

DTC - Resonante Trafos zur Ladungsbestimmung

Gerätemodell und Softwareentwurf

**Andreas Peters
Rudi Steiner
Hannes Reeg
Peter Kainberger**

Dieses Papier enthält die Beschreibung des Gerätemodells „DTC - Resonante Trafos zur Ladungsbestimmung“ und den Entwurf der Gerätesoftware für dieses Gerät.

Für die Hochenergie-Strahltransportstrecken TE und TR sind vier Trafosysteme entwickelt worden, die jeweils einen schnellen Trafo zur Pulsbeobachtung (siehe Gerätemodell „DTP“) und einen resonanten Trafo zur Ladungsbestimmung beinhalten. Für letzteren soll hier dessen Funktionalität und die zugehörige Software auf VME-Kontrollsystemebene beschrieben werden.

Änderungsprotokoll			
Datum	GM-Version	Name	Kommentar
26. Februar 97	0.1	Andreas Peters	erste Annäherung an Gerät
04. Juli 97	0.2	Andreas Peters	erste Arbeitsversion (noch ohne Properties)
11. Aug. 97	0.3	Andreas Peters u.a.	erste Arbeitsversion mit Properties
10. Sep. 97	0.4	P. Kainberger	Änderung der Properties
17. Sep. 97	0.5	P. Kainberger	Änderung der ACTIV-Property
April 00	–	M. Kühn	Überarbeitete und erweiterte T _E X-Version, die sowohl in PostScript als auch in HTML konvertiert werden kann.
13. Feb. 2001	0.6	P. Kainberger	Anpassung an Hardwareänderungen.

Inhaltsverzeichnis

I	Das Gerätemodell	7
1	Die Aufgabe des Gerätes	7
2	Die Hardware des Gerätes	7
2.1	Gerätevarianten	7
3	Die Schnittstelle zum Gerät	7
3.1	Funktionscodes der Interfacekarte	7
3.2	Interlock Interrupt	8
3.3	Data Request (DRQ) Interrupts	9
3.4	Data Ready (DRD) Interrupts	9
3.5	Umfang eines logischen Gerätes	9
3.6	Definition der Bits des Hardwarestatus	9
3.7	Konfigurationsabfrage	9
4	Die Bedienung des Gerätes	9
4.1	Aufgaben im Normalbetrieb	10
4.1.1	Messung vorbereiten	10
4.1.2	Istwerte lesen	10
4.1.3	Status lesen	10
4.1.4	Transmissionsmessung	10
4.1.5	Meßbereichsautomatik	10
4.1.6	Einschalten	11
4.1.7	Ausschalten	11
4.2	Genauigkeitsanforderungen	11
4.3	Zeitkritische Anforderungen	11
4.4	Einordnung in das Timing	11
4.5	Festlegung von Startwerten	12
4.5.1	Kaltstarts	12
4.5.2	Warmstarts	12
4.6	Handbetrieb	12
4.7	Ableitung des Hardwarefehler-Bits aus dem Gerätestatus	12
4.8	Verhalten bei Störungen	12
4.8.1	Geräteinterlock	12
4.8.2	Event-Sequenzfehler	13
4.8.3	Event-Overrun	13
4.8.4	Ausfall der Kommunikation EC – Gerät	13
4.9	Bedienungsfehler vom Operating	13
5	Die Repräsentation des Gerätes	13
5.1	Kennzeichnung des Gerätemodells	13
5.2	Die Master-Properties	13
5.2.1	POWER	14
5.2.2	STATUS	14
5.2.3	INIT	14
5.2.4	RESET	14
5.2.5	VERSION	14
5.2.6	INFOSTAT	15
5.2.7	CONSTANT	16

5.3	Die Slave-Properties	16
5.3.1	ACTIV	17
5.3.2	EQMERROR	17
5.3.3	COPYSET	18
5.3.4	GAINMODS	18
5.3.5	GAINMODI	19
5.3.6	GAINRNGS	19
5.3.7	GAINRNGI	19
5.3.8	CHARGINF	19
5.3.9	SGLCHARG	20
5.3.10	DATCHECK	20
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_ConstantS	21
9.2.2	R_ChargInf	21
9.2.3	R_SglCharg	22
9.2.4	R_DatCheck	22
9.2.5	W_GainModS	22
9.2.6	R_GainModS	22
9.2.7	R_GainModI	22
9.2.8	W_GainRngS	22
9.2.9	R_GainRngS	22
9.2.10	R_GainRngI	22
9.3	Globale Routinen	22
10	EQMs - Equipment Module	22
10.1	Interne Zustände	22
10.1.1	Bedeutung der internen Zustände	22
10.1.2	Übergänge zwischen den Zuständen	23
10.1.3	Standard-Zustandsübergänge	23
10.2	Kommandogetriggerte EQMs	23

