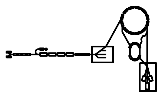

BCU - UNILAC-Chopper

Gerätemodell und Softwareentwurf

L. Hechler

GSI, Gesellschaft für Schwerionenforschung mbH
Planckstraße 1, D-64291 Darmstadt
Tel.: 06159/71-0

Issued: 24. Mär. 1999
Key: F-BCU-01
Printed: 17. Februar 2010

Änderungsprotokoll			
Datum	GM-Version	Name	Kommentar
24. 03. 1999	BCU_01	LH	Rahmen erstellt
30. 04. 1999	BCU_01	LH	Erste Ausbaustufe Istwerte

Inhaltsverzeichnis

I Das Gerätemodell **7**

1 Die Aufgabe des Gerätes **7**

2 Die Hardware des Gerätes **7**

3 Die Schnittstelle zum Gerät **8**
3.1 DACs und ADCs . . . 8
3.2 Jumperung von Interface- und DAC/ADC-Karte . . . 8
3.3 Funktionscodes der Interfacekarte . . . 8
3.4 Interlock Interrupt . . . 9
3.5 Data Request (DRQ) Interrupts . . . 9
3.6 Data Ready (DRD) Interrupts . . . 9
3.7 Definition der Bits des Hardwarestatus . . . 9

4 Die Ansteuerung des Gerätes **10**
4.1 Sollwert . . . 10
4.2 Istwerte . . . 10
4.3 Einordnung in das Timing . . . 11
4.4 Einschalten, Ausschalten . . . 11
4.5 Zeitkritische Anforderungen . . . 11
4.6 Festlegung von Startwerten . . . 12
4.6.1 Kaltstarts . . . 12
4.6.2 Warmstarts . . . 12
4.7 Handbetrieb . . . 12
4.8 Ableitung des Hardwarefehler-Bits aus dem Gerätestatus . . . 12
4.9 Verhalten bei Störungen . . . 12
4.9.1 Geräteinterlock . . . 12
4.9.2 Eventsequenz-Fehler . . . 12
4.9.3 Event-Overrun . . . 13
4.9.4 Emergency-Event . . . 13
4.9.5 Ausfall der Kommunikation EC – Gerät . . . 13
4.10 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 Überblick . . . 13
5.2.2 INFOSTAT . . . 14
5.2.3 INIT . . . 15
5.2.4 POWER . . . 15
5.2.5 RESET . . . 16
5.2.6 STATUS . . . 16
5.2.7 VERSION . . . 16
5.2.8 CONSTANT . . . 16
5.3 Die Slave-Properties . . . 16
5.3.1 Überblick . . . 16
5.3.2 ACTIV . . . 17
5.3.3 COPYSET . . . 17
5.3.4 EQMERROR . . . 17

5.3.5 VOLTAGES . . . 18
5.3.6 VOLTAGEI . . . 18

II Die Gerätesoftware 19

6 Softwareentwurf 19
6.1 EQM-Folge . . . 19
6.2 EQM-Folge mit Leerbeschleunigern . . . 20
6.3 EQM-Laufzeitmessungen an einem Prototyp . . . 21

7 Lokale Datenbasis 21
7.1 Tabelle der Konstanten . . . 21

8 Dualport RAM 21

9 USRs - User Service Routinen 23
9.1 Obligatorische USRs . . . 23
9.1.1 R_InfoStat . . . 23
9.1.2 N_Init . . . 23
9.1.3 R_Power . . . 23
9.1.4 W_Power . . . 24
9.1.5 N_Reset . . . 24
9.1.6 R_Status . . . 24
9.1.7 R_Version . . . 24
9.1.8 R_Active . . . 24
9.1.9 W_Active . . . 24
9.1.10 W_CopySet . . . 24
9.1.11 R_EQMError . . . 24
9.2 Gerätespezifische USRs . . . 24
9.2.1 W_VoltageS . . . 24
9.2.2 R_VoltageS . . . 24
9.2.3 R_VoltageI . . . 25
9.2.4 R_Constant . . . 25
9.3 Globale Routinen . . . 25
9.3.1 GetConst . . . 25

10 EQMs - Equipment Module 25
10.1 Interne Zustände . . . 25
10.1.1 Bedeutung der internen Zustände . . . 25
10.1.2 Übergänge zwischen den Zuständen . . . 26
10.1.3 Standard-Zustandsübergänge . . . 26
10.2 Eventkonnektierte EQMs . . . 26
10.2.1 VoltageS_EQM . . . 26
10.2.2 TriggerADC_EQM . . . 26
10.2.3 VoltageI_EQM . . . 26
10.3 Periodisch konnektierte EQMs . . . 26
10.3.1 Update_Config_EQM . . . 26
10.4 An externe Interrupts konnektierte EQMs . . . 26
10.4.1 Interlock_EQM . . . 26
10.4.2 DRD_EQM . . . 27
10.4.3 DRQ_EQM . . . 27
10.5 Kommandogetriggerte EQMs . . . 27

10.5.1 Dev_Init_EQM 27
10.5.2 Dev_Reset_EQM 27
10.5.3 Status_EQM 27
10.5.4 Active_EQM 27
10.5.5 Power_EQM 27
10.6 EQMs für die Diagnose vor Ort 27
10.6.1 Display_DPR_EQM 27
10.6.2 Display_DevErr_EQM 27
10.7 Sonstige EQMs 28
10.7.1 Startup_EQM 28
10.8 Globale Routinen 28
10.8.1 Read_and_Update_Status 28
10.8.2 Do_Intr_Service_Prep 28
10.9 MIL-Treiber 28

Index **29**

Teil I

Das Gerätemodell

1 Die Aufgabe des Gerätes

Der UNILAC-Chopper im Abschnitt UH2 dient zum Ausschneiden des zu beschleunigenden Strahlpulses aus dem Quellenpuls.

Länge und zeitliche Lage des Strahlpulses bestimmt die Interlockeinheit, die einen Rahmenpuls zur Steuerung des Choppers liefert, der der Länge und Lage des gewünschten Strahlpulses entspricht.

Der Chopper ist eine elektrostatische Einheit. Während des Rahmenpulses wird die Chopper-Spannung ausgetastet ($U = 0$). Der damit aus dem Quellenpuls ausgeschnittene (Teil-) Strahl gelangt ohne Ablenkung in den RFQ und wird weiter beschleunigt.

Die Länge des Pulses kann im Bereich ?

$$10\mu s \leq t_{\mathrm{Puls}} \leq x\mu s$$

liegen.

Strahl außerhalb des Pulses wird durch die Hochspannung an den Platten des Choppers abgelenkt und vernichtet.

