

# DTI - Pulsstrommesseinrichtung (Trafos)

Gerätemodell und Softwareentwurf

**P. Kainberger**

*Dieses Papier enthält die Beschreibung des Gerätemodells „DTI - Pulsstrommeßeinrichtung (Trafos)“ und den Entwurf der Gerätesoftware für dieses Gerät, soweit dies beim augenblicklichen Stand der Entwicklung, der laufenden Diskussionen und der fehlenden Festlegungen bzgl. des Projektes „Hochstrom am UNILAC“ möglich ist.*

<b>Änderungsprotokoll</b>			
Datum	GM-Version	Name	Kommentar
22.Jan. 97	DTI_01	P.Kainberger	Übernahme von DCI und Überarbeitung
25.Jan. 97	DTI_01	P.Kainberger	Aktualisierung
03.Nov. 97	DTI_01	P.Kainberger	Korrektur der Datenstatusbelegung
27.Jun. 98	DTI_02	P.Kainberger	Erweiterung um MMI und Aktualisierung
08.März 99	DTI_02	P.Kainberger	Aktualisierung, Umstellung auf Hochstromtiming
März 2000	–	M. Kühn	Überarbeitete und erweiterte T <sub>E</sub> X-Version, die sowohl in PostScript als auch in HTML konvertiert werden kann.
14.Feb. 01	DTI_02	P.Kainberger	Initwerte aus DBS

# Inhaltsverzeichnis

<b>I</b>	<b>Das Gerätemodell</b>	<b>7</b>
<b>1</b>	<b>Die Aufgabe des Gerätes</b>	<b>7</b>
<b>2</b>	<b>Die Hardware des Gerätes</b>	<b>7</b>
2.1	Quantisierender Stromfrequenzwandler . . . . .	7
<b>3</b>	<b>Die Schnittstelle zum Gerät</b>	<b>9</b>
3.1	Beschreibung der Geräte-Komponenten . . . . .	9
3.2	Funktionscodes der Interfacekarte . . . . .	11
3.3	Umfang eines logischen Gerätes . . . . .	11
3.4	Definition der Bits des Hardwarestatus . . . . .	12
3.5	Konfigurationsabfrage . . . . .	12
<b>4</b>	<b>Die Bedienung des Gerätes</b>	<b>12</b>
4.1	Aufgaben im Normalbetrieb . . . . .	12
4.1.1	Multi-Multiturn-Injektion . . . . .	13
4.2	Strahlüberwachung . . . . .	13
4.2.1	Referenzwertüberwachung . . . . .	13
4.2.2	Sequenzüberwachung . . . . .	13
4.2.3	Stromverlustüberwachung . . . . .	14
4.3	Genauigkeitsanforderungen . . . . .	14
4.4	Zeitkritische Anforderungen . . . . .	14
4.5	Einordnung in das Timing . . . . .	14
4.5.1	Unilac . . . . .	14
4.5.2	SIS . . . . .	15
4.6	Festlegung von Startwerten . . . . .	17
4.6.1	Kaltstarts . . . . .	17
4.6.2	Warmstarts . . . . .	17
4.7	Handbetrieb . . . . .	17
4.8	Ableitung des Hardwarefehler-Bits aus dem Gerätestatus . . . . .	17
4.9	Verhalten bei Störungen . . . . .	17
4.9.1	Geräteinterlock . . . . .	17
4.9.2	Event-Sequenzfehler . . . . .	18
4.9.3	Event-Overrun . . . . .	18
4.9.4	Ausfall der Kommunikation EC – Gerät . . . . .	18
4.10	Bedienungsfehler vom Operating . . . . .	18
4.11	Single-Shot-Messung . . . . .	18
<b>5</b>	<b>Die Repräsentation des Gerätes</b>	<b>19</b>
5.1	Kennzeichnung des Gerätemodells . . . . .	19
5.2	Die Master-Properties . . . . .	19
5.2.1	POWER . . . . .	19
5.2.2	STATUS . . . . .	19
5.2.3	INIT . . . . .	19
5.2.4	RESET . . . . .	20
5.2.5	VERSION . . . . .	20
5.2.6	INFOSTAT . . . . .	20
5.2.7	CONSTANT . . . . .	21
5.2.8	CURRAVG . . . . .	22

5.3	Die Slave-Properties	22
5.3.1	ACTIV	22
5.3.2	COPYSET	23
5.3.3	EQMERROR	23
5.3.4	GATESEL	24
5.3.5	GAINMODS	24
5.3.6	GAINMODI	24
5.3.7	GAINRNGS	25
5.3.8	GAINRNGI	25
5.3.9	CURRINFO	25
5.3.10	SGLRESET	25
5.3.11	SGLCURR	25
5.3.12	CTRLVAL	26
5.3.13	CTRLMODE	26
5.3.14	AVGCNTS	26
5.3.15	AVGCNTI	26
5.3.16	MMIINFO	27
 <b>II Der Entwurf der Software</b>		<b>29</b>
<b>6</b>	<b>Softwareentwurf</b>	<b>29</b>
<b>7</b>	<b>Lokale Datenbasis</b>	<b>29</b>
<b>8</b>	<b>Dualport RAM</b>	<b>29</b>
<b>9</b>	<b>USRs - User Service Routinen</b>	<b>29</b>
9.1	Obligatorische USRs	29
9.1.1	N_Init	29
9.1.2	N_Reset	29
9.1.3	R_Status	29
9.1.4	R_Power	29
9.1.5	W_Power	29
9.1.6	R_Active	29
9.1.7	W_Active	29
9.1.8	W_CopySet	29
9.1.9	R_EQMErr	29
9.1.10	R_Version	29
9.1.11	R_InfoStat	29
9.2	Gerätespezifische USRs	29
9.2.1	R_ConstantS	29
9.2.2	R_Currinfo	29
9.2.3	R_SglCurr	30
9.2.4	N_SglReset	30
9.2.5	W_GateselS	30
9.2.6	R_GateselS	30
9.2.7	W_GainModS	30
9.2.8	R_GainModS	30
9.2.9	R_GainModI	30
9.2.10	W_GainRngS	30
9.2.11	R_GainRngS	30
9.2.12	R_GainRngI	30

9.2.13	W_CtrlMode	30
9.2.14	R_CtrlMode	30
9.2.15	W_CtrlVal	30
9.2.16	R_CtrlVal	30
9.2.17	W_AvgCntS	31
9.2.18	R_AvgCntS	31
9.2.19	R_AvgCntI	31
9.2.20	R_MMInfo	31
9.2.21	R_CurrAvg	31
<b>10</b>	<b>EQMs - Equipment Module</b>	<b>31</b>
10.1	Interne Zustände	31
10.1.1	Bedeutung der internen Zustände	31
10.1.2	Übergänge zwischen den Zuständen	31
10.1.3	Standard-Zustandsübergänge	31
10.2	Eventkonnectierte EQMs	31
10.2.1	PrepUni_EQM	31
10.2.2	MessUni_EQM	32
10.2.3	PrepMMI_EQM	32
10.2.4	PrepSIS_EQM	32
10.2.5	MessSIS_EQM	32
10.3	Periodisch konnectierte EQMs	33
10.3.1	Update_Config_EQM	33
10.4	Kommandogetriggerte EQMs	33
10.4.1	Dev_Init_EQM	33
10.4.2	Dev_Reset_EQM	33
10.4.3	Status_EQM	33
10.4.4	Active_EQM	33
10.4.5	Power_EQM	33
10.5	EQMs für die Diagnose vor Ort	33
10.5.1	Display_DPR_EQM	33
10.5.2	Display_DevErr_EQM	33
10.6	Sonstige EQMs	33
10.6.1	Startup_EQM	33
10.6.2	CalcUni_EQM	34
10.7	Globale Routinen	34
10.7.1	Read_and_Update_Status	34
<b>11</b>	<b>Varianten</b>	<b>34</b>
<b>12</b>	<b>Besonderheiten</b>	<b>34</b>
	<b>Index</b>	<b>35</b>