10.2.1	Dev_Init_EQM	23
10.2.2	Dev_Reset_EQM	23
10.2.3	Status_EQM	23
10.2.4	Active_EQM	23
10.2.5	Power_EQM	23
10.3	Eventkonnectierte EQMs	23
10.3.1	Prep_EQM	23
10.3.2	Mess_EQM	24
10.3.3	SglPrep_EQM	24
10.3.4	End_EQM	24
10.4	Periodisch konnectierte EQMs	24
10.4.1	Update_Config_EQM	24
10.5	An externe Interrupts konnectierte EQMs	24
10.5.1	DRD_EQM	24
10.5.2	DRQ_EQM	24
10.6	EQMs für die Diagnose vor Ort	25
10.6.1	Display_DPR_EQM	25
10.6.2	Display_DevErr_EQM	25
10.7	Sonstige EQMs	25
10.7.1	Startup_EQM	25
10.7.2	UserIni	25
10.8	Globale Routinen	25
10.8.1	Read_and_Update_Status	25
10.8.2	Do_Intr_Service_Prep	25
10.9	Gerätevarianten	25
Index		27

Abbildungsverzeichnis

1	Standardzyklus mit relevanten Events für <i>Resonante Trafos</i>	11
---	--	----

Teil I

Das Gerätemodell

1 Die Aufgabe des Gerätes

Für die Hochenergie-Strahltransportstrecken TE und TR sind vier Trafosysteme entwickelt worden, die jeweils einen schnellen Trafo zur Pulsbeobachtung und einen resonanten Trafo zur Ladungsbestimmung beinhalten. Die letzteren Trafos gestatten damit (über die zusätzliche Kenntnis der spezifischen Ionenladung aus dem INIT) eine Ermittlung der Teilchenzahl am Meßort bei schneller Extraktion. Weiterhin gestatten diese Trafos eine zerstörungsfreie Transmissionsbestimmung zwischen SIS und ESR und entsprechenden Teilstrecken.

2 Die Hardware des Gerätes

....

?

Für weitere Auskünfte zur Hardware stehen in punkto Interfacekarte,etc. Rudi Steiner (Tel. 2392) und Wolfgang Panschow (Tel. 2341/2361) zur Verfügung, weitere Fragen zur Technik des Trafosystems beantwortet gern Hannes Reeg (Tel. 2447/2312). Alle technischen Angaben in diesem Gerätemodell stammen von den o.g. Personen.

2.1 Gerätevarianten

Es gibt derzeit keine Varianten dieses Geräts.

3 Die Schnittstelle zum Gerät

Bei der verwendeten digitalen Schnittstelle zum Rechnersystem handelt es sich um eine Interfacekarte vom Typ xxx und der Karte RTRAF1 (FG 429 100), die den Resonanten Trafo mit den Versorgungsspannungen und den Steuersignalen beliefert.

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_gain1	20	Funktion	Verstärkung = 1 setzen
ifb_gain2	22	Funktion	Verstärkung = 10 setzen
ifb_gain3	24	Funktion	Verstärkung = 100 setzen
ifb_gain4	26	Funktion	Verstärkung = 0.1 setzen (Dämpfung)
ifb_measure	30	Funktion	Messung starten
ifb_reset	34	Funktion	Reset ausführen
ifb_ist	80	Lesen	Istwert lesen (16Bit)
ifb_rdstat	90	Lesen	Gerätstatus lesen (16 Bit)

ifb_gain1

Mit diesem Kommando wird der Verstärkungsbereich 1 gesetzt.

ifb_gain2

Mit diesem Kommando wird der Verstärkungsbereich 2 gesetzt.

ifb_gain3

Mit diesem Kommando wird der Verstärkungsbereich 3 gesetzt.

ifb_gain4

Mit diesem Kommando wird die Dämpfung aktiviert und alle Verstärker deaktiviert.

ifb_measure

Mit diesem Kommando wird das Gerät zur Messung *scharf* gemacht.

ifb_reset

Mit diesem Kommando wird der Verstärkungsbereich 1 gesetzt und ein vorangegangener Meßbefehl gelöscht. Das Gerät ist dann wieder zur Messung bereit.

ifb_ist

Lesen des 16-Bit Istwertes.

ifb_rdstat

Lesen des Gerätestatus. Die Bedeutung der einzelnen Bits ist in der folgenden Tabelle zusammengefaßt.

Bit	Name	Bedeutung	
		High (1)	Low (0)
0	+15 V ok	ok	off
1	-15 V ok	ok	off
2...3		0: Gain 1 1: Gain 2 2: Gain 3	
4	SW-Test	on	off
5	HW-Test	on	off
6	Dämpfung	on	off
7...11		reserved	
12...15		Versionskennung der EPLDs	

3.2 Interlock Interrupt

Ein Interlock-Interrupt wird von diesem Gerät nicht generiert.

3.3 Data Request (DRQ) Interrupts

Ein DRQ-Interrupt wird von diesem Gerät nicht generiert.

3.4 Data Ready (DRD) Interrupts

Ein DRD-Interrupt wird von diesem Gerät nicht generiert.

3.5 Umfang eines logischen Gerätes

Hier liegt der einfachste Fall vor: Eine Interfacekarte repräsentiert ein logisches Gerät.