2 Die Hardware des Gerätes

Der Chopper ist eine elektrostatische Einheit. Der Strahl wird durch ein elektrisches Feld beeinflusst. Die beiden Platten des Choppers werden mit einer gegen Erde symmetrischen Spannung von maximal ±15kV betrieben. Das NG wird mit der Spannungs-*Differenz* der Platten angesteuert, also mit 0...30kV. Dies ist die strahl-relevante Spannung. Das NG macht daraus die erforderlichen 0...±15kV.

Ebenso wird als Istwert die Differenzspannung vom NG geliefert. Für die Istwerte gibt es zwei Ausbaustufen:

Erste Ausbaustufe: Es wird nur ein Istwert geliefert. Der ADC wird von einem EQM getriggert, das mit dem Event `Pretrig_Beam` getartet wird.

Zweite Ausbaustufe: Als zweiter Istwert wird die Spannung im Gap geliefert, ebenso als Differenzspannung.

Bedingt durch die mögliche kurze Pulslänge muss zumindest der ADC für den Gap-Istwert extern getriggert werden. Der Trigger wird vom Rahmenpuls abgeleitet - evtl. mit Verzögerung. Womöglich ist es auch günstig, beide ADCs extern zu triggern. ?

Dies muss hardware-mäßig noch realisiert werden.

3 Die Schnittstelle zum Gerät

3.1 DACs und ADCs

Zum Setzen des Sollwertes und zum Lesen der Istwerte arbeitet das Gerät mit 16 Bit breiten DACs bzw. ADCs. Die Normierung ist wie folgt:

$$\begin{aligned} 0 &\mathrel{\widehat{=}} 0\,V \\ 7\mathrm{FFF}_{\mathrm{hex}} &\mathrel{\widehat{=}} 10\,V \end{aligned}$$

3.2 Jumperung von Interface- und DAC/ADC-Karte

Da der ADC nicht wie üblich frei läuft, sondern in der ersten Ausbaustufe von der Software getriggert wird, ist eine spezielle Jumperung auf der DAC/ADC-Karte notwendig, um diese Funktion zu ermöglichen. Die IFK behält ihre standard-mäßige Jumperung. Die Jumperung der IFK ist in Abbildung 1, die der DAC/ADC-Karte in Abbildung 2 dargestellt.

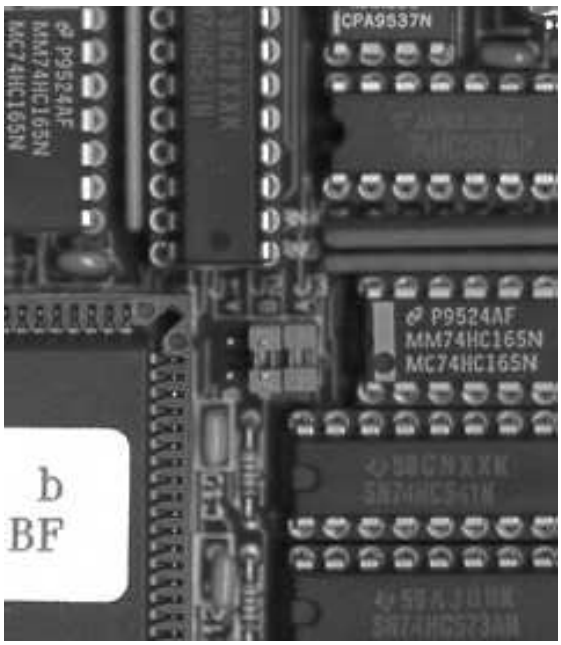

Abbildung 1: IFK FG 380.201

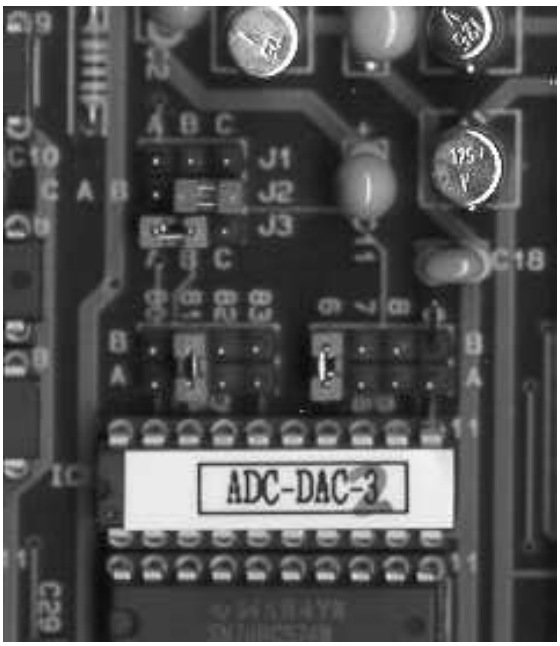

Abbildung 2: DAC/ADC FG 429.044

Für die zweite Ausbaustufe, wenn beide Istwerte gelesen und die ADCs durch externe Hardware getriggert werden, ändert sich die Jumperung noch einmal. Dies ist hier noch nicht dargestellt.

3.3 Funktionscodes der Interfacekarte

Funktionscode		Modus	Bedeutung
Name	Hex		
ifb_reset	01	Funktion	Reset
ifb_power_on	02	Funktion	Netz einschalten
ifb_power_off	03	Funktion	Netz ausschalten
ifb_soll_1	06	Schreiben	Spannung einstellen
ifb_intr_mask	12	Schreiben	Interruptmaske setzen
ifb_ist_1	81	Lesen	Spannung im Flattop lesen
ifb_ist_2	82	Lesen	Spannung im Gap lesen
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	Internen Interfacekartenstatus lesen

3.4 Interlock Interrupt

Interlocks werden nicht durch Interrupt gemeldet, sondern durch Polling festgestellt. Die Systemsoftware (ECM) wird entsprechend konfiguriert.

3.5 Data Request (DRQ) Interrupts

DRQ-Interrupts werden nicht erwartet bzw. verarbeitet.

3.6 Data Ready (DRD) Interrupts

DRD-Interrupts werden nicht erwartet bzw. verarbeitet.

3.7 Definition der Bits des Hardwarestatus

Die Bits 0...7 sind die systemweit einheitlichen Software-Statusbits. Die Statusbits im Einzelnen sind in der folgenden Tabelle zusammengefasst.