## Abbildungsverzeichnis

1	Standardzyklus (Unilac) mit relevanten Events für Strommessung	15
2	Standardzyklus (SIS) mit relevanten Events für Strommessung	16





	$K$	Proportionalitätskonstante
	$t$	Rahmenpulsdauer
	$f_{takt}$	Taktfrequenz

$$\text{Meßwertzählung} = K \cdot I_{mess} \cdot t \cdot f_{takt}$$

Der Zeitzähler erfaßt während der Dauer des Rahmenpulses die gezählten Impulse des Taktgebers:

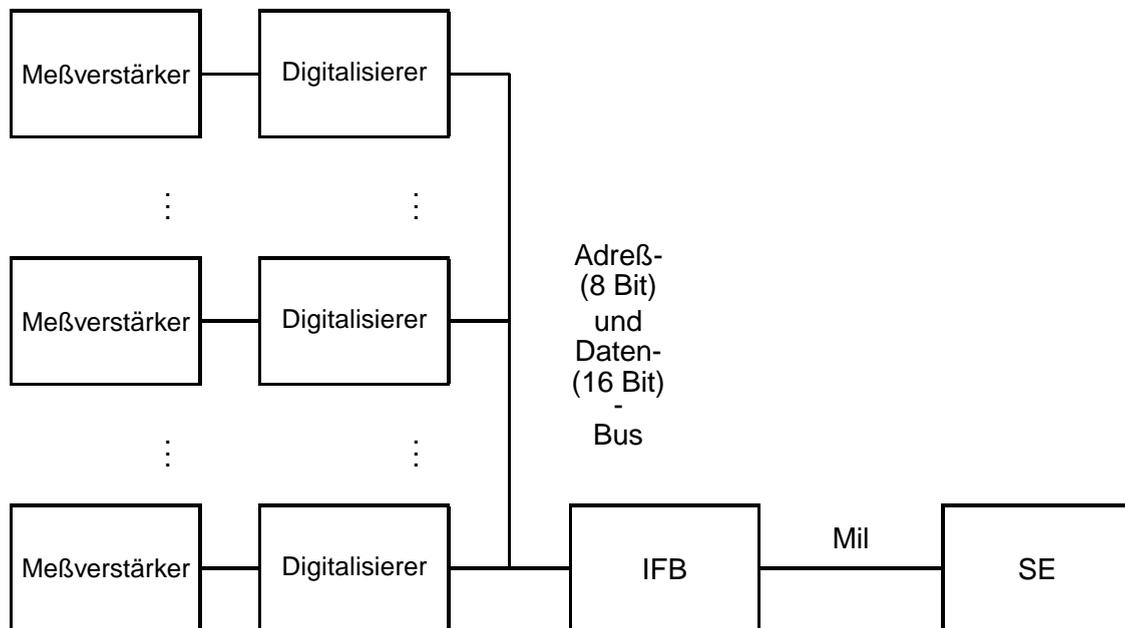
$$\text{Zeitählung} = t \cdot f_{takt}$$

Der Quotient

$$\frac{\text{Meßzählung}}{\text{Zeitählung}} = \frac{K \cdot I_{mess} \cdot t \cdot f_{takt}}{t \cdot f_{takt}} = K \cdot I_{mess}$$

### 3 Die Schnittstelle zum Gerät

#### 3.1 Beschreibung der Geräte-Komponenten



Jedes Gerät besteht aus zwei Karten.

1. Meßverstärker
2. Digitalisierer

Bis zu 8 Geräte sind über Adreß- und Datenbus und eine Interfacekarte (IFB) an einer SE angeschlossen. An den Meßverstärkern sind 5 Meßbereiche einstellbar:

Meßbereich	Meßbereichsendwert [A]	Auflösung [A]
1	100 E -3	100 E -6
2	10 E -3	10 E -6
3	1 E -3	1 E -6
4	100 E -6	100 E -9
5	10 E -6	10 E -9

Um die durch getrennte Timingbereiche gegebenen Grenzen – insbesondere für Transmissions-Überwachungsmessungen – zu überwinden, wurden sog. *Slave*-Elektroniken entwickelt. Dazu werden die Daten des letzten Trafos eines Timingbereichs einer Slave-Elektronik übergeben, die als erster Trafo des folgenden Timingbereichs definiert ist. Diese Slave-Elektronik benötigt keine Meßvorbereitung und ist ansonsten wie eine Master-Elektronik zu behandeln.

Bisher ist noch nicht ganz klar wie verfahren werden soll, wenn zwei Strahlzweige zu einem zusammengeführt werden. Dann erhält der weiterführende Strahlzweig 2 Slave-Elektroniken und muß anhand noch nicht spezifizierter Informationen ermitteln welche der beiden Elektroniken auszuwerten ist.

Pro Interfacekarte können 8 Elektroniken betrieben werden. Die Interfacekarten unterscheiden sich in ihren Adressen nur in den oberen 4 Bit.

Am Adreßbus werden die einzelnen Elektroniken ebenfalls nur durch die oberen 4 Bit unterschieden. Die unteren 4 Bit werden zur Unterscheidung der einzelnen Funktionen der Elektronik benutzt.

Jede Elektronik belegt mit ihren Teilfunktionen am Adreßbus 6 Adressen (auch *Sub-Adressen* genannt), die folgende Bedeutung haben:

Adreß-Offset	Datenfluß-Richtung	Bedeutung
0	Lesen	Status-Bits der Elektronik lesen
1	Schreiben	Rahmenpulsverzögerung (Start) schreiben (Sollwert muß negiert werden)
2	Schreiben	Rahmenpulsverzögerung (Stop) schreiben (Sollwert muß negiert werden)
3	Schreiben	Meßanforderung schreiben
4	Lesen	Meßzähler der Elektronik lesen
5	Lesen	Zeitähler der Elektronik lesen

Die Bits des Datenstatus der Elektronik sind bei Master- und Slave-Elektroniken unterschiedlich belegt.

- Statusbelegung bei Master-Elektronik:

Bit	Name	Bedeutung	
		High (1)	Low (0)
0	Überlauf Zeitmessung	ok	Überlauf
1	Signalaussteuerung	< 100% (ok)	> 100% (nicht ok)
2	Signalaussteuerung	< 10% (nicht ok)	> 10% (ok)
3	Single-Shot-Puls	kein Single-Shot	Single-Shot
4	Messung nicht beendet	ok	nicht beendet
5	Master/Slave-Kennung	Master	Slave
6	Gerätetypkennung	Trafo	Cup
7	Sequenzfehler	kein Sequenzfehler	Sequenzfehler

- Statusbelegung bei Slave-Elektronik:

Bit	Name	Bedeutung	
		High (1)	Low (0)
0 ... 2	Meßbereich mit dem die Masterelektronik gemessen hat		
3	??? Kalibriermessung ???	normale Messung	Kalibriermessung
4	unbenutzt	immer ok	–
5	Master/Slave-Kennung	Master	Slave
6	unbenutzt	immer ok	–
7	Summenstatus des Master	Master ok	Master nicht ok

Da mehrere Geräte über eine einzige Interfacekarte mit Adreß- und Daten-Bus angeschlossen werden können, muß bei der Bedienung einer einzelnen Elektronik darauf geachtet werden, daß diese auch (mit der richtigen Teilfunktion) am Adreßbus selektiert ist. Die einzelnen Soll- oder Istwerte erreicht man dann durch Schreiben oder Lesen des Datenbus.

Da dieses Verfahren aber immer gleich ist, könnte man für diesen speziellen Zweck auch einen eigenen MIL-Bus-Treiber zur Verfügung stellen, der aus der IFB\_Adr (4 Bit für die Adresse der Interfacekarte und 4 Bit für die Basis-Adresse der Elektronik am Adreßbus) und dem Funktionscode (selektiert als Offset zur Basis-Adresse der Elektronik die gewünschte Teilfunktion und die Richtung des Datentransfers) die richtige Elektronik selektiert und dann die Daten über den Datenbus transferiert.