3.6 Definition der Bits des Hardwarestatus

Das Gerät liefert mit dem Funktionscode `ifb_rdstat` 16 Bit Statusinformation (s. Abs. 3.1 auf Seite 8), von denen nur die *statischen* Informationen (Zustand der Spannungsversorgung) in den Gerätestatus übernommen 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
8	+15 V ok	ok	off
9	-15 V ok	ok	off
10 . . . 31		reserved	

3.7 Konfigurationsabfrage

Ein logisches Gerät ist ansprechbar und damit im Kontrollsystem vorhanden, wenn von der Interfacekarte mit dem Funktionscode `C0hex` ein Status gelesen werden kann.

4 Die Bedienung des Gerätes

Hier wird beschrieben, wie das Gerät (die Hardware) bedient werden muß (die „dynamische“ Schnittstelle). Das beinhaltet sowohl die Anforderungen *vom* Gerät als auch die Anforderungen *an das* Gerät. Zudem werden Anfangszustände und das Verhalten bei Störungen festgelegt.

4.1 Aufgaben im Normalbetrieb

Zur Messung eines schnell extrahierten Strahlpulses muß das Gerät für jeden Schuß zur Messung vorbereitet und nach erfolgter Extraktion die Istwerte gelesen werden. Dabei ist zu berücksichtigen, daß während eines Zyklus bis zu zwei Messungen durchgeführt werden müssen (Doppelschuß-Extraktion).

4.1.1 Messung vorbereiten

Zur Vorbereitung der Messung wird an das Gerät der eingestellte Meßbereich geschickt und mit dem Funktionscode `ifb_measure` die Messung gestartet (s. Abs. 3.1 auf Seite 7).

4.1.2 Istwerte lesen

Nach erfolgter Extraktion werden vom Gerät die Istwerte (ADC-Wert und Status) gelesen (s. Abs. 3.1 auf Seite 7). und ein Datenstatus generiert. Der Datenstatus ergibt sich aus dem 16-Bit ADC-Wert und hat folgende Belegung:

Bit	Bedeutung
0	Summenbit (immer dann „0“, wenn eines der Bits 1 ... 3 „0“ zeigt, sonst „1“)
1	ADC Overage: immer dann, wenn MSB true
2	ADC Underrange: immer dann, wenn die 4 MSB false sind
3	ADC Overload: immer dann, wenn die 4 MSB true sind
4	nicht belegt
⋮	⋮
15	nicht belegt

4.1.3 Status lesen

Wie schon in Kapitel 3.1 unter dem Abschnitt `ifb_rdstat` erläutert, werden als Gerätestatus nur die Anzeigen der Spannungsversorgung ausgewertet.

4.1.4 Transmissionsmessung

Die Online-Transmissions-Überwachung verlangt, daß zur Transmissionsmessung die Istwerte eines Zyklus, der dem Event `EVT_Gap_Tra_Mess (97)` folgt, in einem separaten Speicher abzulegen sind.

4.1.5 Meßbereichsautomatik

Zur automatischen Einstellung des optimalen Meßbereichs ist auch für die *Resonanten Trafos* eine Meßbereichsautomatik vorzusehen, die mit Hilfe des Datenstatus (s. 4.1.2) den optimalen Meßbereich zu finden versucht. Das bisher ungelöste Problem dabei ist, daß bei einer *Doppelschuß-Extraktion* für beide Messungen der gleiche Meßbereich gelten soll. Beide Meßwerte können aber sehr unterschiedlich sein, weshalb die Meßbereichsautomatik erkennen muß, ob beide Meßwerte, oder nur einer davon zur Auswertung herangezogen werden können.

Das Verfahren hierzu ist bisher noch nicht ausreichend spezifiziert. Die Properties sollen aber (aus Kompatibilitätsgründen) trotzdem implementiert werden. Das eigentliche Verfahren kann dann später nachgerüstet werden.

?

4.1.6 Einschalten

... geht nur per Hand.

4.1.7 Ausschalten

... geht auch nur per Hand.

4.2 Genauigkeitsanforderungen

Gibt's dazu brauchbare Informationen?

?

4.3 Zeitkritische Anforderungen

Bei der Handhabung des Gerätes sind keine zeitkritischen Anforderungen zu berücksichtigen.

4.4 Einordnung in das Timing

Das Gerät nimmt an der Puls-zu-Puls-Modulation teil.

Ein Beispiel eines SIS Standardzyklus mit relevanten Events für *Resonante Trafos* ist in Abbildung 1 angegeben.