Bit	Name	Bedeutung	
		High (1)	Low (0)
0	Netz	ein	aus
1	Hoheit	Rechner	Hand
2	(reserviert)	–	–
3	(reserviert)	–	–
4	Emergency	nein	ja
5	Interlock	nein	ja
6	Hardware	ok	Warnung
7	Software	ok	Warnung
8	Netz	ein	aus
9			
10			
11			
12			
13			
14			

Forts. auf nächster Seite

Forts. von letzter Seite

Bit	Name	Bedeutung	
		High (1)	Low (0)
15			
16			
17			
18			
19			
20			
21			
22			
23			
24	Hoheit	Rechner	Hand
25			
26			
27			
28			
29			
30			
31			

4 Die Ansteuerung des Gerätes

Hier wird beschrieben, wie das Gerät (die Hardware) angesteuert werden muss. Das beinhaltet die Anforderungen *vom* Gerät als auch die Anforderungen *an das* Gerät.

4.1 Sollwert

Das Gerät erhält als Sollwert die Flattop-Spannung. Der Einstellbereich beträgt:

$$0 \leq U_{\text{Soll}} \leq 30\text{kV}$$

Dabei gilt folgende Normierung für die DAC-Werte:

$$7\text{FFF}_{\text{hex}} \mathrel{\widehat{=}} 30\text{kV}$$

4.2 Istwerte

Erste Ausbaustufe: Das Gerät liefert als Istwert die Flattop-Spannung.

Zweite Ausbaustufe: Das Gerät liefert die beiden Istwerte Flattop-Spannung und Gap-Spannung.

Die Normierung entspricht jeweils der des Sollwertes.

Istwerte werden *immer* gelesen, unabhängig vom Zustand des Gerätes - also auch, wenn das Gerät ausgeschaltet, inaktiv oder im Interlock ist. Ebenso wird der Istwert gelesen, wenn am Zyklusanfang kein Sollwert gesetzt wurde (Sequenzfehler). Die entsprechenden Fehlermeldungen werden trotzdem, wie üblich, generiert.

Voraussetzung für das Istwertlesen ist, dass der ADC getriggert wurde. Der Trigger wird vom EQM *immer* geschickt, unabhängig vom Zustand des Gerätes oder der Software.

Istwerte werden demnach *nicht* gelesen, wenn zuvor der ADC nicht getriggert wurde.

4.3 Einordnung in das Timing

Der Chopper hängt am Unilac-Timing.

Der Sollwert wird am Zyklusanfang zum Zeitpunkt des Events `Prep_Next_Acc` an das Gerät geschickt.

Erste Ausbaustufe: Der Istwert wird zum Zeitpunkt des Events `Pretrig_Beam` gemessen. D. h., dass zu diesem Zeitpunkt der ADC getriggert wird, der den Istwert konvertiert und zwischenspeichert. Der Trigger wird von einem EQM geliefert.

Zweite Ausbaustufe: Wann die Istwerte gemessen werden, steht noch nicht fest. ?

Gelesen, also zur SE übertragen, wird der konvertierte Istwert, wenn Zeit dazu ist.

Die Event-Konnektierungen sind in folgender Tabelle zusammengefasst.

Aktion	Event	Delay
Sollwert setzen	`Prep_Next_Acc`	0
ADC triggern	`Pretrig_Beam`	0
Istwert lesen	`Uni_End_Cycle`	1ms

Zur Trennung in Messen (ADC triggern) und Übertragen (Lesen) des Istwertes siehe Kapitel 6.1 auf Seite 19.

4.4 Einschalten, Ausschalten

Das Gerät schaltet innerhalb 1s ein und aus.

4.5 Zeitkritische Anforderungen

Zeitpunkt und Länge des Chopper-Fensters werden von der Interlockeinheit bestimmt, die dazu einen entsprechenden Rahmenpuls liefert. Die Gerätesoftware hat keine Möglichkeit zu überprüfen, ob der Rahmenpuls gekommen ist oder nicht. Ein indirekter Hinweis ist die gemessene Gap-Spannung, die im Normalfall Null sein sollte.

Start- und Laufzeit des TriggerADC-EQMs sind zeitkritisch. Siehe dazu Punkt „Timing“ im Kapitel 10.2.2 auf Seite 26.

Zweite Ausbaustufe mit 2 Istwerten:

- Bei Pulsen $< 50\mu s$ spielt die Konvertierungszeit des ADC (zur Zeit $55\mu s$) bereits eine Rolle. ? Man muss den Triggerzeitpunkt genau festlegen, um die Gap-Spannung richtig zu messen.

  Wie lange muss die zu messende Spannung tatsächlich anliegen, damit sie vom ADC korrekt ? konvertiert werden kann?

  Es ist eine ADC-Karte mit *Sample and Hold*-Technologie in Planung bzw. in Arbeit. Diese braucht dann weniger Messzeit.

- Wann genau sollen die beiden Istwerte gemessen werden? Flattop vor oder nach dem Gap? ?

4.6 Festlegung von Startwerten

4.6.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset durchgeführt.
- Alle Sollwerte werden für alle virtuellen Beschleuniger auf Null gesetzt.
- Die Istwerte, Eventstamp und Timestamp werden auf Null gesetzt.
- Das Gerät wird für alle virtuellen Beschleuniger aktiv geschaltet.
- Die Interlockbehandlung wird aktiviert.
- Die SE wird in den Eventmode-Betrieb geschaltet.
- Die Standard-Eventkonnektierungen werden gesetzt (siehe Tabelle auf Seite 11).

4.6.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset durchgeführt.
- Die Istwerte, Eventstamp und Timestamp werden auf Null gesetzt.
- Die Interlockbehandlung wird aktiviert.

4.7 Handbetrieb

Das Gerät kann auf Handbetrieb (local) geschaltet werden. Der Zustand wird im Status angezeigt. Nach dem Zurückschalten auf Rechnerbetrieb (remote) werden die aktuellen Sollwerte wieder realisiert. Lokal eingestellte Werte werden nicht übernommen.

4.8 Ableitung des Hardwarefehler-Bits aus dem Gerätestatus

4.9 Verhalten bei Störungen

4.9.1 Geräteinterlock

Tritt ein Interlock auf, wird die Spannung des Gerätes auf Null gesetzt.

4.9.2 Eventsequenz-Fehler

Kommen die relevanten Events nicht in der richtigen Reihenfolge (Sollwert setzen, ADC triggern, Istwert lesen) oder fehlt ein Event, wird ein Eventsequenz-Fehler gemeldet.

Die Erkennung von Eventsequenz-Fehlern ist durch die notwendige Schachtelung der EQMs einerseits und zeitkritischer EQM-Laufzeiten eventuell nicht möglich. Siehe dazu Kapitel 6.2 auf Seite 20.

4.9.3 Event-Overrun

Wird das Sollwert-EQM verspätet gestartet, wird kein Sollwert zum Gerät geschickt.

Das Istwert-EQM wird immer im Overrun gestartet. Das Delay in der Event-Konnektierung ist bewußt so gewählt, dass der Start des EQMs in den Lauf des Sollwert-EQMs fällt. Siehe dazu Kapitel 6.1 auf Seite 19.

