

RPG - Rahmenpulsgenerator

Gerätemodell und Softwareentwurf

P. Kainberger

Dieses Papier enthält die Beschreibung des Gerätemodells ‘RPG - Rahmenpulsgenerator’ und den Entwurf der Gerätesoftware für dieses Gerät.

Zur präzisen Triggerung von Geräten (auch Strahldiagnose) entlang des UNILAC ist eine Verzögerungseinrichtung nötig, mit der die Strahlaufzeiten in der Maschine kompensiert werden können.

Änderungsprotokoll			
Datum	GM-Version	Name	Kommentar
29. Jan. 92	–	L. Hechler	Rahmen erstellt.
17. Mai 93	RPG_01	P. Kainberger	Adaption für RPG
26. Mai 93	RPG_01	P. Kainberger	Korrekturen u. Ergänzungen
14. Sep 93	RPG_01	P. Kainberger	Korrekturen u. Ergänzungen
15. Sep 93	RPG_02	P. Kainberger	Property DYNSTAT ergänzt
21. Dez 98	RPG_03	P. Kainberger	Umstellung auf Modulbusversion
28. Jan 99	RPG_03	P. Kainberger	Entprellung schaltbar
August 00	–	M. Kühn	Überarbeitete und erweiterte T _E X-Version, die sowohl in PostScript als auch in HTML konvertiert werden kann.
10. Jun 05	RPG_05	P. Kainberger	Erweiterungen durch Firmware 1.05 ergänzt
18. Oct 05	RPG_06	P. Kainberger	Property GPLENGTH ergänzt

Inhaltsverzeichnis

I	Das Gerätemodell	7
1	Die Aufgabe des Gerätes	7
2	Die Hardware des Gerätes	7
2.1	Technische Daten	7
2.2	Erweiterungen ab V01.05	8
2.3	Funktionen	8
3	Die Schnittstelle zum Gerät	10
3.1	Funktionscodes der Modulbus-Interfacekarte	11
3.2	Modulbus-Subadressen des Rahmenpulsgenerator	11
3.3	Interlock Interrupt	14
3.4	Data Request (DRQ) Interrupts	14
3.5	Data Ready (DRD) Interrupts	14
3.6	Umfang eines logischen Gerätes	15
3.7	Definition der Bits des Hardwarestatus	15
3.8	Konfigurationsabfrage	15
4	Die Bedienung des Gerätes	16
4.1	Aufgaben im Normalbetrieb	16
4.1.1	Sollwerte	16
4.1.2	Dynamischer Status	16
4.2	Genauigkeitsanforderungen	16
4.3	Zeitkritische Anforderungen	16
4.4	Einordnung in das Timing	16
4.5	Festlegung von Startwerten	17
4.5.1	Kaltstarts	17
4.5.2	Warmstarts	17
4.6	Handbetrieb	18
4.7	Verhalten bei Störungen	18
4.7.1	Event-Sequenzfehler	18
4.7.2	Event-Overrun	18
4.7.3	Ausfall der Kommunikation EC – Gerät	18
4.8	Bedienungsfehler vom Operating	18
5	Die Repräsentation des Gerätes	18
5.1	Kennzeichnung des Gerätemodells	18
5.2	Die Master-Properties	18
5.2.1	POWER	19
5.2.2	STATUS	19
5.2.3	INIT	19
5.2.4	RESET	19
5.2.5	VERSION	19
5.2.6	INFOSTAT	20
5.2.7	CONSTANT	21
5.2.8	SWITCH	21
5.3	Die Slave-Properties	22
5.3.1	ACTIV	22
5.3.2	EQMERROR	23