Die SE zeigt der Elektronik mit der Meßanforderung, wie die nächste Messung ablaufen soll. Die Meßanforderung besteht aus einem 16 Bit Sollwert. Die einzelnen Bits haben folgende Bedeutung:

Bit	Erläuterung
0 ... 2	Meßbereich
3	Test-Eingang enable
4 ... 5	Rahmenpuls- Anwahl
6	frei
7	Fehlerauslösung per Software (d.h. dieser Puls wird blockiert)
8 ... 15	frei

Die Auswahl des Rahmenpulses (Gate) erfolgt mit den Bits 4 und 5. Damit könnten 4 Rahmenpulse adressiert werden, wovon jedoch nur 3 zur Verfügung stehen. Der vierte Rahmenpulseingang wird verwendet, um durch ein extern eingespeistes Signal einen bestimmten Strahlpuls als *den* Single-Shot-Puls zu kennzeichnen. Dies ist grundlegende Voraussetzung für die Realisierung einer Transmissionsmessung von der Quelle bis zum Experiment (auch über verschiedene Beschleuniger hinweg).

Mit Bit 7 wird eine Blockierung ausgelöst. Dies ist insbesondere dann interessant, wenn im Hochstrombetrieb die Stromüberwachung feststellt, daß in einem virtuellen Beschleuniger zu viel Strom zwischen zwei Meßstellen verlorengeht. Durch diese Blockierung am Gerät sollte hardwaremäßig der Chopper verriegelt werden.

### 3.2 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.

Da bei diesem Gerät nicht jedes Gerät auch eine eigene Interfacekarte besitzt, sondern immer bis zu 8 Geräte an einer Interfacekarte mit Adreß- und Daten-Bus angeschlossen sind, beschränken sich hier die verwendeten Funktions-Codes auf die wenigen, die man zum Handling von Adreß- und Daten-Bus benötigt und einige Standard-Codes.

Funktionscode		Modus	Bedeutung
Name	hex		
ifb_reset	01	Funktion	Reset
ifb_data_bus_w	10	Schreiben	16 Bit Daten auf den Datenbus schreiben
ifb_adr_bus_w	11	Schreiben	8 Bit Daten auf den Adreßbus schreiben
ifb_data_bus_r	90	Lesen	16 Bit Daten vom Datenbus lesen
ifb_rdstat_int	C9	Lesen	Status der Interfacekarte lesen

### 3.3 Umfang eines logischen Gerätes

Jeder Trafo hat eine eigene Nomenklatur und ist ein *logisches Gerät* im Sinne des Kontrollsystems. An einer Interfacekarte (mit Adreß- und Datenbus) sind bis zu 8 Digitalisierungseinheiten anschließbar. Jeder Steckplatz am Adreß- und Datenbus hat eine eigene Nummer (0 ... 7). Die physikalische Adresse eines Gerätes (der über die VME-Datenbank genau eine Nomenklatur zugeordnet wird) ergibt sich dann aus

Adresse der Interfacekarte + Steckplatznummer

Aus dieser Abbildung ergibt sich die Vorschrift, daß die Adresse einer Interfacekarte nur ein Vielfaches von 8 sein darf.

### 3.4 Definition der Bits des Hardwarestatus

Das Gerät liefert 1 Byte Statusinformation. Dabei handelt es sich allerdings nicht um Geräteinformationen, sondern um Informationen, die das letzte Meßergebnis betreffen (quasi ein Datenstatus). Die Bedeutung der einzelnen Bits ist in der folgenden Tabelle zusammengefaßt:

Die Bits 0 . . . 7 sind die systemweiten sogenannten generierten Softwarestatusbits (in engl. derived status bits).

Bit	Name	Bedeutung	
		High (1)	Low (0)
0	Power	on	off
1	Remote/Local	Remote	Local
2		reserved	
3		reserved	
4		reserved	
5	Interlock	no	yes
6	HW Error	no	yes
7	SW Error	no	yes
8	unbenutzt (immer high)		
⋮	⋮		
31	unbenutzt (immer high)		

### 3.5 Konfigurationsabfrage

Ein logisches Gerät ist ansprechbar und damit im Kontrollsystem vorhanden, wenn folgende Bedingungen erfüllt sind:

1. Nach Anwahl der entsprechenden Digitalisierungseinheit, am Adreßbus (siehe hierzu Abschnitt 3.3 auf Seite 11), kann von der Interfacekarte mit dem Funktionscode  $C0_{hex}$  (ifb\_rdstat) ein Status gelesen werden.
2. Im gelesenen Statuswort ist das höchstwertige Bit (MSB) *Low* (0).

Auf der Backplane des Einschubs ist dieses Statusbit über einen Pullup-Widerstand fest auf 5 V (*High*) verdrahtet. Eine selektierte Digitalisierungseinheit zieht diese Leitung nach 0 V (*Low*).

## 4 Die Bedienung des Gerätes

### 4.1 Aufgaben im Normalbetrieb

Bei der Digitalisierung eines Strahlpulses werden zwei Werte aufgenommen.

1. Die Amplitude des aufintegrierten Stromes
2. Die Dauer des Rahmenpulses

Aus beiden Werten wird ein Quotient ermittelt, der dem Pulsstrom entspricht. Über eine einstellbare Anzahl von Messungen wird ein Mittelwert gebildet.

Der Rahmenpuls soll den zu messenden Ausschnitt des Strahlpulses „umrahmen“, deshalb verwendet man dafür zweckmäßigerweise ein vom Chopper abgeleitetes Signal (bis zu 3 verschiedene Rahmenpulse können gewählt werden, je nachdem wo der Strahlpuls herkommt).

Wegen der Notwendigkeit kurze Integrationszeiten für die A/D-Wandlung zu ermöglichen und der endlichen Laufzeit von elektrischen Signalen, sowie der energieabhängigen Strahllaufzeit zwischen den einzelnen Meßeinrichtungen, ist es notwendig, das Meßzeitfenster (Rahmenpuls) zeitlich verschieben zu können. Hierzu wurde in die Digitalisierungselektronik eine programmierbare Verzögerungseinrichtung eingebaut, die Beginn und Ende des Rahmenpulses unabhängig voneinander zu verzögern gestattet.

#### 4.1.1 Multi-Multiturn-Injektion

Die Meßeinrichtungen ab TK7 werden nicht mehr vom Unilac- sondern vom SIS-Timing aus betrieben. D. h. an diesen Meßstellen (z. B. TK9DT8) läuft die Strommessung mit der im SIS verwendeten Beschleunigernummer und als Rahmenpuls wird der Chopperpuls des TK7BC1L verwendet. Im Falle einer *Multi-Multiturn-Injektion* werden alle injezierten Strahlpulse im TK einzeln gemessen und in einem Puffer gespeichert, der dann per Property gelesen werden kann.

## 4.2 Strahlüberwachung

Die Digitalisierungseinrichtungen eignen sich besonders für folgende Arten der Strahlüberwachung:

- Referenzwertüberwachung
- Sequenzüberwachung
- Stromverlustüberwachung

#### 4.2.1 Referenzwertüberwachung

Für jeden Trafo soll ein Referenzwert und eine maximal erlaubte Abweichung von diesem vorgebar sein. Abweichungen, die über diese Toleranz hinausgehen, führen zu einem *ExtDevSpecAlarm* im Kontrollsystem. Die Alarm-Extension soll folgende Informationen beinhalten:

1. Stromwert in A, der den Alarm ausgelöst hat
2. Referenzwert in A
3. maximal erlaubte Abweichung vom Referenzwert in A
4. Anzahl der aufgetretenen Fehler für diese Überwachungsart seit der letzten Quittierung.

Diese Überwachung soll zuschaltbar sein (default ist „disabled“).