4.9.4 Emergency-Event

Wird ein Emergency-Event empfangen, wird die Spannung des Gerätes auf Null gesetzt.

4.9.5 Ausfall der Kommunikation EC – Gerät

Der Ausfall der Kommunikation zwischen EC und Gerät führt zu Timeouts. Ein Timeout ist zu melden und mit der Bearbeitung des Zyklus fortzufahren.

4.10 Bedienungsfehler vom Operating

Sollwerte außerhalb des zulässigen Bereichs sind zurückzuweisen.

5 Die Repräsentation des Gerätes

Dieses Kapitel beschreibt, wie das Gerät nach höheren Ebenen hin abgebildet wird.

5.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **BCU**.

Die Gerätemodellnummer ist 63_{dez}.

5.2 Die Master-Properties

5.2.1 Überblick

In der Tabelle angegeben sind Name und Klasse der Property, Anzahl und Typ der Parameter, Anzahl und Typ der Daten sowie die physikalisch technische Einheit der Daten und der zugehörige Exponent zur Basis 10.

Liefert eine Property z. B. Milliampere (mA), dann ist die physikalisch technische Einheit 'A' und der Exponent '-3'.

INFOSTAT, INIT, POWER, RESET, STATUS und VERSION sind die für jedes Gerätemodell obligatorischen Properties.

Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
INFOSTAT	RA	0	–	25	BitSet32	1	0
INIT	N	0	–	0	–	–	–
POWER	R/W	0	–	1	BitSet16	1	0

Forts. auf nächster Seite

Forts. von letzter Seite

Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
RESET	N	0	–	0	–	–	–
STATUS	R	0	–	1	BitSet32	1	0
VERSION	RA	0	–	36	BitSet8	1	0
CONSTANT	RA	0	–	2	RealF	Volt	0

5.2.2 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.

5.2.3 INIT

Bedeutung: Initialisierung des Gerätes (Kaltstart). Für die dabei durchzuführenden Aktionen siehe Kapitel 4.6.1 auf Seite 12.

Parameter: Keine.

Daten: Keine.

5.2.4 POWER

Bedeutung: Gibt an, ob der Leistungsteil des Gerätes ein- oder ausgeschaltet ist bzw. werden soll. Hat bei diesem Gerät keine reale Bedeutung und ist nur aus Kompatibilitätsgründen vorhanden.

Parameter: Keine.

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

5.2.5 RESET

Bedeutung: Reset des Gerätes (Warmstart). Für die dabei durchzuführenden Aktionen siehe Kapitel 4.6.2 auf Seite 12.

Parameter: Keine.

Daten: Keine.

5.2.6 STATUS

Bedeutung: Auslesen des 32bit Gerätestatus.

Parameter: Keine.

Daten: Das 32bit Statuswort. Die Bits entsprechen den Statusbits, wie sie in Kapitel 3.7 auf Seite 9 und in der Tabelle 2 auf Seite 9 erklärt sind.

5.2.7 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 MIL-Treibers
37...48	Variante der EQMs

5.2.8 CONSTANT

Bedeutung: Lesen der Gerätekonstanten.

Parameter: Keine.

Daten: Die 2 Werte bedeuten im Einzelnen:

1: Minimal einstellbarer Wert der Spannung in Volt.
2: Maximal einstellbarer Wert der Spannung in Volt.

5.3 Die Slave-Properties

5.3.1 Überblick

In der Tabelle angegeben sind Name und Klasse der Property, Anzahl und Typ der Parameter, Anzahl und Typ der Daten sowie die physikalisch technische Einheit der Daten und der zugehörige Exponent zur Basis 10.

Liefert eine Property z. B. Milliampere (mA), dann ist die physikalisch technische Einheit 'A' und der Exponent '-3'.

ACTIV, COPYSET und EQMERROR sind die für jedes Gerätemodell obligatorischen 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
VOLTAGES	R/W	0	–	1	RealF	Volt	0
VOLTAGEI	RA	0	–	2	RealF	Volt	0

5.3.2 ACTIV

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

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Null heißt, das Gerät nimmt für den zugeordneten Beschleuniger *nicht* an der PPM teil bzw. soll *nicht* an der PPM teilnehmen. Eins heißt, das Gerät nimmt für den zugeordneten Beschleuniger an der PPM teil bzw. soll an der PPM teilnehmen.

5.3.3 COPYSET

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

Parameter: Keine.

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

Bemerkung: Die Nummer des zugehörigen ('eigenen') virtuellen Beschleunigers, in den die Geräteeinstellungen kopiert werden sollen, wird dem Standard entsprechend im XSR-Header geliefert.

5.3.4 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

$\vdots$

$m+1$: letzte Master-Fehlermeldung

$m+2$: erste Slave-Fehlermeldung

$\vdots$

$l+1$: letzte Slave-Fehlermeldung

$l+2$: Länge b des Fehlerpuffers

$l+3$: Zahl der Einträge im Fehlerpuffer

$l+4$: Index des ersten freien Platzes im Fehlerpuffer (der Fehlerpuffer ist ein Ringpuffer)

$l+5$: Erster Speicherplatz im Fehlerpuffer

$\vdots$

$t+4$: Letzter Speicherplatz im Fehlerpuffer

5.3.5 VOLTAGES

Bedeutung: Setze oder lese den Spannungs-Sollwert.

Parameter: Keine.

Daten: Der zu setzende bzw. der gelesene Spannungs-Sollwert in Volt.

5.3.6 VOLTAGEI

Bedeutung: Lese die beiden Spannungs-Istwerte.

Parameter: Keine.

Daten: Die gelesenen Istwerte in Volt.

Erster Wert: Flattop-Spannung.
Zweiter Wert: Gap-Spannung.

Bis zur Realisierung der zweiten Ausbaustufe wird für die Gap-Spannung 0,0 V zurückgeliefert.

Teil II
Die Gerätesoftware

6 Softwareentwurf

6.1 EQM-Folge

Die in diesem Kapitel getroffenen Aussagen beziehen sich ausschließlich auf die erste Ausbaustufe!

Die event-gebundenen Aktionen der EQMs sind in Kapitel 4.3 auf Seite 11 definiert. Da der Strahlpuls an jeder Stelle innerhalb des Quellenpulses liegen kann, ist das Event `Pretrig_Beam` fast über die gesamten zweiten 10ms eines 20ms-Zyklus' verschiebbar. Mit `Pretrig_Beam` wird der ADC getriggert (TriggerADC-EQM). Danach gibt es keinen, auch noch im Worst-Case freien Zeitpunkt im aktuellen Zyklus mehr, in dem das EQM zum Lesen des Istwertes (VoltageI-EQM) laufen könnte. Das Lesen des Istwertes muss damit in den nächsten Zyklus verschoben werden. Daraus ergibt sich das in Abbildung 3 dargestellte Timing mit verschachtelten EQMs.