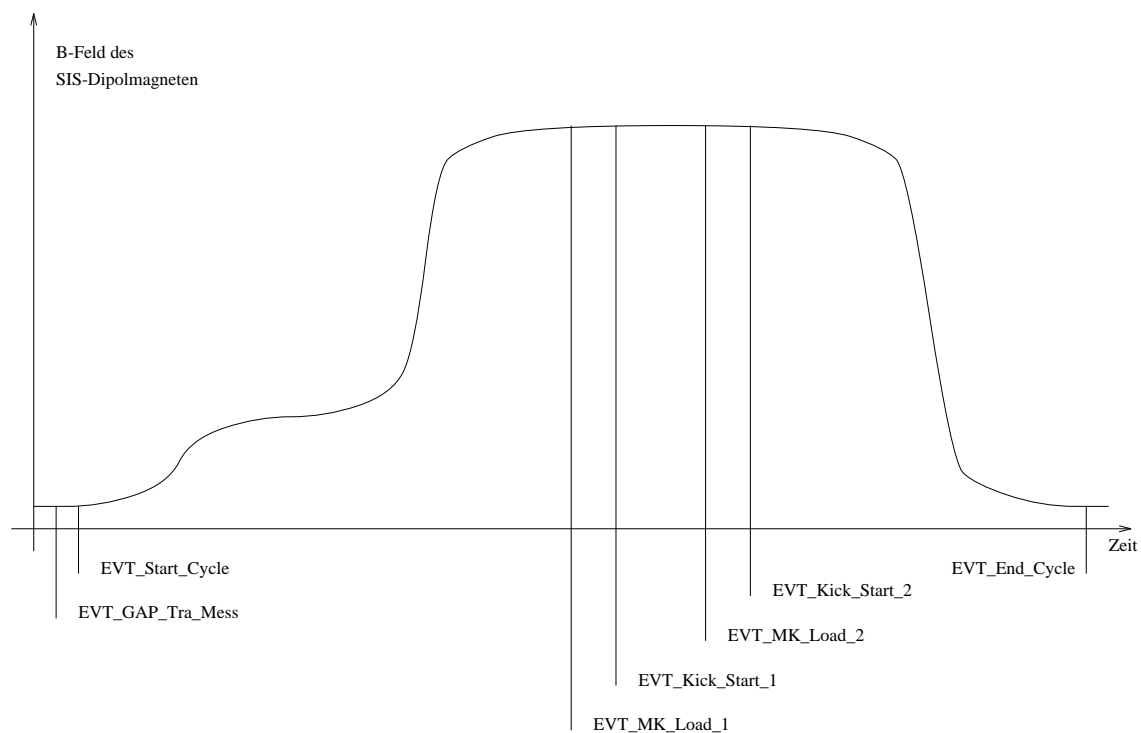


Abbildung 1: Standardzyklus mit relevanten Events für *Resonante Trafos*

Die Konnektierungen sind in Tabelle 3 zusammengefasst.

Aktion	Event
Messung vorbereiten	Evt_MK_Load_1 (47)
Messung vorbereiten	Evt_MK_Load_2 (48)
Meßwerte lesen, Datenstatus generieren	Evt_Kick_Start_1 (49)
Meßwerte lesen, Datenstatus generieren	Evt_Kick_Start_2 (69)
Meßdaten als gültig kennzeichnen	Evt_End_Cycle (55)
Flag für Transmissionsmessung setzen	Evt_Gap_Tra_Mess (97)

Tabelle 3: Standard-Eventkonn timerungen für *Resonante Trafos*

4.5 Festlegung von Startwerten

4.5.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt:

- Mit dem Funktionscode `ifb_reset` wird ein Geräteset durchgeführt.
- Alle Sollwerte werden für alle virtuellen Beschleuniger folgendermaßen initialisiert:

Bedeutung	Wert bei Kaltstart
Meßbereichsmodus	manuell
Meßbereich	1 (unempfindlich)

- Alle Istwerte und Timestamps werden mit 0 initialisiert.
- Das Gerät wird für alle virtuellen Beschleuniger auf „aktiv“ gesetzt
- Bei einem Kaltstart der SE wird diese in den Eventmode-Betrieb geschaltet.
- Bei einem Kaltstart der SE werden die Standard-Eventkonn timerungen gesetzt (siehe Tabelle 3 auf Seite 12).

4.5.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Mit dem Funktionscode `ifb_reset` wird ein Geräteset durchgeführt.
- Alle Istwerte und Timestamps werden mit 0 initialisiert.

4.6 Handbetrieb

Ein Handbetrieb des Gerätes ist nicht möglich.

4.7 Ableitung des Hardwarefehler-Bits aus dem Gerätestatus

Bei diesem Gerät wird kein Hardwarefehler aus den Bits des Gerätestatus abgeleitet.

4.8 Verhalten bei Störungen

4.8.1 Geräteinterlock

Das Gerät kann keinen Geräteinterlock erzeugen.

4.8.2 Event-Sequenzfehler

Die normale Behandlung bei einem Sequenz-Fehler: Zyklus abbrechen.

4.8.3 Event-Overrun

Die normale Behandlung bei einem Overrun-Fehler: Zyklus abbrechen.

4.8.4 Ausfall der Kommunikation EC – Gerät

Ein Ausfall der Kommunikation zwischen EC und Gerät führt zu *Timeouts* bei der Istwert-Auslesung. Das betroffene Gerät wird innerhalb der turnusmäßigen Konfigurationsabfrage durch den ECM als *offline* erkannt und nicht mehr bedient.

4.9 Bedienungsfehler vom Operating

Nichtdefinierte Sollwerte von der Operating-Ebene müssen explizit abgefangen werden und führen zu einer Fehlermeldung. Alle gerätespezifischen Fehlermeldungen sollen in deutscher Sprache abgefaßt sein, die Ersteller dieses Papiers sind bei der Formulierung der Texte gern behilflich.

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 3.5 definierten logischen Gerätes.

5.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **DTC**.

Die Gerätemodellnummer ist 41_{dez} .