#### 4.2.2 Sequenzüberwachung

Die Erkennbarkeit eines Referenzwertfehlers hängt ab von der einwandfreien Funktion der SE. Falls die SE nicht einwandfrei arbeitet, soll ein Sequenzfehler ausgelöst werden.

Vor jedem Puls muß eine Meßanforderung von der SE an die Elektronik gesendet werden. Tritt ein Rahmenpuls ohne vorangehende Meßanforderung auf, wird von der Elektronik ein Sequenzfehler ausgelöst. Dieser ist sofort wirksam und verriegelt den Chopper (Blockierung). Die Blockierung gilt, bis eine folgende Meßanforderung diesen Fehler zurücksetzt. Des Weiteren generiert die SE einen Sequenzfehler, wenn beim Anfordern der Meßwerte noch kein Rahmenpuls vorhanden war

(Status: busy). Dies kann vor allem bei einem Fehler in der Elektronik oder bei falscher Rahmenpulsauswahl auftreten.

Beide Fehler werden mit der Meßanforderung für einen neuen Puls zurückgesetzt. Bei der Realisierung der Gerätesoftware sollte unbedingt dafür gesorgt werden, daß die Anzahl der aufgetretenen Sequenzfehler (pro virtuellem Beschleuniger) und die Häufigkeit der Fehler (z.B. wieviele Pulse sind ok, bis zum nächsten Sequenzfehler) protokolliert wird.

### 4.2.3 Stromverlustüberwachung

Für jeden Trafo und jede Pulssorte soll ein maximal erlaubter Stromverlustwert einstellbar sein (als absoluter Stromwert).

Dieser Wert gibt an, wie groß die Stromdifferenz vom Vorgängertrafo bis zum aktuellen Trafo sein darf. Bei Überschreiten des erlaubten Werts, wird diese Pulssorte beim nächsten Erscheinen durch eine Fehlerauslösung an der Elektronik blockiert (siehe auch Abschnitt 4.2 auf Seite 13) und im Kontrollsystem ein *ExtDevSpecAlarm* verschickt. Die Alarm-Extension soll folgende Informationen beinhalten:

1. Stromwert in A, der den Alarm ausgelöst hat
2. die Stromdifferenz zum Vorgängertrafo in A
3. maximal erlaubte Stromdifferenz zum Vorgängertrafo in A
4. Anzahl der aufgetretenen Fehler für diese Überwachungsart seit der letzten Quittierung.

Voraussetzung für das funktionieren dieser Überwachungsart ist, daß der Stromwert des Vorgängertrafos auch bekannt ist. D.h. alle Trafos, zwischen denen der Stromverlust überwacht werden soll, müssen auch an derselben SE angeschlossen sein.

Eine bestehende Blockierung kann nur durch RESET oder INIT des Gerätes oder durch Rücksetzen der Überwachung aufgehoben werden. Diese Überwachung soll zuschaltbar sein (default ist „disabled“).

## 4.3 Genauigkeitsanforderungen

In einem Versuchsaufbau wurden die Grenzen ermittelt, bei denen die Meßgenauigkeit unter 1 % (bezogen auf den Meßwert) sank :

Maximale Taktfrequenz: 20 MHz Minimale Pulsbreite: 5  $\mu$ s

Für eine maximale Meßzeit von 8.2 ms ist eine Taktfrequenz von 8 MHz sinnvoll, die Zähler sind 16 Bit - Zähler. Die kleinste Pulsbreite für die geforderte Meßgenauigkeit ist dann 12  $\mu$ s.

## 4.4 Zeitkritische Anforderungen

Wie in Abschnitt 4.2.2 auf Seite 13 beschrieben, muß die Vorgabe der Meßanforderung immer vor dem Beginn des Rahmenpulses gesetzt werden, sonst blockiert die Elektronik den Chopper und verhindert den Strahltransport.

## 4.5 Einordnung in das Timing

### 4.5.1 Unilac

Ein Beispiel eines UNILAC Standardzyklus mit relevanten Events für die Strommessung ist in Abbildung 1 angegeben.

Die Konnektierungen sind in der nachfolgenden Tabelle zusammengefaßt.

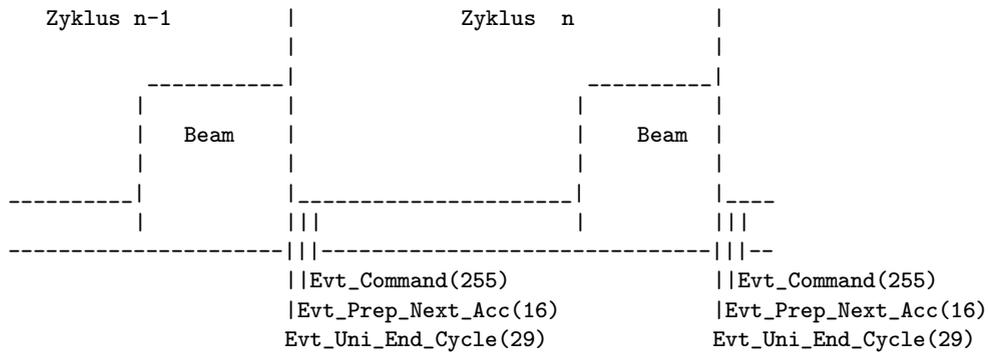


Abbildung 1: Standardzyklus (Unilac) mit relevanten Events für Strommessung

Aktion	Event-Name	Event-Nr.
Messung vorbereiten	Evt_Prep_Next_Acc	16
Messung auswerten	Evt_Uni_End_Cycle	29

Tabelle 4: Standard-Eventkonn timerungen für Strommessung (Unilac)

#### 4.5.2 SIS

Ein Beispiel eines SIS-Standardzyklus inkl. Multi-Multiturn-Injektion mit relevanten Events für die Strommessung ist in Abbildung 2 angegeben.

Die Konnektierungen sind in der folgenden Tabelle zusammengefaßt.

Aktion	Event-Name	Event-Nr.
Messung vorbereiten, MMI-Verwaltung rücksetzen	Evt_Start_Cycle	32
Vorbereitung an Gerät schicken	Evt_Uni_Ready	38
Meßdaten vom Gerät lesen (nur bei MMI)	Evt_Inj_Ready	108
Meßdaten vom Gerät lesen	Evt_Ramp_Start	43

Tabelle 5: Standard-Eventkonn timerungen für Strommessung (SIS)