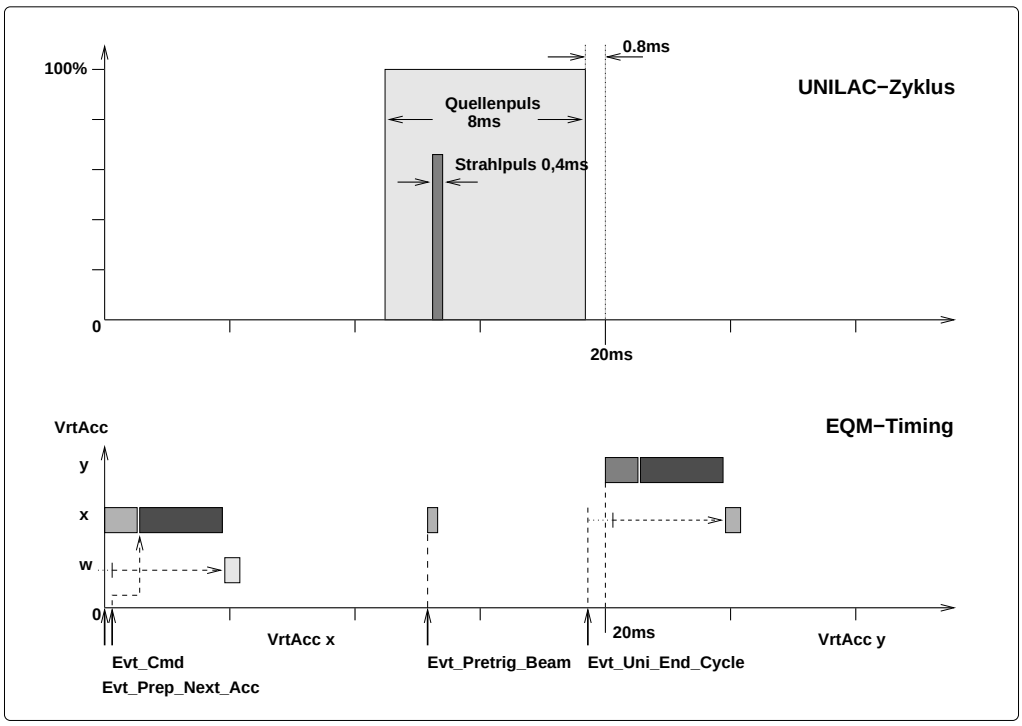

Abbildung 3: Timing mit verschachtelten EQMs

Im **VrtAcc x** läuft zunächst das VoltageS-EQM, das für **VrtAcc x** den Sollwert setzt, sporadisch gefolgt vom beschleuniger-unabhängigen UpdateConfig-EQM oder einem anderen Kommando-EQM. Anschließend läuft das VoltageI-EQM, das den Istwert aus dem vorhergehenden **VrtAcc w** liest, gefolgt vom TriggerADC-EQM, das den ADC zum Messen des Istwertes im **VrtAcc x** triggert. Das VoltageI-EQM zum Lesen des Istwertes aus **VrtAcc x** wird um 1ms verzögert vom Event `Uni_End_Cycle` gestartet, so dass es im folgenden Beschleuniger läuft. Dabei wird das Delay der Event-Konnektierung so gewählt, dass der Start des EQMs in den Lauf des nächsten

VoltageS-EQMs fällt, so dass es nach diesem im **VrtAcc y** läuft.

Das Delay für das VoltageI-EQM beträgt 1ms. Es ergibt sich aus folgenden Voraussetzungen:

Zykluslänge	= 20,0ms
Zyklusjitter	= $\pm 0,2\mu$s
Event `Uni_End_Cycle` bis Zyklusende	= 0,7ms
Laufzeit VoltageS-EQM	= 1,3ms

Das Delay muss so gewählt werden, dass bei Zykluslängen von 19,8ms bis 20,2ms der Start des VoltageI-EQMs *immer* in den Lauf des folgenden VoltageS-EQMs fällt.

Die Laufzeit des VoltageS-EQMs ergibt sich aus Prototyp-Messungen. Siehe dazu Kapitel 6.3.

6.2 EQM-Folge mit Leerbeschleunigern

Die Verschachtelung der EQMs ändert sich, wenn Leerbeschleuniger verschickt werden, in denen die für BCU relevanten Events nicht vorkommen.

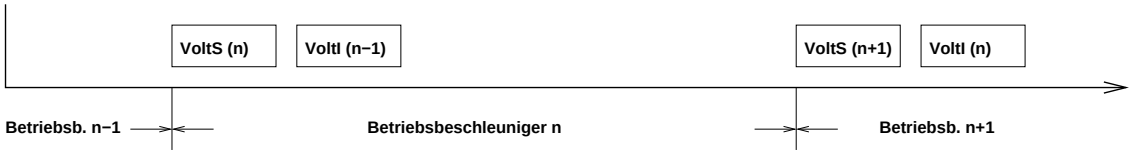

Abbildung 4: EQM-Folge ohne Leerbeschleuniger

Ein Superzyklus ohne Leerbeschleuniger ist vereinfacht in Abbildung 4 dargestellt. Hier erkennt man die im vorigen Kapitel beschriebene Verschachtelung. Nicht relevante EQMs sind dabei nicht dargestellt.

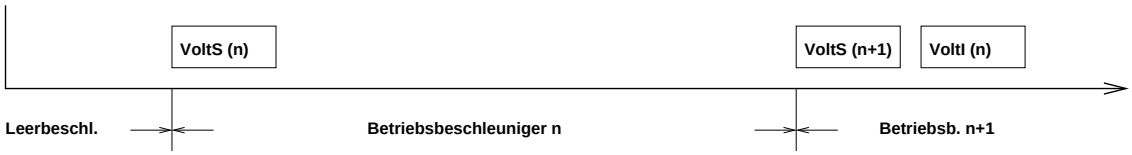

Abbildung 5: EQM-Folge nach einem Leerbeschleuniger

In Abbildung 5 folgen zwei Betriebsbeschleuniger auf einen Leerbeschleuniger. Da es im Leerbeschleuniger kein Event `Uni_End_Cycle` gibt, läuft auch das (mit Verzögerung zu startende) VoltageI-EQM nicht. Damit folgen zwei VoltageS-EQMs direkt aufeinander, was normalerweise zu einem Sequenzfehler führen würde. In diesem Fall ist das eine normale Betriebsart.