5.2 Die Master-Properties

Master-Properties							
Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
POWER	R/W	0	–	1	BitSet16	1	0
STATUS	R	0	–	1	BitSet32	1	0
INIT	N	0	–	0	–	–	–
RESET	N	0	–	0	–	–	–
VERSION	RA	0	–	48	BitSet8	1	0
INFOSTAT	RA	0	–	25	BitSet32	1	0
CONSTANT	RA	0	–	10	RealF	1	0

5.2.1 POWER

Bedeutung: Gibt an, ob der Leistungsteil des Gerätes ein- oder ausgeschaltet ist bzw. werden soll. Da dieses Gerät keinen Netzschalter besitzt, kann es auch nicht geschaltet werden. Jeder Schaltversuch führt zu einer Fehlermeldung. Zur Ermittlung des POWER-Zustandes werden die Statusanzeigen der Spannungsversorgung ausgewertet (s. Abs. 3.6 auf Seite 9).

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. Die Bits entsprechen den Statusbits, wie sie im Abschnitt 3.6 auf Seite 9 und im Abschnitt 3.6 auf Seite 9 erklärt sind.

5.2.3 INIT

Bedeutung: Initialisierung des Gerätes (Kaltstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.5.1 auf Seite 12.

Parameter: Keine.

Daten: Keine.

5.2.4 RESET

Bedeutung: Reset des Gerätes (Warmstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.5.2 auf Seite 12.

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 Meßbereichsinformationen (Meßbereichsauflösung u Endwert). Diese gerätespezifischen Konstanten werden dem Operating als Tabelle zur Verfügung gestellt.

Parameter: keine

Daten: 10 RealF Werte

- 1 Anzahl der Meßbereiche
- 2 Einheit der Meßbereiche (hier: [C, As]; entsprechende Kodierung im Kontrollsystem: 19)
- 3 Meßbereichsendwert für Meßbereich 1
- 4 Meßbereichsauflösung für Meßbereich 1
- 5 Meßbereichsendwert für Meßbereich 2
- 6 Meßbereichsauflösung für Meßbereich 2
- 7 Meßbereichsendwert für Meßbereich 3
- 8 Meßbereichsauflösung für Meßbereich 3
- 9 Meßbereichsendwert für Meßbereich 4
- 10 Meßbereichsauflösung für Meßbereich 4

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

Slave-Properties							
Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
COPYSET	W	0	–	1	BitSet16	1	0
EQMERROR	RA	0	–	137	Integer32	1	0
GAINMODS	R/W	0	–	1	BitSet16	1	0
GAINMODI	R	0	–	1	BitSet16	1	0
GAINRNGS	R/W	0	–	1	BitSet16	1	0
GAINRNGI	R	0	–	1	BitSet16	1	0
CHARGINF	RA	0	–	13	RealF	1	0
SGLCHARG	RA	0	–	13	RealF	1	0
DATCHECK	RA	0	–	13	RealF	1	0

Die meisten Slave-Properties tauchen zweimal auf, einerseits mit der Endung **I** für Istwert (nur Lesen), andererseits mit der Endung **S** für Sollwerte (Lesen und Schreiben). Auch für die Sollwerte, die nicht direkt vom Gerät zurückgelesen werden können, macht das Sinn, wenn die (neuen) Sollwerte erst dann auf der SE-Ebene in die Istwerte kopiert werden, wenn die Solldaten ohne Fehler in die Interfacekarte geschrieben werden konnten. Der Wert dieser so spezifizierten „Istwerte“ ist bei Hardwaretests und im Servicefall, aber auch im Operating nicht zu unterschätzen, denn es läßt sich schnell und einfach feststellen, ob ein (neuer) Sollwert auch auf der VME-, bzw. Geräteebene angekommen ist.

5.3.1 ACTIV

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

Zur Vereinfachung der Handhabung der verschiedenen Meßaufgaben des Gerätes wurde vereinbart, daß das Gerät immer *aktiv* geschaltet sein soll. Der Versuch, das Gerät *inaktiv* zu setzen führt zu einer Fehlermeldung.

Parameter: Keine.

Daten: Das Datum kann nur den Wert Eins annehmen. 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: Keine.

Daten: Die Anzahl der Fehlermeldungen sei bezeichnet durch:

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

Weiterhin soll gelten:

$$l = m + s$$

$$t = m + s + b$$

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.

Parameter: Keine.

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

5.3.4 GAINMODS

Bedeutung: Dient zum Ein- bzw. Ausschalten der Meßbereichsautomatik. Bei eingeschalteter Meßbereichsautomatik paßt sich der Meßbereich der jeweiligen Signalstärke an. Siehe hierzu Abschnitt 4.1.5 auf Seite 10.

Parameter: Keine

Daten: 1 BitSet16 Wert mit folgender Bedeutung:

Data	Erläuterung
1	Manuell
2	Automatik
3	Halbautomatisch

1. **Manuell** : In diesem Modus werden die Meßbereiche manuell eingegeben. Eine automatische Angleichung an den Meßbereich erfolgt nicht.