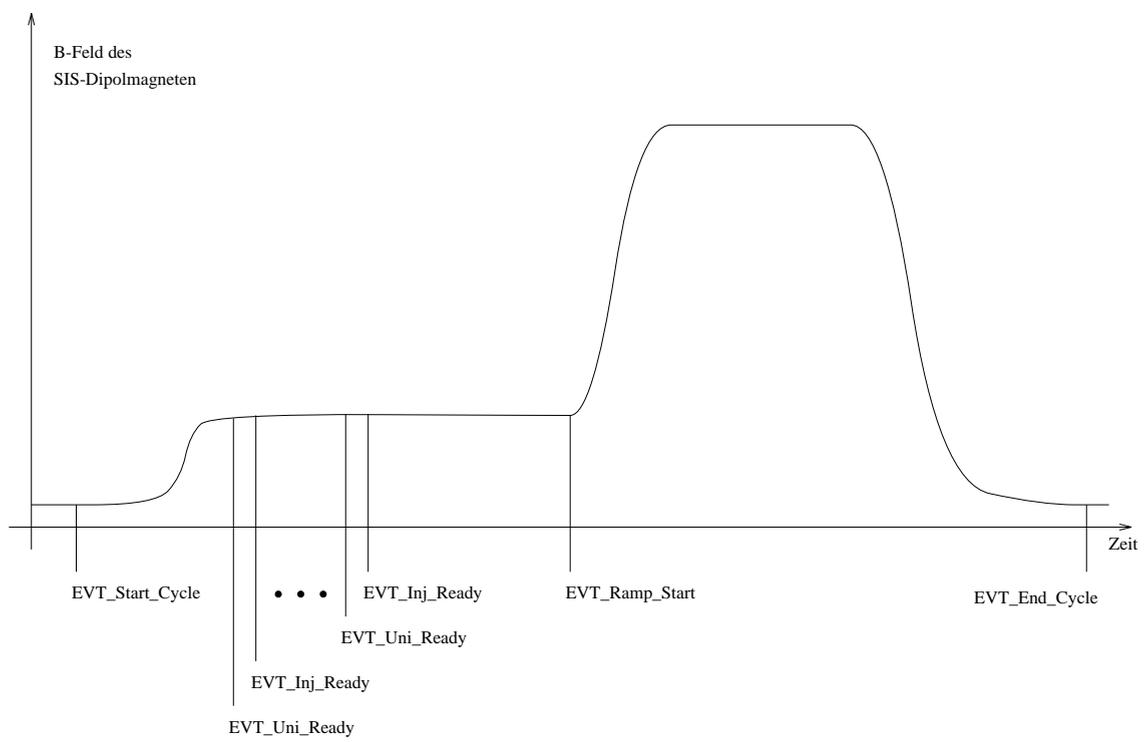


Abbildung 2: Standardzyklus (SIS) mit relevanten Events für Strommessung

## 4.6 Festlegung von Startwerten

### 4.6.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset durchgeführt.
- Der Gerätestatus wird ermittelt.
- Nur bei einem Kaltstart in Verbindung mit einem *powerup* werden folgende Initialisierungen durchgeführt:
  - Der unempfindlichste Meßbereich wird eingestellt.
  - Als Rahmenpuls wird der **externe Rahmen 1** selektiert.
- Die Grenzen der Überwachungsalgorithmen werden auf unempfindlich gesetzt.
- Referenzwertüberwachung und Stromverlustüberwachung werden disabled.
- Die Slave Istwerte werden mit 0 initialisiert und sowohl ein Eventstamp als auch Timestamp von 0 eingetragen.
- Der interne Fehlerpuffer wird zurückgesetzt.
- Die Standard-Eventkonnectierungen werden gesetzt (siehe Tabelle 4 auf Seite 15).
- Die SE wird in den Eventmode-Betrieb geschaltet (nur bei Kaltstart der SE).

### 4.6.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset durchgeführt.
- Der Gerätestatus wird ermittelt.
- Die Slave Istwerte werden mit 0 initialisiert und sowohl ein Eventstamp als auch Timestamp von 0 eingetragen.
- Der interne Fehlerpuffer wird zurückgesetzt.
- Alle Blockierungen, die durch die diversen Überwachungen ausgelöst worden sein könnten, werden aufgehoben.

## 4.7 Handbetrieb

Ein Handbetrieb des Gerätes ist nicht möglich.

## 4.8 Ableitung des Hardwarefehler-Bits aus dem Gerätestatus

Da kein richtiger Gerätestatus vorliegt, sondern nur Informationen über den Verlauf der letzten Messung, kann auch kein Hardwarefehler aus dem Status abgeleitet werden.

## 4.9 Verhalten bei Störungen

### 4.9.1 Geräteinterlock

Bei diesem Gerät nicht benutzt.

#### **4.9.2 Event-Sequenzfehler**

Die korrekte Reihenfolge der benutzten Events, ist unbedingt erforderlich, da sonst die Sequenzüberwachung in Aktion tritt (s. Abschnitt 4.2.2 auf Seite 13).

#### **4.9.3 Event-Overrun**

Event-Overruns können unberücksichtigt bleiben, solange sich das Overrunhandling nicht soweit verzögert, daß die Sequenzüberwachung in Aktion tritt.

#### **4.9.4 Ausfall der Kommunikation EC – Gerät**

Beim Ausfall der Kommunikation zwischen EC und Gerät tritt automatisch durch das Ausbleiben der Meßvorbereitung die Sequenzüberwachung in Kraft (siehe Abschnitt 4.2.2 auf Seite 13).

### **4.10 Bedienungsfehler vom Operating**

Alle Sollwerte werden auf ihre Korrektheit überprüft, und gegebenenfalls mit einem Fehler zurückgewiesen.

### **4.11 Single-Shot-Messung**

Analog zu den anderen vorhandenen Strommeßeinrichtungen muß es auch hier möglich sein, den Stromwert eines ganz bestimmten Pulses einer Pulssorte zur Transmissionsmessung explizit abzuspeichern.

Dazu wird am vierten Rahmenpulseingang, dessen Status über ein Bit im Datenstatus erkennbar ist, ein Signal angelegt, wenn der für Transmissionszwecke relevante Puls vorbeikommt. Dieser so *ausgezeichnete* Wert wird solange auf der SE zwischengespeichert und nicht überschrieben, bis mit SGLRESET (Single-Shot-Reset) die Messung wieder freigegeben worden ist.

Mit SGLCURR kann dieser Wert gelesen werden.

## 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).

### 5.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **DTI**. Die Gerätemodellnummer ist  $25_{dez}$ .

### 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	–	1	0
RESET	N	0	–	0	–	1	0
VERSION	RA	0	–	48	BitSet8	1	0
INFOSTAT	RA	0	–	25	BitSet32	1	0
CONSTANT	RA	0	–	11	RealF	1	0
CURRAVG	RA	0	–	2	RealF	1	0

#### 5.2.1 POWER

**Bedeutung:** Diese Property ist nur aus Kompatibilitätsgründen vorhanden, hat bei diesem Gerät aber keine reale Bedeutung.

**Parameter:** Keine.

**Daten:** Beim Lesen immer 1, d.h. das Gerät ist eingeschaltet. Jeder Schreibzugriff wird mit einer Fehlermeldung abgewiesen.

#### 5.2.2 STATUS

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

**Parameter:** Keine.

**Daten:** Das 32bit Statuswort. Die Bits entsprechen den Statusbits, wie sie in Abschnitt 3.4 auf Seite 12 und in der Tabelle 3.4 auf Seite 12 erklärt sind.

#### 5.2.3 INIT

**Bedeutung:** Initialisierung des Gerätes (Kaltstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.6 auf Seite 17.

**Parameter:** Keine.

**Daten:** Keine.

#### 5.2.4 RESET

**Bedeutung:** Reset des Gerätes (Warmstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.6.2 auf Seite 17.

**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 und Endwert). Diese gerätespezifischen Konstanten werden dem Operating als Tabelle zur Verfügung gestellt.

**Parameter:** keine

**Daten:** 11 RealF Werte

- 1** Anzahl der Meßbereiche
- 2** Meßbereichsendwert für Meßbereich 1 (in Ampere)
- 3** Meßbereichsauflösung für Meßbereich 1 (in Ampere)
- 4** Meßbereichsendwert für Meßbereich 2

- 5 Meßbereichsauflösung für Meßbereich 2
- ⋮
- 10 Meßbereichsendwert für Meßbereich 5
- 11 Meßbereichsauflösung für Meßbereich 5

### 5.2.8 CURRAVG

**Bedeutung:** Lesen des über die letzten 20 Messungen gemittelten Stromwertes. Dieser Wert ist hauptsächlich interessant für Programme, die sich für den Strom vor dem ersten Choppermagneten interessieren.