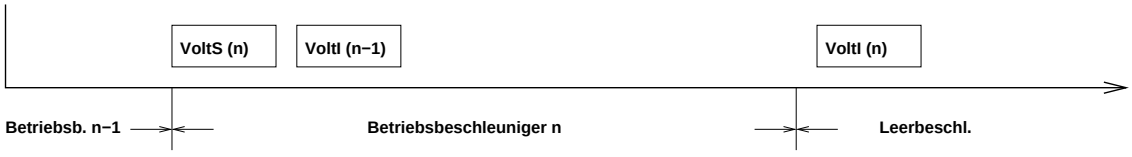

Abbildung 6: EQM-Folge in einem Leerbeschleuniger

Im umgekehrten Fall passiert ähnliches. In Abbildung 6 folgt ein Leerbeschleuniger auf einen Betriebsbeschleuniger. Im Leerbeschleuniger gibt es kein Event `Prep_Next_Acc` und somit auch kein VoltageS-EQM. Damit folgen zwei VoltageI-EQMs direkt aufeinander, was auch hier normalerweise zu einem Sequenzfehler führen würde. Auch diese ist aber hier eine normale Betriebsart.

Die EQMs VoltageS und VoltageI können nicht unterscheiden, ob es sich um einen echten Sequenzfehler handelt, oder ob ein Leerbeschleuniger im Spiel war. Sequenzfehler werden von diesen EQMs deshalb nicht überprüft.

6.3 EQM-Laufzeitmessungen an einem Prototyp

Messungen an einem Prototyp der EQMs (Ausbaustufe 1 mit nur einem Istwert) ergaben die in den Abbildungen 7 und 8 dargestellten EQM-Laufzeiten:

VoltageS-EQM	= 1,27ms
VoltageI-EQM	= 0,63ms
TriggerADC-EQM	= 0,35ms

Bei den Messungen galten folgende Randbedingungen:

Prozessortakt SEMAX:	20 MHz
Compiler:	Organon Pascal 211M
ECM-Version:	8.20
Anzahl Geräte:	1
USR-Version:	BCU 08.00.00
EQM-Version:	BCU 08.00.00
EQM-Variante:	–
SWPZ:	mit Worst-Case-HSI-Timing: Zykluslänge = 19,8ms.
Trigger:	Event `Prep_Next_Acc` vom TIF, im Kanal D1
EQM-Timing:	vom BiWa-Pin, im Kanal D2

7 Lokale Datenbasis

7.1 Tabelle der Konstanten

Für jedes Gerät gibt es eine Beschreibung bestehend aus 2 Elementen in der Konstantentabelle der lokalen Datenbasis. Die Elemente haben in der Reihenfolge folgende Bedeutung:

1: Minimal einstellbarer Wert der Spannung in Volt.

2: Maximal einstellbarer Wert der Spannung in Volt.

Siehe auch die Beschreibung der Property CONSTANT.

8 Dualport RAM

Der gerätespezifische Teil des Dualport-RAM enthält für ein Gerät folgende Elemente:

Master-Daten

m_sts 32 Bit Gerätestatus.