2. **Automatik** : Die Automatik steuert selbständig die Meßbereiche nach, sobald sich die Signalstärke ändert.
3. **Halbautomatisch** : Bei der Halbautomatik sucht die Meßbereichsautomatik selbständig nach dem passenden Meßbereich und stellt dann automatisch auf manuell zurück.

Erläuterung: Konform zu den meisten Strahldiagnosegeräten ist der unempfindlichste Meßbereich jener mit der Nummer „1“. Mit steigender Meßbereichsnummer wird die Messung empfindlicher. Das heißt für den Automatikmodus, bei zu schwachem Signal muß die Meßbereichsnummer um 1 erhöht werden, bei zu starkem Signal muß die Meßbereichsnummer um 1 erniedrigt werden.

5.3.5 GAINMODI

Bedeutung: Liest den mit GAINMODS eingestellten Modus der Meßbereichsautomatik zurück.

Parameter: Keine

Daten: Siehe GAINMODS (5.3.4)

5.3.6 GAINRNGS

Bedeutung: Dient zum manuellen Setzen des Meßbereichs.

Parameter: Keine

Daten: 1 BitSet16 Wert, der die Meßbereichsnummer (1–3) angibt. Weitere Angaben findet man unter CONSTANT (5.2.7).

Erläuterung: Konform zu den meisten Strahldiagnosegeräten ist der unempfindlichste Meßbereich jener mit der Nummer „1“. Mit steigender Meßbereichsnummer wird die Messung empfindlicher.

5.3.7 GAINRNGI

Bedeutung: Liest den aktuellen Meßbereich des Gerätes.

Parameter: Keine

Daten: Siehe GAINRNGS (5.3.6)

5.3.8 CHARGINF

Bedeutung: Mit dem Kommando CHARGINF wird ein 13 Werte großes Datenpaket gelesen, das alle Informationen zu den Messungen eines resonanten Trafos in einem virtuellen Beschleuniger enthält.

Parameter: Keine

Daten: 13 RealF Werte

1. Kennzeichnung, ob erster Meßwert vorhanden ist (=1) oder nicht (=0)
2. Erster Meßwert in Coulomb (bei schneller Extraktion: 1. Schuß; bei Durchschuß-Betrieb der einzige Meßwert)
3. Datenstatus des ersten Meßwertes (siehe 4.1.2)
4. Kennzeichnung, ob zweiter Meßwert vorhanden ist (=1) oder nicht (=0)

?

5. Zweiter Meßwert in Coulomb (bei schneller Extraktion: 2. Schuß, falls vorhanden)
6. Datenstatus des zweiten Meßwertes (siehe 4.1.2)
7. Meßbereichsendwert in Coulomb
8. Meßbereichsauflösung (0.01 entspr. 1%)
9. Meßbereichsnummer (Istwert) siehe 5.3.7
10. Meßbereichsnummer (Sollwert) siehe 5.3.6
11. Meßbereichsmodus (Istwert) siehe 5.3.5
12. Meßbereichsmodus (Sollwert) siehe 5.3.4
13. Aktivstatus siehe 5.3.1

5.3.9 SGLCHARG

Bedeutung: Lesen der Meßwerte, die für die Transmissionsmessung separat gespeichert wurden.

Parameter: keine

Daten: 13 RealF Werte (siehe Property CHARGINF auf Seite 19)

5.3.10 DATCHECK

Bedeutung: Lesen der Meßwerte, die für die Transmissionsmessung separat gespeichert wurden. Diese Property ist speziell zur Konnektierung durch ein Operatingprogramm gemacht, welches jedes Meßwertpaket nur genau einmal lesen will. Deshalb wird bei dieser Property das Meßwertpaket nur dann geschickt, wenn die interne Verwaltung auf der VME-Ebene sagt, daß dieser Wert noch nie mit *DATCHECK* gelesen wurde.

Parameter: keine

Daten: 13 RealF Werte (siehe Property CHARGINF auf Seite 19)

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 *Resonanten Trafos* folgende gerätespezifischen USRs benötigt:

9.2.1 R_ConstantS

Liefert die gerätespezifischen Konstanten.

9.2.2 R_ChargInf

Liefert alle relevanten Daten aus einem Zyklus.

9.2.3 R_SglCharg

Liefert alle relevanten Daten aus einem Single-Shot-Zyklus.

9.2.4 R_DatCheck

Liefert genauso wie R_SglCharg alle relevanten Daten aus einem Single-Shot-Zyklus. Zusätzlich wird aber verwaltet, daß jedes Meßwertpaket nur ein einziges mal verschickt wird. Damit ist eine konnektierte Ausführung möglich, die nur dann Daten liefert, wenn neue Meßergebnisse vorliegen. Dabei darf aber nur genau eine Konnektierung pro Beschleuniger vorliegen!

9.2.5 W_GainModS

Setzen des Meßbereichsmodus.

9.2.6 R_GainModS

Lesen des Meßbereichsmodus (Sollwert).

9.2.7 R_GainModI

Lesen des Meßbereichsmodus (Istwert).

9.2.8 W_GainRngS

Setzen des Meßbereichs.

9.2.9 R_GainRngS

Lesen des Meßbereichs (Sollwert).

9.2.10 R_GainRngI

Lesen des Meßbereichs (Istwert).