**Parameter:** keine

**Daten:** 2 RealF Werte

- 1 Anzahl der Strommittelungen
- 2 gemittelter Stromwert

## 5.3 Die Slave-Properties

*Bisher noch nicht endgültig festgelegt sind die Properties, die im Zusammenhang mit Hochstrom und den diversen Strahlüberwachungen künftig gebraucht werden. Weil bisher noch nicht zweifelsfrei geklärt ist, welche Daten in welcher Form benötigt werden.*

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	0	–	137	Integer32	1	0
GATESEL	RA/WA	0	–	3	BitSet16	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
CURRINFO	RA	0	–	13	RealF	1	0
SGLRESET	N	0	–	0	-	1	0
SGLCURR	RA	0	–	13	RealF	1	0
CTRLVAL	RA/WA	1	BitSet16	2/3	RealF	1	0
CTRLMODE	R/W	1	BitSet16	1	BitSet16	1	0
AVGCNTS	R/W	0	–	1	Integer16	1	0
AVGCNTI	R	0	–	1	Integer16	1	0
MMIINFO	RA	0	–	137	RealF	1	0

### 5.3.1 ACTIV

**Bedeutung:** Gibt an, ob das Gerät an der Puls zu Puls Modulation teilnehmen soll.

Dieses Gerät muß immer *active* für alle Beschleuniger sein, weil die Elektronik für

jeden Rahmenpuls, der ohne Meßvorbereitung kommt, einen Sequenzfehler auslöst (siehe hierzu auch Abschnitt 4.2.2 auf Seite 13).

**Parameter:** Keine.

**Daten:** Das Datum kann nur den Wert 1 annehmen. Jeder Versuch den Aktivstatus auf 0 zu setzen führt zu einem Fehler.

### 5.3.2 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.

### 5.3.3 EQMERROR

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

**Parameter:** Keine.

**Daten:** Die Anzahl der Fehlermeldungen sei bezeichnet durch:

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

Weiterhin soll gelten:

$$l = m + s$$

$$t = m + s + b$$

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

### 5.3.4 GATESEL

**Bedeutung:** GATESEL erlaubt es eine Rahmenpulsquelle (Gate) zu wählen.

**Parameter:** Keine

**Daten:** 3 BitSet16 Werte

- 1 Rahmenpulsnummer (1 bis 3)
- 2 Rahmenpuls-Start-Verzögerung (Zeiteinheit 80 ns)
- 3 Rahmenpuls-Stop-Verzögerung (Zeiteinheit 80 ns)

### 5.3.5 GAINMODS

**Bedeutung:** Dient zum ein- bzw. ausschalten der Meßbereichsautomatik. Bei eingeschalteter Meßbereichsautomatik paßt sich der Meßbereich der jeweiligen Strommenge automatisch an.

Der Automatikmode steht bei Trafos im SIS-Timing derzeit noch nicht zur Verfügung.

**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 schaltet selbständig und sofort den Meßbereich um, sobald der Datenstatus *Signalaussteuerung* > 100% anzeigt. Wird im Datenstatus jedoch *Signalaussteuerung* < 10% angezeigt, wird der Meßbereich nicht sofort angepaßt, sondern erst dann, wenn dies 10 mal in Folge im gleichen Beschleuniger passiert, oder eine Zeit von mindestens 5 Sekunden vergangen ist.
3. **Halbautomatisch** : Bei der Halbautomatik sucht die Meßbereichsautomatik selbständig nach dem passenden Meßbereich und stellt dann automatisch auf manuell zurück.

### 5.3.6 GAINMODI

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

**Parameter:** Keine

**Daten:** Siehe GAINMODS (5.3.5)

### 5.3.7 GAINRNGS

**Bedeutung:** Dient zum manuellen Setzen des Meßbereichs.

**Parameter:** Keine

**Daten:** 1 BitSet16 Wert, der die Meßbereichsnummer angibt (1 bis 5).

### 5.3.8 GAINRNGI

**Bedeutung:** Liest den aktuellen Meßbereich des Gerätes.

**Parameter:** Keine

**Daten:** Siehe GAINRNGS (5.3.7)

### 5.3.9 CURRINFO

**Bedeutung:** Mit dem Kommando CURRINFO wird ein 13 Worte großes Datenpaket gelesen, das alle Informationen zu einem Stromwert enthält.

**Parameter:** Keine

**Daten:** 13 RealF Werte

1. Stromwert in Ampere
2. Meßbereichsendwert in Ampere
3. Meßbereichsauflösung (0.01 entspr. 1%)
4. Integrationszeit in Sekunden
5. Meßbereichsnummer (Istwert) siehe 5.3.8
6. Meßbereichsnummer (Sollwert) siehe 5.3.7
7. Meßbereichsmodus (Istwert) siehe 5.3.6
8. Meßbereichsmodus (Sollwert) siehe 5.3.5
9. Aktivstatus siehe 5.3.1
10. Datenstatus siehe 3.1
11. gemittelter Stromwert in Ampere
12. Anzahl der Werte, die zur Strommittelung verwendet werden (Istwert)
13. Anzahl der Werte, die zur Strommittelung verwendet werden (Sollwert)

### 5.3.10 SGLRESET

**Bedeutung:** Gibt die Single-Shot-Messung wieder frei (siehe Abschnitt 4.11 auf Seite 18).

**Parameter:** Keine

**Daten:** Keine

### 5.3.11 SGLCURR

**Bedeutung:** Liest die Informationen zur letzten Single-Shot-Messung (siehe Abschnitt 4.11 auf Seite 18).

**Parameter:** Keine

**Daten:** Siehe CURRINFO (5.3.9)

### 5.3.12 CTRLVAL

**Bedeutung:** Setzen bzw. Lesen der Werte für eine Überwachungsart.

**Parameter:** 1 BitSet16 Wert, gibt die Überwachungsart an

1. Referenzwertüberwachung
2. Strahlverlustüberwachung

**Daten:** 3 RealF Werte (beim Schreiben nur 2), die je nach Überwachungsart unterschiedliche Bedeutung haben:

- Referenzwertüberwachung
  1. Referenzwert in A
  2. max. erlaubte Abweichung in A
  3. Anzahl der aufgetretenen Fehler für diese Überwachungsart seit der letzten Quittierung
- Strahlverlustüberwachung
  1. max. erlaubte Stromdifferenz zum Vorgängertrafo in A
  2. dummy (immer 0)
  3. Anzahl der aufgetretenen Fehler für diese Überwachungsart seit der letzten Quittierung

### 5.3.13 CTRLMODE

**Bedeutung:** Enable oder Disable einer Überwachungsart

**Parameter:** 1 BitSet16 Wert, gibt die Überwachungsart an

1. Referenzwertüberwachung
2. Strahlverlustüberwachung

**Daten:** 1 BitSet16 Wert gibt an, ob die Überwachung enabled oder disabled werden soll:

- 0:** disable  
**1:** enable

### 5.3.14 AVGCNTS

**Bedeutung:** Setzen bzw. Lesen der Anzahl von Meßwerten, die zur Bildung des Strommittelwertes verwendet werden.

**Parameter:** Keine

**Daten:** 1 BitSet16 Wert, der die Anzahl der zu mittelnden Meßwerte angibt.

### 5.3.15 AVGCNTI

**Bedeutung:** Lesen der Anzahl von Meßwerten, die zur Bildung des Strommittelwertes verwendet wurden.

**Parameter:** Keine

**Daten:** 1 BitSet16 Wert, der die Anzahl der gemittelten Meßwerte angibt.

### 5.3.16 MMIINFO

**Bedeutung:** Lesen aller Meßwerte der letzten Multi-Multiturn-Injektion.  
Diese Property steht nur für Trafos im SIS-Timing zur Verfügung.

**Parameter:** Keine

**Daten:** 9...137 RealF Werte

- 1:** Anzahl der Multi-Multiturn-Injektionen
- 2:** Meßbereichsendwert in Ampere
- 3:** Meßbereichsauflösung (0.01 entspr. 1%)
- 4:** Integrationszeit in Sekunden
- 5:** Meßbereichsnummer (Istwert) siehe 5.3.8
- 6:** Meßbereichsnummer (Sollwert) siehe 5.3.7
- 7:** Meßbereichsmodus (Istwert) siehe 5.3.6
- 8:** Meßbereichsmodus (Sollwert) siehe 5.3.5
- 9:** Aktivstatus siehe 5.3.1
- 10:** Stromwert der ersten Injektion in Ampere
- 11:** Datenstatus zum ersten Meßwert (siehe 3.1)
- :**
- 136:** Stromwert der 64-ten Injektion in Ampere
- 137:** Datenstatus zum 64-ten Meßwert



## Teil II

# Der Entwurf der Software

## 6 Softwareentwurf

Keine erwähnenswerten Besonderheiten.

