immer 1.

	Name	Bedeutung	
		High (1)	Low (0)
Cup:			
11	Cup draußen	ja	nein
12	Cup drinnen	ja	nein
Vorkuumpumpen:			
16	Vorkuumpumpen	ein	aus
Medienversorgung:			
18	Power Rack	ein	aus
19	Kühlwasser P1	ok	Fehler
20	Kühlwasser P2	ok	Fehler
21	Vorlauftemperatur	ok	zu hoch
22	Rücklauftemperatur	ok	zu hoch
23	Durchfluß Kühlanlage	ok	zu gering
24	Durchfluß Ionentauscher	ok	zu gering
25	Kühlanlage	ok	Summenfehler
26	Kühlanlage	ein	aus
Turbopumpen:			
27	Pumpe 1 80% Drehzahl	erreicht	unterschritten
28	Pumpe 2 80% Drehzahl	erreicht	unterschritten
29	Pumpe 3 80% Drehzahl	erreicht	unterschritten
30	Pumpe 4 80% Drehzahl	erreicht	unterschritten
Für alle Komponenten gilt:			
31	Steuerung	Rechner	Hand

Die Statusbelegung der Motorsteuerung ist:

	Name	Bedeutung	
		High (1)	Low (0)
0	Power	on	-
Fortsetzung auf nächster Seite			

Fortsetzung			
	Name	Bedeutung	
		High (1)	Low (0)
1	Remote/Local	Remote	Local
2		reserved	
3		reserved	
4	Emergency	no	yes
5	Interlock	no	-
6	HW Error	no	-
7	SW Error	no	yes
8		unbenutzt (OK)	
9	Remote/Local	Remote	Local
10		unbenutzt (OK)	
11		unbenutzt (OK)	
12		unbenutzt (OK)	
13		unbenutzt (OK)	
14	Endlage innen	erreicht	nicht erreicht
15	Endlage aussen	erreicht	nicht erreicht
16		unbenutzt (OK)	
		...	
31		unbenutzt (OK)	

#### 4.2.3 INIT

**Bedeutung:** Initialisierung des Gerätes (Kaltstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 3.3.1 auf Seite 11.

**Parameter:** Keine.

**Daten:** Keine.

#### 4.2.4 RESET

**Bedeutung:** Reset des Gerätes (Warmstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 3.3.2 auf Seite 11.

**Parameter:** Keine.

**Daten:** Keine.

#### 4.2.5 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

#### 4.2.6 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.

#### 4.2.7 CONSTANT

**Bedeutung:** Diese Property liefert die gerätespezifischen Konstanten, mit denen die einzelnen Properties des Gerätes eingestellt wurden. Z.B. Funktionscodes, Statusmasken,...

**Parameter:** Keine.

**Daten:** Die 120 Realwerte enthalten im Einzelnen:

- 1: `power_sel_mask`: Bitmaske, die angibt, welche Bits im 32-Bit Status den Ein/Aus-Zustand des Gerätes anzeigen.
- 2: `power_val_mask`: Bitmaske, die angibt, welche Werte die Bits in der `power_sel_mask` im 'Ein'-Zustand des Gerätes haben.
- 3: `remote_sel_mask`: Bitmaske, die angibt, welche Bits im 32-Bit Status den Rechner/Hand-Zustand des Gerätes anzeigen.
- 4: `remote_val_mask`: Bitmaske, die angibt, welche Werte die Bits in der `remote_sel_mask` im 'Rechner'-Zustand des Gerätes haben.
- 5: `power_pulse_length`: Zeit in ms, gibt an, wie lange die Ein/Aus-Funktionscodes am Gerät gehalten werden müssen.
- 6: `poll_fct_code`: Funktionscode zum Lesen eines Statusbyte, aus dem der Ein/Aus-Zustand des Gerätes ermittelt werden kann.
- 7: `poll_sel_mask`: Bitmaske, die angibt, welche Bits im (mit dem `poll_fct_code` gelesenen) Statusbyte den Ein/Aus-Zustand des Gerätes anzeigen.
- 8: `poll_val_mask`: Bitmaske, die angibt, welche Werte die Bits in der `poll_sel_mask` im 'Ein'-Zustand des Gerätes haben.
- 9: `poll_rate`: Pollrate in s, gibt an in welchem Zeitabstand nach dem Ein/Ausschalten des Gerätes mit den gegebenen Pollparametern der erfolgreiche Abschluß der Schaltfunktion überprüft werden soll.
- 10: `poll.count`: maximale Anzahl der Pollversuche..
- 11 ... 15: Gerätespezifische Einstellungen für Sollwert 1:
- 11: `min_value`: minimaler Sollwert.
- 12: `max_value`: maximaler Sollwert.