9.3 Globale Routinen

Bisher keine.

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.
error	Während der Abarbeitung eines EQMs wurde ein Fehler erkannt.
ready	Das Gerät ist bereit für Aktionen. Ausgangszustand am Beginn eines virtuellen Beschleunigers.
busy	Das Gerät hat eine Messung gestartet.

10.1.2 Übergänge zwischen den Zuständen

Die Zustände und die Übergänge zwischen denselben sind in Tabelle 7 zusammengefaßt. Die Legende zu diesen Tabellen ist in Tabelle 10.1.2 zu finden.

Tabelle der Zustandsübergänge				
von↓ nach→		error	ready	busy
error	U:	–	RESET, Zyklusende	Vorbereitungs- event
	A:	–	Reset_EQM, End_EQM	Prep_EQM
ready	U:	Sequenzfehler etc.	–	Vorbereitungs- event
	A:	div. EQMs	–	Prep_EQM
busy	U:	Sequenzfehler etc.	Messevent	–
	A:	Mess_EQM	Mess_EQM	–

Tabelle 7: Zustandsübergangsdiagramm

Legende

- Die Priorität der Zustände (höchste Priorität zuerst): error, ready, busy.
- U: Auslösende Ursache.
 RESET Reset wird per Kommando oder Knöpfchendrücken ausgelöst.
 Sequenzfehler Vorbereitungs- und Mess-Events kommen in der falschen Reihenfolge.
- A: Ausführende Stelle des Zustandübergangs.
 ..._EQM Innerhalb des EQMs ..._EQM.

Tabelle 8: Legende zu den Zustandsübergangsdiagrammen

10.1.3 Standard-Zustandsübergänge

ready -> busy -> ready

10.2 Kommandogetriggerte EQMs

10.2.1 Dev_Init_EQM

10.2.2 Dev_Reset_EQM

10.2.3 Status_EQM

10.2.4 Active_EQM

Abweichend von der Standardbehandlung kann dieses Gerät nicht *inaktiv* geschaltet werden.

10.2.5 Power_EQM

10.3 Eventkonnektierte EQMs

10.3.1 Prep_EQM

Event: Evt_MK_Load_1 (47), Evt_MK_Load_2 (48).

Aktion: Die Messung im aktuellen Beschleuniger wird vorbereitet:

- Status lesen.

- Meßbereich setzen.
- Messung *enablen*.

10.3.2 Mess_EQM

Event: Evt_Kick_Start_1 (49), Evt_Kick_Start_2 (69).

Aktion: Lesen der Meßergebnisse und Generierung des Datenstatus.

- Meßwert lesen.
- Status lesen.
- Datenstatus generieren.

10.3.3 Sgl_Prep_EQM

Event: Evt_Gap_Tra_Mess (97)

Aktion: Setzen eines Flags, welches den nächsten Zyklus als *Single-Shot-Zyklus* markiert.

10.3.4 End_EQM

Event: Evt_End_Cycle (55)

Aktion: Aufräumarbeiten am Ende des Zyklus.

- Sync-Buffer umschalten und damit Daten zum Lesen freigeben.
- Falls das Single-Shot-Flag gesetzt ist, werden die Meßwerte in den Single-Shot-Buffer kopiert.

10.4 Periodisch konnektierte EQMs

10.4.1 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.

10.5 An externe Interrupts konnektierte EQMs

10.5.1 DRD_EQM

Interrupt: Data Ready Interrupt.

Aktion: Keine gerätespezifischen Aktionen.

10.5.2 DRQ_EQM

Interrupt: Data Request Interrupt.

Aktion: Keine gerätespezifischen Aktionen.

10.6 EQMs für die Diagnose vor Ort

10.6.1 Display_DPR_EQM

Parameter: Das EQM benötigt 2 Parameter.

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

Daten: Keine.

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

10.6.2 Display_DevErr_EQM

Parameter: Das EQM benötigt 2 Parameter.

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

Daten: Keine.

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

10.7 Sonstige EQMs

10.7.1 Startup_EQM

Installiert die Event-EQM-Konnektierung für alle virtuellen Beschleuniger (siehe hierzu auch Abschnitt 4.4 auf Seite 11) und schaltet die SE in den Event-Mode.

10.7.2 UserIni

- Setzen der Dualport-RAM-Konstanten,
- setzen der Dualport-RAM-Pointer,
- setzen des IFB_Group_Count,
- setzen der Geräte- und Versions-ID
- setzen des Modus für die Summeninterlock-Behandlung,
- aufsetzen des periodischen Auftrags zum Konfigurationscheck,
- anmelden der gerätespezifischen EQMs.

10.8 Globale Routinen

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

10.8.1 Read_and_Update_Status

10.8.2 Do_Intr_Service_Prep

10.9 Gerätevarianten

Es gibt keine Varianten.