## 7 Lokale Datenbasis

In der lokalen Datenbasis muß neben der Zuordnung *Gerätenomenklatur* zu *Geräteadresse* auch für jedes Gerät der *Default-Rahmenpuls* und die *Default-Delay-Zeit* (Start-/Stop-Delay) 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 des Rahmengerätes folgende gerätespezifischen USRs benötigt:

9.2.1 R\_ConstantS

Liefert die gerätespezifischen Konstanten.

9.2.2 R\_Currinfo

Liefert alle Daten zu einer Strommessung.

### **9.2.3 R\_SglCurr**

Liefert alle Daten zu einer Single-Shot-Strommessung.

### **9.2.4 N\_SglReset**

Gibt Single-Shot-Strommessung frei.

### **9.2.5 W\_GateselS**

Selektieren eines Rahmenpulses.

### **9.2.6 R\_GateselS**

Liefert die Nummer des selektierten Rahmenpulses.

### **9.2.7 W\_GainModS**

Setzen des Meßbereichsmodus.

### **9.2.8 R\_GainModS**

Lesen des Meßbereichsmodus (Sollwert).

### **9.2.9 R\_GainModI**

Lesen des Meßbereichsmodus (Istwert).

### **9.2.10 W\_GainRngS**

Setzen des Meßbereichs.

### **9.2.11 R\_GainRngS**

Lesen des Meßbereichs (Sollwert).

### **9.2.12 R\_GainRngI**

Lesen des Meßbereichs (Istwert).

### **9.2.13 W\_CtrlMode**

Enable/Disable einer Überwachungsart.

### **9.2.14 R\_CtrlMode**

Lesen des Enable/Disable-Status einer Überwachungsart.

### **9.2.15 W\_CtrlVal**

Setzen der Werte für eine Überwachungsart.

### **9.2.16 R\_CtrlVal**

Lesen der Daten für eine Überwachungsart.

### 9.2.17 W\_AvgCntS

Anzahl der Werte für Strommittelung festlegen.

### 9.2.18 R\_AvgCntS

Anzahl der Werte für Strommittelung lesen (Sollwert).

### 9.2.19 R\_AvgCntI

Anzahl der Werte für Strommittelung lesen (Istwert).

### 9.2.20 R\_MMIIInfo

Meßwerte der letzten MMI lesen (nur SIS).

### 9.2.21 R\_CurrAvg

Strommittelwert der letzten 20 Messungen (unabhängig vom virtuellen Beschleuniger) lesen.

## 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	Messung ist vorbereitet.

#### 10.1.2 Übergänge zwischen den Zuständen

Die Zustände und die Übergänge zwischen denselben sind in Tabelle 9 zusammengefaßt. Die Legende zu diesen Tabellen ist in Tabelle 10 zu finden.

#### 10.1.3 Standard-Zustandsübergänge

ready -> busy -> ready

### 10.2 Eventkonnektierte EQMs

#### 10.2.1 PrepUni\_EQM

**Event:** Evt\_Prep\_Next\_Acc (16).

**Aktion:** Die Messung im aktuellen Beschleuniger wird vorbereitet.

Tabelle der Zustandsübergänge				
von↓ nach→		error	ready	busy
error	U:	–	RESET, Zyklusende	Zyklusstart
	A:	–	Reset_EQM, CleanUp_EQM	Prep_EQM
ready	U:	Sequenzfehler etc.	–	Zyklusstart
	A:	div. EQMs	–	Prep_EQM
busy	U:	Sequenzfehler etc.	Zyklusende	–
	A:	Mess_EQM	Mess_EQM	–

Tabelle 9: Zustandsübergangsdiagramm

#### Legende

	Priorität der Zustände (höchste Priorität zuerst) error, ready, busy
U	auslösende Ursache <b>RESET</b> Reset wird per Kommando oder Knöpfchendrücken ausgelöst.
A	ausführende Stelle des Zustandsübergangs <b>..._EQM</b> Innerhalb des EQMs ..._EQM.

Tabelle 10: Legende zu den Zustandsübergangsdiagrammen

#### 10.2.2 MessUni\_EQM

**Event:** Evt\_Uni\_End\_Cycle (29).

**Aktion:** Lesen der Meßergebnisse und Berechnung des Stromwertes. Evtl. Durchführen der div. Überwachungen. Start des CalcUni\_EQM (s.A. 10.6.2 auf Seite 34) mit einer Verzögerung von 5 ms.

Die Verzögerung von 5 ms sorgt lediglich dafür, daß das EQM in jedem Fall innerhalb des nächsten Unilac-Zyklus ausgeführt wird, selbst dann, wenn dieser ein *Leer-Zyklus* sein sollte.

#### 10.2.3 PrepMMI\_EQM

**Event:** Evt\_Start\_Cycle (32).

**Aktion:** Der Datenpuffer für die Istwerte der *Multi-Multiturn-Injektion* wird initialisiert.

#### 10.2.4 PrepSIS\_EQM

**Event:** Evt\_Uni\_Ready (38).

**Aktion:** Die Messung im aktuellen Beschleuniger wird vorbereitet und der MMI-Puffer verwaltet.

#### 10.2.5 MessSIS\_EQM

**Event:** Evt\_Inj\_Ready (108) und Evt\_Ramp\_Start (43).

**Aktion:** Lesen der Meßergebnisse und Berechnung des Stromwertes. Eintragen der Meßergebnisse in den MMI-Puffer.

## 10.3 Periodisch konnektierte EQMs

### 10.3.1 Update\_Config\_EQM

**Zeit:** 10s

**Anzahl:** Unendlich.

**Aktion:** Aktualisieren der Geräteverfügbarkeit: Es wird versucht, mögliche Geräte am Adreßbus zu selektieren und anschließend den Status der Interfacekarte zu lesen. Erfolgt eine Reaktion, wird der Status der Interfacekarte ausgewertet und das *Online-Bit* überprüft (muß 0 sein). Ist dies der Fall, wird der Gerätestatus gelesen und überprüft, ob das Gerät ein Trafo ist.

Waren alle Überprüfungen erfolgreich, wird das Gerät als *online* geführt.

## 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

## 10.5 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.

## 10.6 Sonstige EQMs

### 10.6.1 Startup\_EQM

Installiert die Event-EQM-Konnektierung für alle virtuellen Beschleuniger (siehe hierzu auch Abschnitte 4.5 auf Seite 14 und 4.5.2 auf Seite 15) und schaltet die SE in den Event-Mode.

### 10.6.2 CalcUni\_EQM

Dieses EQM wird automatisch vom `MessUni_EQM` mit einer Verzögerung von 5 ms aufgerufen. Nach Berechnung des Strommittelwertes werden die Daten der einzelnen Überwachungsarten überprüft und ggf. die entsprechenden *ExtDevSpecAlarmer* erzeugt (siehe Abschnitte 4.2.1 und 4.2.3).

## 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

Lesen des aktuellen Gerätestatus.

## 11 Varianten

Die Varianten im Einzelnen:

**Uni** (DTI\$UNI.PIN) Trafos mit Unilac-Timing.

**SIS** (DTI\$SIS.PIN) Trafos mit SIS-Timing. Die Trafos im SIS-Timing können derzeit nicht im *Automatik-Mode* betrieben werden!

## 12 Besonderheiten

Verfahren zur Strahlverlustüberwachung:

Für jede Elektronik werden die aktuellen Meßwerte im Dual-Port-RAM abgelegt und der zugehörige Stromwert berechnet (absolut, in Ampere). Bei der nächsten Elektronik (nicht aber bei der ersten) wird dieser Wert zum Vergleich für die Stromverlustüberwachung benutzt. Auf diese Art ist gewährleistet, daß bei der Bedienung und Auswertung einer Elektronik nur auf die Datenstruktur dieser Elektronik schreibend zugegriffen wird, was die Synchronisierbarkeit und Konsistenzgewährleistung der Daten erleichtert.