- 13:** dac\_max: maximaler Wert am DAC.
- 14:** dac\_offset: konstanter Offset des DAC.
- 15:** fct\_code: Funktionscode, mit dem der Sollwert ans Gerät geschickt werden muß.
- 16 ... 20:** Entsprechend Punkt 11 ... 15, aber für Sollwert 2.
- ⋮
- 46 ... 50:** Entsprechend Punkt 11 ... 15, aber für Sollwert 10.
- 51 ... 54:** Gerätespezifische Einstellungen für Istwert 1:
- 51:** max\_value: maximaler Istwert.
- 52:** adc\_max: maximaler Wert am ADC.
- 53:** adc\_offset: konstanter Offset des ADC.
- 54:** fct\_code: Funktionscode, mit dem der Istwert vom Gerät gelesen werden muß.
- 55 ... 58:** Entsprechend Punkt 51 ... 54, aber für Istwert 2.
- ⋮
- 87 ... 90:** Entsprechend Punkt 51 ... 54, aber für Istwert 10.
- 91 ... 92:** Gerätespezifische Einstellungen für Schaltfunktion 1:
- 91:** fct\_code: Funktionscode, der als Schaltfunktion ans Gerät geschickt werden muß.
- 92:** length: Zeit in ms, gibt an, wie lange der Schaltfunktionscode am Gerät gehalten werden muß.
- 93 ... 94:** Entsprechend Punkt 91 ... 92, aber für Schaltfunktion 2.
- ⋮
- 109 ... 110:** Entsprechend Punkt 91 ... 92, aber für Schaltfunktion 10.
- 111 ... 120:** Reserviert für Erweiterungen.

#### 4.2.8 GAPRADIS

**Bedeutung:** Lesen bzw. Setzen der gewünschten radialen Lage einer Blende.

**Parameter:** Keine.

**Daten:** Das Datum gibt die Position der Blende in Bezug auf "irgend einer gedachte Mittellinie" (laut P.Spädtke) in mm an.

#### 4.2.9 GAPRADII

**Bedeutung:** Lese die aktuelle radiale Lage einer Blende.

**Parameter:** Keine.

**Daten:** Das Datum gibt die Position der Blende in Bezug auf "irgend einer gedachte Mittellinie" (laut P.Spädtke) in mm an.

#### 4.2.10 GAPLONGS

**Bedeutung:** Lesen bzw. Setzen der gewünschten longitudinalen Lage einer Blende.

**Parameter:** Keine.

**Daten:** Das Datum gibt den Abstand der beiden Extraktor-Elektroden in mm an.

#### 4.2.11 GAPLONGI

**Bedeutung:** Lese die aktuelle longitudinale Lage einer Blende.

**Parameter:** Keine.

**Daten:** Das Datum gibt den Abstand der beiden Extraktor-Elektroden in mm an.

#### 4.2.12 REVOLUTI

**Bedeutung:** Lese die aktuelle Drehzahl einer Turbopumpe.

**Parameter:** Keine.

**Daten:** Das Datum gibt die Drehzahl der Turbopumpe in 1/min an.

#### 4.2.13 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*.

#### 4.2.14 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)

#### 4.2.15 SOURCEID

**Bedeutung:** Lese die aktuelle Identifikationsnummer der eingebauten Quelle.

**Parameter:** Keine.

**Daten:** Das Datum gibt die Identifikationsnummer der Quelle an.

#### 4.2.16 MOTPOSS

**Bedeutung:** Schreiben und Lesen des Sollwertes für den Stellmotor.

**Parameter:** Keine.

**Daten:** Position im Bereich 0...10.

#### 4.2.17 MOTPOSI

**Bedeutung:** Lesen des Istwertes des Stellmotors.

**Parameter:** Keine.

**Daten:** Position im Bereich 0...10.

### 4.3 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

#### 4.3.1 ACTIV

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

Da alle Komponenten der Terminalversorgung DC-Geräte sind, liefert der Versuch ACTIV zu schreiben einen entsprechenden Fehler. Der Versuch ACTIV zu lesen liefert eine entsprechende *Information*.

**Parameter:** Keine.

**Daten:** Das Datum hat immer den Wert 1.

#### 4.3.2 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

### 4.3.3 COPYSET

**Bedeutung:** Kopiert alle Geräteeinstellungen (Sollwerte) eines virtuellen ('fremden') Beschleunigers in den zugehörigen ('eigenen') Beschleuniger.