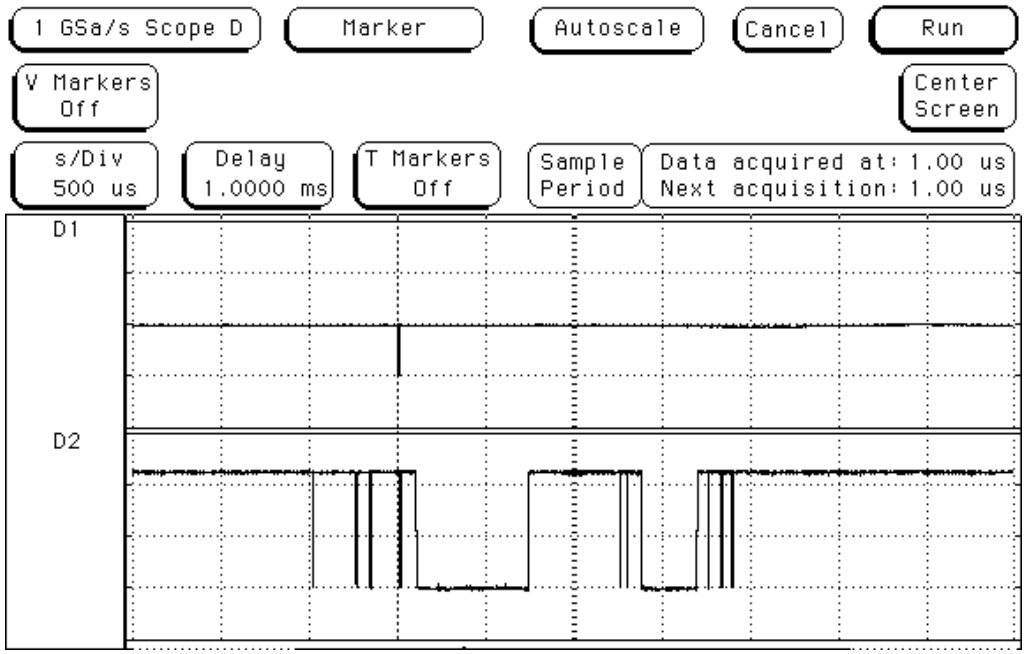

Abbildung 7: Laufzeit VoltageS-EQM und VoltageI-EQM

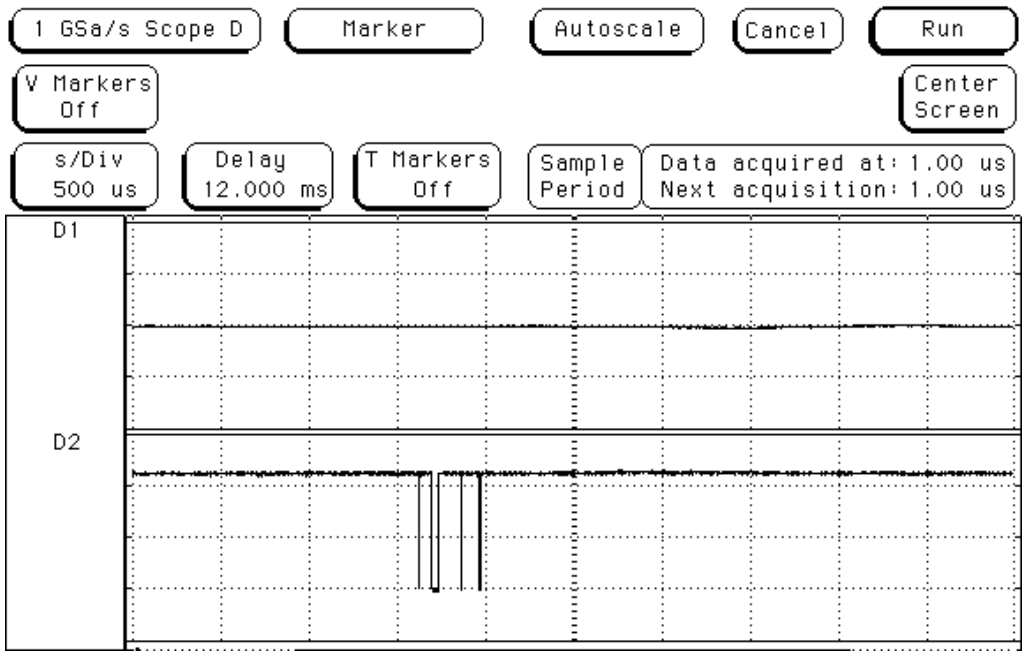

Abbildung 8: Laufzeit TriggerADC-EQM

int_sts Interner Gerätezustand.

const_actual Flag, das angibt, ob die Gerätekonstanten im DPR (dev_const) aktuell sind.

dev_const Die aus der lokalen Datenbasis geladenen Gerätekonstanten.

Slave-Daten

sync_rfc Wechselpuffer-Index für den Sollwert (voltages).

voltages Array aus 2 Elementen (Wechselpuffer), die jeweils den Sollwert enthalten.

sync_act Wechselpuffer-Index für die Istwerte (act).

act Array aus 2 Elementen (Wechselpuffer), die jeweils die beiden Istwerte und die Stamps enthalten:

voltagei_ft Flattop-Spannung.

voltagei_gap Gap-Spannung.

stamps Time- und Event-Stamp.

Geräte-übergreifende Daten

Im geräte-übergreifenden RAM (Dev_Commen_Buffer) wird die Zustandsvariable adc_sts untergebracht. Sie kann folgende Zustände einnehmen:

adc_not_set Initialer Zustand.

adc_triggered ADC wurde getriggert.

adc_ready ADC wurde ausgelesen und ist bereit für den nächsten Trigger.

Ein Zustand gilt jeweils für *alle* Geräte, da der ADC per Broadcast und somit für alle Geräte gleichzeitig getriggert wird.

9 USRs - User Service Routinen

9.1 Obligatorische USRs

9.1.1 R_InfoStat

Standard-Implementation.

9.1.2 N_Init

Standard-Implementation.

9.1.3 R_Power

Standard-Implementation.

9.1.4 W_Power

Standard-Implementation.

9.1.5 N_Reset

Standard-Implementation.

9.1.6 R_Status

Standard-Implementation.

9.1.7 R_Version

Standard-Implementation.

9.1.8 R_Active

Standard-Implementation.

9.1.9 W_Active

Standard-Implementation.

9.1.10 W_CopySet

BCU-eigene Implementation.

9.1.11 R_EQMError

Standard-Implementation.

9.2 Gerätespezifische USRs

Außer den obligatorischen USRs werden für die Steuerung des Choppers die folgenden gerätespezifischen USRs benötigt.

Alle gerätespezifischen USRs laden vor der Ausführung ihrer eigentlichen Aufgaben die Konstanten der lokalen Datenbasis ins Dualport-RAM, falls dies nicht bereits geschehen ist.

9.2.1 W_VoltageS

Test, ob Spannungswert innerhalb des definierten Bereichs liegt mit eventueller Fehlermeldung. Umrechnung Spannungwert (Real) in Bitmuster für DAC.

9.2.2 R_VoltageS

Umrechnung Bitmuster für DAC in Spannungswert (Real).

9.2.3 R_VoltageI

Berechnung Spannungwert (Real) aus Bitmuster des ADC.

9.2.4 R_Constant

Lesen der Gerätekonstanten.

9.3 Globale Routinen

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

9.3.1 GetConst

Überprufen, ob die Gerätekonstanten aus der lokalen Datenbasis bereits im Dualport-RAM liegen. Falls nicht, diese ins Dualport-RAM kopieren und als aktuell markieren.

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.

emergency Ein Emergency-Event wurde empfangen. Dieser Zustand darf nur durch Rücksetzen vom Operating verlassen werden.

interlock Ein Interlock wurde gemeldet. In einem periodisch ablaufenden Auftrag wird überprüft, ob die Interlock-Ursache noch vorliegt. Falls nein, Übergang nach ready.

local Das Gerät wird mit Handsteuerung betrieben.

power_off Das Gerät ist ausgeschaltet.

power_seq Das Gerät schaltet gerade ein oder aus.

inverting Der Polwender schaltet gerade.

error Während der Abarbeitung eines EQMs wurde ein Fehler erkannt. (besser Erklären).

ready Das Gerät ist bereit für Aktionen. Ausgangszustand am Beginn eines virtuellen Beschleunigers.

10.1.2 Übergänge zwischen den Zuständen

10.1.3 Standard-Zustandsübergänge

10.2 Eventkonnektierte EQMs

10.2.1 VoltageS_EQM

Event: `Prep_Next_Acc`.

Aktion: Setzen des Flattop-Spannungssollwertes.

10.2.2 TriggerADC_EQM

Event: Pretrig_Beam.

Aktion: Triggern des ADCs zur Messung des Spannungsistwertes auf dem Flattop.

Timing: Das Event `Pretrig_Beam` liegt minimal 200µs vor dem Strahlpuls, also vor dem Event Beam_On. Mit Beam_On wird das Gap des Choppers begonnen. Die Zeit zum Konvertieren des Flattop-Istwertes ist somit recht knapp. Messungen, die in [1, Punkt 6] dokumentiert sind, ergeben eine Zeit von 140µs zwischen Event und Auslösung des Triggers. Der ADC braucht dann nochmals 55µs zur Konvertierung. Damit ist die Messung 5µs vor dem Beginn des Gaps beendet.

10.2.3 VoltageI_EQM

Event: `Uni_End_Cycle` plus 1ms Delay.

Aktion: Lesen des Flattop-Spannungsistwertes. In der zweiten Ausbaustufe wird zusätzlich der Spannungsistwert im Gap gelesen.

Timing: Siehe dazu Kapitel 6.1 auf Seite 19.

10.3 Periodisch konnektierte EQMs

10.3.1 Update_Config_EQM

Zeit: 10s

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.

Pro Lauf des UpdateConfig-EQMs werden 10 Geräteadressen gescannt.

10.4 An externe Interrupts konnektierte EQMs

10.4.1 Interlock_EQM

Interrupt: Summen-Interlock.

Aktion: Internen Zustand auf ‘Interlock’ setzen, falls er nicht ‘Emergency’ ist. Sollwert Null setzen.