# Index

## —Symbole —

Änderungsprotokoll ..... 2

## —A—

Abriß ..... 2  
Active\_EQM ..... 33  
Aufgabe des Gerätes ..... 7

## —B—

Bedienung des Gerätes ..... 12  
Bedienungsfehler ..... 18  
Besonderheiten ..... 34

## —C—

CalcUni\_EQM ..... 34

## —D—

Datenbasis ..... 29  
Dev\_Init\_EQM ..... 33  
Dev\_Reset\_EQM ..... 33  
Display\_DevErr\_EQM ..... 33  
Display\_DPR\_EQM ..... 33  
Dualport RAM ..... 29

## —E—

EQMs ..... 31

- Eventkonnektierte ..... 31
  - MessSIS\_EQM ..... 32
  - MessUni\_EQM ..... 32
  - PrepMMI\_EQM ..... 32
  - PrepSIS\_EQM ..... 32
  - PrepUni\_EQM ..... 31
- für die Diagnose vor Ort ..... 33
  - Display\_DevErr\_EQM ..... 33
  - Display\_DPR\_EQM ..... 33
- Globale Routinen ..... 34
  - Read\_and\_Update\_Status ..... 34
- Kommandogetriggerte ..... 33
  - Active\_EQM ..... 33
  - Dev\_Init\_EQM ..... 33
  - Dev\_Reset\_EQM ..... 33
  - Power\_EQM ..... 33

- Status\_EQM ..... 33
- Periodisch konnektierte ..... 33
  - Update\_Config\_EQM ..... 33
- Sonstige ..... 33
  - CalcUni\_EQM ..... 34
  - Startup\_EQM ..... 33

Event-Overrun ..... 18  
Event-Sequenzfehler ..... 18  
Eventkonnektierte EQMs ..... 31  
Eventkonnektierungen

- SIS ..... 15
- Unilac ..... 14

## —F—

Funktionscodes ..... 11

## —G—

Genauigkeitsanforderungen ..... 14  
Gerät

- Adresse ..... 11
- Aufgabe ..... 7
- Bedienung ..... 12
- Hardware ..... 7
- logisches ..... 11
- Repräsentation ..... 19
- Schnittstelle ..... 9

Gerätemodell ..... 7

- Kennzeichnung ..... 19
- Master-Properties ..... 19
- Slave-Properties ..... 22

Globale Routinen ..... 34

## —H—

Handbetrieb ..... 17  
Hardware des Gerätes ..... 7  
Hardwarefehler-Bit ..... 17  
Hardwarestatus ..... 12

## —I—

Init ..... 17  
Interfacekarte ..... 11  
Interlock ..... 17  
Interne Zustände ..... 31

—K—

Kaltstarts .....	17
Kommandogetriggerte EQMs .....	33
Konfigurationsabfrage.....	12

—L—

logisches Gerät .....	11
Lokale Datenbasis.....	29

—M—

Master-Properties.....	19
Meßanforderung .....	11
Meßgenauigkeit .....	14
MessSIS_EQM .....	32
MessUni_EQM.....	32
Multi-Multiturn-Injektion .....	13, 15

—N—

N_Init .....	29
N_Reset.....	29
N_SglReset .....	30
Normalbetrieb .....	12

—O—

Overrun .....	18
---------------	----

—P—

Periodisch konnektierte EQMs .....	33
Power_EQM .....	33
PrepMML_EQM.....	32
PrepSIS_EQM .....	32
PrepUni_EQM.....	31
Properties	
• ACTIV.....	22
• AVGCNTI .....	26
• AVGCNTS .....	26
• CONSTANT .....	21
• COPYSET .....	23
• CTRLMODE.....	26
• CTRLVAL .....	26
• CURRAVG .....	22
• CURRINFO .....	25
• EQMERROR .....	23
• GAINMODI.....	24
• GAINMODS .....	24

• GAINRNGI.....	25
• GAINRNGS.....	25
• GATESEL .....	24
• INFOSTAT .....	20
• INIT .....	19
• Master- .....	19
• MMIINFO .....	27
• POWER .....	19
• RESET .....	20
• SGLCURR .....	25
• SGLRESET .....	25
• Slave- .....	22
• STATUS .....	19
• VERSION.....	20

—R—

R.Active.....	29
R_AvgCntI .....	31
R_AvgCntS .....	31
R_ConstantS.....	29
R_CtrlMode .....	30
R_CtrlVal .....	30
R_CurrAvg .....	31
R_Currinfo.....	29
R_EQMErr .....	29
R_GainModI .....	30
R_GainModS .....	30
R_GainRngI .....	30
R_GainRngS .....	30
R_GateselS .....	30
R_InfoStat .....	29
R_MMInfo.....	31
R_Power .....	29
R_SglCurr .....	30
R_Status.....	29
R_Version .....	29
Referenzwertüberwachung.....	13
Repräsentation des Gerätes .....	19
Reset .....	17

—S—

Schnittstelle zum Gerät .....	9
Sequenzüberwachung .....	13
Sequenzfehler .....	18
SingleShot .....	18
Slave-Properties .....	22
Softwareentwurf .....	29
Softwarestatus .....	12
Sonstige EQMs .....	33
Störungen .....	17

- Event-Overrun ..... 18
- Event-Sequenzfehler ..... 18
- Interlock ..... 17
- Kommunikation EC – Gerät ..... 18

Standardzyklus SIS ..... 15

Standardzyklus UNILAC ..... 14

Startup\_EQM ..... 33

Startwerte ..... 17

Status\_EQM ..... 33

Statusbits ..... 12

Strahlüberwachung ..... 13

Stromverlustüberwachung ..... 14

—T—

Timing

- SIS ..... 15
- Unilac ..... 14

—U—

Update\_Config\_EQM ..... 33

USRs ..... 29

- gerätespezifische ..... 29
  - N\_SglReset ..... 30
  - R\_AvgCntI ..... 31
  - R\_AvgCntS ..... 31
  - R\_ConstantS ..... 29
  - R\_CtrlMode ..... 30
  - R\_CtrlVal ..... 30
  - R\_CurrAvg ..... 31
  - R\_Currinfo ..... 29
  - R\_GainModI ..... 30
  - R\_GainModS ..... 30
  - R\_GainRngI ..... 30
  - R\_GainRngS ..... 30
  - R\_GateselS ..... 30
  - R\_MMInfo ..... 31
  - R\_SglCurr ..... 30
  - W\_AvgCntS ..... 31
  - W\_CtrlMode ..... 30
  - W\_CtrlVal ..... 30
  - W\_GainModS ..... 30
  - W\_GainRngS ..... 30
  - W\_GateselS ..... 30
- obligatorische ..... 29
  - N\_Init ..... 29
  - N\_Reset ..... 29
  - R\_Active ..... 29
  - R\_EQMErr ..... 29
  - R\_InfoStat ..... 29
  - R\_Power ..... 29

- R\_Status ..... 29
- R\_Version ..... 29
- W\_Active ..... 29
- W\_CopySet ..... 29
- W\_Power ..... 29

—V—

Varianten ..... 34

- Betriebs- ..... 14
- Software- ..... 34

—W—

W\_Active ..... 29

W\_AvgCntS ..... 31

W\_CopySet ..... 29

W\_CtrlMode ..... 30

W\_CtrlVal ..... 30

W\_GainModS ..... 30

W\_GainRngS ..... 30

W\_GateselS ..... 30

W\_Power ..... 29

Warmstarts ..... 17

—Z—

Zeitkritische Anforderungen ..... 14

Zustände

- Interne ..... 31
  - Übergänge ..... 31
  - Bedeutung ..... 31
  - Standard-Übergänge ..... 31