Da alle Komponenten der Terminalversorgung DC-Geräte sind, liefert der Versuch COPYSET zu schreiben eine entsprechende *Information*.

**Parameter:** Keine.

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



## Teil II

# Der Entwurf der Software

## 5 Softwareentwurf

Die Geräte der Terminalversorgung sind allesamt *primitive* DC-Geräte, die an das Kontrollsystem keine besonderen Ansprüche stellen. Deshalb kann auf der SE-Ebene Standard-Software verwendet werden, die keine gerätespezifischen Eigenheiten fest verdrahtet hat.

Auf der G $\mu$ P-Ebene (USRs und ihre Properties) kann für jeden Gerätetyp innerhalb der Terminalversorgung durch eine eigene Konfigurationsdatei ein entsprechendes Gerätemodell (*ISAU*, *ISDP*, *ISVT* u. *ISEE*) zusammengestellt werden. Dafür stellt das Kontrollsystem die sog. *Standard-USRs* zur Verfügung.

Eine Beschreibung der eben erwähnten Programme ist in [?] zu finden.

## 6 Lokale Datenbasis

### 6.1 Tabelle der Konstanten

Die Struktur der Konstantentabelle ist durch die Verwendung von ‘Standard-USRs’ vorgegeben und wird hier nicht explizit erläutert.

Eine detaillierte Beschreibung der Strukturen ist in [?] zu finden.

## 7 USRs - User Service Routinen

Für die Properties der Komponenten der Terminalversorgung sind keine speziellen USRs nötig. Zum Einsatz kommt eine Konfiguration der ‘Standard-USRs’.

Eine detaillierte Beschreibung der ‘Standard-USRs’ ist in [?] zu finden.

Folgende Liste gibt an, mit welcher ‘Standard-USR’ die jeweiligen Properties der Gerätemodelle realisiert werden:

Gerätemodell	Property	Standard-USR
ISAU	POWER	STD\$W_Power\$USR
	SOURCEID	STD\$R_Ist1\$USR
ISDP	POSITS	STD\$W_Soll1\$USR
	POSITI	STD\$R_Ist1\$USR
ISVT	REVOLUTI	STD\$R_Ist1\$USR
ISEE	GAPRADIS	STD\$W_Soll1\$USR
	GAPRADII	STD\$R_Ist1\$USR
	GAPLONGS	STD\$W_Soll2\$USR
	GAPLONGI	STD\$R_Ist2\$USR
alle	CONSTANT	STD\$R_Constant\$USR
	COPYSET	STD\$W_Copysset\$USR

Alle nicht aufgeführten Properties werden mit ‘Default-USRs’ versorgt.

## 8 EQMs - Equipment Module

Zur Steuerung der Komponenten der Terminalversorgung sind keine speziellen EQMs nötig. Zum Einsatz kommt die Variante *ISAU* der ‘Standard-EQMs’.

Besonderheiten der ISAU-Variante:

- *IFB\_GroupCount = 9* : Damit werden aus einer Interfacekarte 9 logische Geräteeinheiten.
- Eine eigene Online-Erkennung sorgt aber dafür, dass mehrere logische Geräte pro Interfacekarte nur für Adressen ab *min\_mult\_dev\_addr* (207) eingetragen werden.

Für Adressen bis zu dieser Grenze wird nur ein logisches Gerät pro IFK online! Trotzdem muss die IFB-Adresse Vielfache des *IFB\_GroupCount* betragen.

- Für jede Geräteadresse innerhalb einer *Group* (1..9) ist eine *Status\_Sel\_Mask* eingestellt, in der alle Bits im Status deklariert sind, die für dieses Gerät (und damit nur für diese Nomenklatur) eine Bedeutung haben. Diese Maske selektiert jeweils genau die Statusbits, die in der Beschreibung der Property *STATUS* auf Seite ??

Folgende Masken sind deklariert:

Nummer	Gerät	Maske (Hex)
1	Extraktionselektrode	800000FF
2	Turbopumpe 1	880000FF
3	Turbopumpe 2	900000FF
4	Turbopumpe 3	A00000FF
5	Turbopumpe 4	C00000FF
6	Cup m. Preßluftantrieb	800018FF
7	Vorvakuumpumpe	800100FF
8	Quellenidentifikation	800000FF
9	Medienversorgung	87FC00FF

- *periodic\_data\_update = TRUE* : Damit werden alle Sollwerte von allen Geräten periodisch (alle 200 ms) an die Geräte geschickt.