10.4.2 DRD_EQM

Interrupt: Data Ready Interrupt.

Aktion: Keine. Sollte bei MS nicht vorkommen.

10.4.3 DRQ_EQM

Interrupt: Data Request Interrupt.

Aktion: Keine. Sollte bei MS nicht vorkommen.

10.5 Kommandogetriggerte EQMs

10.5.1 Dev_Init_EQM

10.5.2 Dev_Reset_EQM

10.5.3 Status_EQM

10.5.4 Active_EQM

10.5.5 Power_EQM

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 Kapitel 4.3 auf Seite 11) und schaltet die SE in den Event-Mode.

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 MIL-Treiber

Zum Ansteuern der Interfacekarte wird der Standard-MIL-Treiber eingesetzt.

Literatur

[1] Ludwig Hechler. Laufzeitmessungen der MX-EQMs im Hochstromtiming, Weitere Ergänzungen. Accelerator Controls Note, Gesellschaft für Schwerionenforschung, Darmstadt, Dezember 1998. (Source: hsi/messungen_v3.tex).

Index

— **A** —

A/D-Wandler ..... *siehe* ADC
Active_EQM 27
ADC 8
  • Jumperung 8
  • Trigger 10
An Interrupts konnektierte EQMs 26
Ansteuerung des Gerätes 10
Aufgabe des Gerätes 7
Ausbaustufe
  • erste 7
  • zweite 7
Ausschalten 11

— **B** —

Bedienungsfehler 13

— **D** —

D/A-Wandler ..... *siehe* DAC
DAC 8
  • Jumperung 8
Datenbasis 21
Dev_Init_EQM 27
Dev_Reset_EQM 27
Display_DevErr_EQM 27
Display_DPR_EQM 27
DRD Interrupt 9
DRD_EQM 27
DRQ Interrupt 9
DRQ_EQM 27
Dualport RAM 21

— **E** —

Einschalten 11
Emergency-Event 13
EQMs 25
  • an Interrupts konnektierte 26
    – DRD_EQM 27
    – DRQ_EQM 27
    – Interlock_EQM 26
  • Eventkonnektierte
    – TriggerADC_EQM 26
    – VoltageI_EQM 26
    – VoltageS_EQM 26
  • eventkonnektierte 26
  • für die Diagnose vor Ort 27
    – Display_DevErr_EQM 27
    – Display_DPR_EQM 27
  • Folge 19
    – mit Leerbeschleunigern 20
  • Globale Routinen 28
    – Do_Intr_Service_Prep 28
    – Read_and_Update_Status 28
  • kommandogetriggerte 27
    – Active_EQM 27
    – Dev_Init_EQM 27
    – Dev_Reset_EQM 27
    – Power_EQM 27
    – Status_EQM 27
  • Laufzeit 21
  • MIL-Treiber 28
  • periodisch konnektierte 26
    – Update_Config_EQM 26
  • sonstige 28
    – Startup_EQM 28
  • Verschachtelung 19
    – mit Leerbeschleunigern 20
  • Zeitkritisches Timing 19, 21
Event
  • Prep_Next_Acc 10
  • Pretrig_Beam 10
  • Uni_End_Cycle 10
Event-Overrun 13
Eventkonnektierte EQMs 26
Eventkonnektierungen 11
Eventsequenz-Fehler 12

— **F** —

Folge von EQMs 19, 20
Funktionscodes 8

— **G** —

Gerät
  • Ansteuerung 10
  • Aufgabe 7
  • Hardware 7
  • Repräsentation 13
  • Schnittstelle 8
Gerätemodell 7
  • Kennzeichnung 13
  • Master-Properties 13
  • Slave-Properties 16

GetConst 25
Globale Routinen 25, 28

— H —

Handbetrieb 12
Hardware des Gerätes 7
Hardwarefehler-Bit 12
Hardwarestatus 9

— I —

Init 12
Interfacekarte 8
• Jumperung 8
• MIL-Treiber 28
Interlock 9, 12
Interlock_EQM 26
Interne Zustände 25
Interrupt
• DRD Interrupt 9
• DRQ Interrupt 9
• Interlock 9
Istwert 10

— K —

Kaltstarts 12
Kommandogetriggerte EQMs 27

— L —

Lokale Datenbasis 21
Lokalen Datenbasis
• Tabelle der Konstanten 21

— M —

Master-Properties 13
MIL-Treiber 28

— N —

N_Init 23
N_Reset 24
Normierung 8, 10

— O —

Overrun 13

— P —

Periodisch konnektierte EQMs 26
Power_EQM 27
Properties
• ACTIV 17
• CONSTANT 16
• COPYSET 17
• EQMERROR 17
• INFOSTAT 14
• INIT 15
• Master- 13
• POWER 15
• RESET 16
• Slave- 16
• STATUS 16
• VERSION 16
• VOKTAGEI 18
• VOLTAGES 18

— R —

R_Active 24
R_Constant 25
R_EQMError 24
R_InfoStat 23
R_Power 23
R_Status 24
R_Version 24
R_VoltageI 25
R_VoltageS 24
Repräsentation des Gerätes 13
Reset 12

— S —

Schnittstelle zum Gerät 8
Sequenzfehler 12
Slave-Properties 16
Software-Status 9
Softwareentwurf 19
Sollwert 10
Sonstige EQMs 28
Störungen 12
• Emergency-Event 13
• Event-Overrun 13
• Eventsequenz-Fehler 12
• Interlock 12
• Kommunikation EC – Gerät 13
Startup_EQM 28
Startwerte 12
Status_EQM 27

Statusbits 9

— T —

Timing 11
TriggerADC_EQM 26

— U —

Update_Config_EQM 26
USRs 23
- gerätespezifische 24
  - R_Constant 25
  - R_VoltageI 25
  - R_VoltageS 24
  - W_VoltageS 24
- Globale Routinen 25
  - GetConst 25
- obligatorische 23
  - N_Init 23
  - N_Reset 24
  - R_Active 24
  - R_EQMError 24
  - R_InfoStat 23
  - R_Power 23
  - R_Status 24
  - R_Version 24
  - W_Active 24
  - W_CopySet 24
  - W_Power 24

— V —

Verschachtelung von EQMs 19, 20
VoltageI_EQM 26
VoltageS_EQM 26

— W —

W_Active 24
W_CopySet 24
W_Power 24
W_VoltageS 24
Warmstarts 12

— Z —

Zeitkritische Anforderungen 11
- TriggerADC_EQM 26
- VoltageI_EQM 26
- VoltageS_EQM 26

Zustände
- interne 25
  - Übergänge 26
  - Bedeutung 25
  - Standard-Übergänge 26