Index

—Symbole —

? 7, 10, 11, 20
Änderungsprotokoll 2

—A—

Abriß 2
Active_EQM 23
An externe Interrupts konnektierte EQMs 24
Aufgabe des Gerätes 7
Ausschalten 11

—B—

Bedienung des Gerätes 9
Bedienungsfehler 13

—D—

Datenbasis 21
Dev_Init_EQM 23
Dev_Reset_EQM 23
Display_DevErr_EQM 25
Display_DPR_EQM 25
DRD Interrupt 9
DRD_EQM 24
DRQ Interrupt 9
DRQ_EQM 24
Dualport RAM 21

—E—

Einschalten 11
End_EQM 24
EQMs 22

- An externe Interrupts konnektierte 24
 - DRD_EQM 24
 - DRQ_EQM 24
- Eventkonnektierte 23
 - End_EQM 24
 - Mess_EQM 24
 - Prep_EQM 23
 - Sgl_Prep_EQM 24
- für die Diagnose vor Ort 25
 - Display_DevErr_EQM 25
 - Display_DPR_EQM 25
- Globale Routinen 25
 - Do_Intr_Service_Prep 25

- Read_and_Update_Status 25
- Kommandogetriggerte 23
 - Active_EQM 23
 - Dev_Init_EQM 23
 - Dev_Reset_EQM 23
 - Power_EQM 23
 - Status_EQM 23
- Periodisch konnektierte 24
 - Update_Config_EQM 24
- Sonstige - 25
 - Startup_EQM 25
 - UserIni 25

Event-Overrun 13
Event-Sequenzfehler 13
Eventkonnektierte EQMs 23
Eventkonnektierungen 11

—F—

Funktionscodes 7

- ifb_gain1 8
- ifb_gain2 8
- ifb_gain3 8
- ifb_gain4 8
- ifb_list 8
- ifb_measure 8
- ifb_rdstat 8
- ifb_reset 8

—G—

Genauigkeitsanforderungen 11
Gerät

- Aufgabe 7
- Bedienung 9
- Hardware 7
- logisches 9
- Repräsentation 13
- Schnittstelle 7

Gerätemodell 7

- Kennzeichnung 13
- Master-Properties 13
- Slave-Properties 16

Gerätevarianten 7, 25
Globale Routinen 22, 25

—H—

Handbetrieb 12

Hardware des Gerätes	7
Hardwarefehler-Bit	12
Hardwarestatus	9

—I—

ifb_gain1	8
ifb_gain2	8
ifb_gain3	8
ifb_gain4	8
ifb_ist	8
ifb_measure	8
ifb_rdstat	8
ifb_reset	8
Init	12
Interfacekarte	7
Interlock	8, 12
Interne Zustände	22
Interrupt	
• DRD Interrupt	9
• DRQ Interrupt	9
• Interlock	8
Istwerte lesen	10

—K—

Kaltstarts	12
Kommandogetriggerte EQMs	23
Konfigurationsabfrage	9

—L—

logisches Gerät	9
Lokale Datenbasis	21

—M—

Master-Properties	13
Meßbereichsautomatik	10
Mess_EQM	24
Messung vorbereiten	10

—N—

N_Init	21
N_Reset	21
Normalbetrieb	10

—O—

Overrun	13
---------------	----

—P—

Periodisch konnektierte EQMs	24
Power_EQM	23
Prep_EQM	23

Properties

• ACTIV	17
• CHARGINF	19
• CONSTANT	16
• COPYSET	18
• EQMERROR	17
• GAINMODI	19
• GAINMODS	18
• GAINRNGI	19
• GAINRNGS	19
• INFOSTAT	15
• INIT	14
• Master-	13
• POWER	14
• RESET	14
• Slave-	16
• STATUS	14
• VERSION	14

Propertis

• DATCHECK	20
• SGLCHARG	20

—R—

R_Active	21
R_ChargInf	21
R_ConstantS	21
R_DatCheck	22
R_EQMErr	21
R_GainModI	22
R_GainModS	22
R_GainRngI	22
R_GainRngS	22
R_InfoStat	21
R_Power	21
R_SglCharg	22
R_Status	21
R_Version	21
Repräsentation des Gerätes	13
Reset	12

—S—

Schnittstelle zum Gerät	7
Sequenzfehler	13
Sgl_Prep_EQM	24
Slave-Properties	16

Softwareentwurf	21
Softwarestatus	9
Sonstige EQMs	25
Störungen	12
• Event-Overrun	13
• Event-Sequenzfehler	13
• Interlock	12
• Kommunikation EC – Gerät	13
Standardzyklus SIS	11
Startup_EQM	25
Startwerte	12
Status lesen	10
Status_EQM	23
Statusbits	9

—T—

Timing	11
Transmissionsmessung	10

—U—

Update_Config_EQM	24
USRs	21
• gerätespezifische	21
– R_ChargInf	21
– R_ConstantS	21
– R_DatCheck	22
– R_GainModI	22
– R_GainModS	22
– R_GainRngI	22
– R_GainRngS	22
– R_SglCharg	22
– W_GainModS	22
– W_GainRngS	22
• Globale Routinen	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

• Geräte-	7, 25
-----------------	-------

—W—

W_Active	21
W_CopySet	21
W_GainModS	22
W_GainRngS	22
W_Power	21
Warmstarts	12

—Z—

Zeitkritische Anforderungen	11
Zustände	
• Interne	22
– Übergänge	23
– Bedeutung	22
– Standard-Übergänge	23