# Index

## —Symbole —

Änderungsprotokoll ..... 2

## —A—

Abriß ..... 2

## —B—

Bedienung des Gerätes ..... 11

## —C—

Cup ..... 8, 11

## —D—

Datenbasis ..... 23

DRD Interrupt ..... 8

DRQ Interrupt ..... 8

## —E—

EQMs ..... 23

Extraktionselektroden ..... 7, 11

## —F—

Funktionscodes ..... 6

- ifb\_ist\_gaplong ..... 7
- ifb\_ist\_gapradi ..... 7
- ifb\_ist\_revolut\_1 ..... 7
- ifb\_ist\_revolut\_2 ..... 7
- ifb\_ist\_revolut\_3 ..... 7
- ifb\_ist\_revolut\_4 ..... 7
- ifb\_ist\_sourceid ..... 7
- ifb\_posits\_in ..... 7
- ifb\_posits\_out ..... 7
- ifb\_power\_off ..... 6
- ifb\_power\_on ..... 6
- ifb\_reset ..... 6
- ifb\_soll\_gaplong ..... 7
- ifb\_soll\_gapradi ..... 7

## —G—

Gerät

- Bedienung ..... 11

- Hardware ..... 5
- logisches ..... 8
- Repräsentation ..... 12
- Schnittstelle ..... 5

Gerätemodell ..... 5

- Kennzeichnung ..... 12
- Master-Properties ..... 12
- Slave-Properties ..... 20

## —H—

Hardware des Gerätes ..... 5

Hardwarestatus ..... 9

Hardwarewarning-Bit ..... 12

## —I—

ifb\_ist\_gaplong ..... 7

ifb\_ist\_gapradi ..... 7

ifb\_ist\_revolut\_1 ..... 7

ifb\_ist\_revolut\_2 ..... 7

ifb\_ist\_revolut\_3 ..... 7

ifb\_ist\_revolut\_4 ..... 7

ifb\_ist\_sourceid ..... 7

ifb\_posits\_in ..... 7

ifb\_posits\_out ..... 7

ifb\_power\_on, ifb\_power\_off ..... 6

ifb\_reset ..... 6

ifb\_soll\_gaplong ..... 7

ifb\_soll\_gapradi ..... 7

Init ..... 11

Interfacekarte ..... 6

Interlock ..... 8

Interrupt

- DRD Interrupt ..... 8
- DRQ Interrupt ..... 8
- Interlock ..... 8

## —K—

Kaltstarts ..... 11

Konfigurationsabfrage ..... 10

## —L—

logisches Gerät ..... 8

Lokale Datenbasis ..... 23

- Tabelle der Konstanten ..... 23

—M—	
Master-Properties .....	12
Medienversorgung .....	8, 11
—N—	
Normalbetrieb .....	11
—P—	
Properties	
• ACTIV .....	20
• CONSTANT .....	17
• COPYSET .....	21
• EQMERROR .....	20
• GAPLONGI .....	19
• GAPLONGS .....	19
• GAPRADII .....	18
• GAPRADIS .....	18
• INFOSTAT .....	16
• INIT .....	15
• Master- .....	12
• MOTPOSI .....	20
• MOTPOSS .....	19
• POSITI .....	19
• POSITS .....	19
• POWER .....	13
• RESET .....	15
• REVOLUTI .....	19
• Slave- .....	20
• SOURCEID .....	19
• STATUS .....	13
• VERSION .....	15
—Q—	
Quellenidentifikation .....	8, 11
—R—	
Repräsentation des Gerätes .....	12
Reset .....	11
—S—	
Schnittstelle zum Gerät .....	5
Slave-Properties .....	20
Softwareentwurf .....	23
Softwarestatus .....	13
Störungen .....	12
• Kommunikation EC – Gerät .....	12
Startwerte .....	11
Status	
• Cup .....	14
• Extraktionselektroden .....	14
• Hardware .....	9
• Medienversorgung .....	14
• Quellenidentifikation .....	14
• Turbopumpen .....	14
• Vorvakuumpumpen .....	14
Statusbits .....	9
—T—	
Terminalversorgung-Hardware .....	7
Timing .....	11
Turbopumpen .....	8, 11
—U—	
USRs .....	23
—V—	
Varianten	
• Betriebs- .....	11
Vorvakuumpumpen .....	8, 11
—W—	
Warmstarts .....	11