5.3.3	COPYSET	23
5.3.4	DYNSTAT	24
5.3.5	COUNTVAL	24
5.3.6	WORKMODE	24
5.3.7	TRIGDESC	24
5.3.8	OUTCHAN	25
5.3.9	TIFDESC	25
5.3.10	OUTACTIV	26
5.3.11	OUTTYPE	26
5.3.12	BURST	26
5.3.13	GPLENGTH	26
II	Der Entwurf der Software	27
6	Softwareentwurf	27
7	Lokale Datenbasis	27
8	Dualport RAM	27
9	USRs - User Service Routinen	27
9.1	Obligatorische USRs	27
9.1.1	N_Init	27
9.1.2	N_Reset	27
9.1.3	R_Status	27
9.1.4	R_Power	27
9.1.5	W_Power	27
9.1.6	R_Active	27
9.1.7	W_Active	27
9.1.8	W_CopySet	27
9.1.9	R_EQMErr	27
9.1.10	R_Version	27
9.1.11	R_InfoStat	27
9.2	Gerätespezifische USRs	27
9.2.1	R_Constant	27
9.2.2	W_Countval	28
9.2.3	R_Countval	28
9.2.4	R_DynStat	28
10	EQMs - Equipment Module	28
10.1	Interne Zustände	28
10.1.1	Bedeutung der internen Zustände	28
10.1.2	Übergänge zwischen den Zuständen	28
10.1.3	Standard-Zustandsübergänge	28
10.2	Eventkonnektierte EQMs	28
10.2.1	Prep_EQM	28
10.2.2	End_EQM	30
10.2.3	Start_EQM	30
10.3	Periodisch konnektierte EQMs	30
10.3.1	Update_Config_EQM	30
10.4	An externe Interrupts konnektierte EQMs	30
10.4.1	Interlock_EQM	30

10.4.2	DRD_EQM	30
10.4.3	DRQ_EQM	30
10.5	Kommandogetriggerte EQMs	31
10.5.1	Dev_Init_EQM	31
10.5.2	Dev_Reset_EQM	31
10.5.3	Status_EQM	31
10.5.4	Active_EQM	31
10.5.5	Power_EQM	31
10.6	EQMs für die Diagnose vor Ort	31
10.6.1	Display_DPR_EQM	31
10.6.2	Display_DevErr_EQM	31
10.7	Sonstige EQMs	31
10.7.1	Startup_EQM	31
10.8	Globale Routinen	31
10.8.1	Read_and_Update_Status	31
10.8.2	Do_Intr_Service_Prep	31
11	Varianten	32
12	Besonderheiten	32
Index		33

Abbildungsverzeichnis

1	Vergleich der Betriebsmodi <i>independent</i> und <i>transparent</i> , Schaubild zum <i>Burstmode</i>	9
2	Standardzyklus UNILAC mit relevanten Events für Rahmenpulsgenerator	16
3	Standardzyklus SIS mit relevanten Events für Rahmenpulsgenerator	17

Teil I

Das Gerätemodell

1 Die Aufgabe des Gerätes

Zur Erzeugung von zeitlich präzisen Gatepulsen entlang des UNILAC und TK wurde der sog. *Rahmenpulsgenerator* (auch *Gatepulsgenerator* genannt) entwickelt, mit dessen Hilfe die energieabhängige Strahllaufzeit entlang des UNILAC ausgeglichen werden kann.

Damit ist es z.B. möglich alle Geräte eines Beschleunigerabschnitts mit dem Timing des davorliegenden Choppers zu triggern. Die Signalverzögerung, die nötig ist, weil der Strahlpuls vom Chopper bis zum jeweiligen Meßort länger braucht als das Timingsignal, wird durch den *Rahmenpulsgenerator* mit einer Auflösung von 20 ns erzeugt.

Weiterhin soll der *Rahmenpulsgenerator* an den HF-Anlagen in SIS und evtl. ESR eingesetzt werden. Dort soll zum Ein- und Ausschalten der Phasenregelung durch den *Rahmenpulsgenerator* ein Gate erzeugt werden.

2 Die Hardware des Gerätes

2.1 Technische Daten

Modulbuskarte FG 450 380 3HE/4TE

EPLD-filename GPGEN1 (stimmt das noch???)

Modulbus-Ident-Code 38 (26_{Hex})

Anpaßkarte FG 450 680 (Anpassung der externen Ein-/Ausgänge)

Eingänge

- 8 externe Trigger-Eingänge, *high/low-active* schaltbar, TTL bis 50 Ohm, 1-polig Lemo isoliert
- Software-Trigger über Modulbus
- Eventbus, 2-polig Lemo isoliert
- 6 externe Status-Eingänge, mit Optokoppler, 1-polig Lemo isoliert (4 statisch, 2 dynamisch)

Ausgänge

- 8 externe Gatepuls-Ausgänge, *high/low-active* schaltbar, TTL bis 50 Ohm, 1-polig Lemo
- Monitorausgang des Triggerpulses auf Frontplatte, TTL bis 50 Ohm, 1-polig Lemo
- Monitorausgang des Gatepulses auf Frontplatte, TTL bis 50 Ohm, 1-polig Lemo
- TIF1: Eventtrigger-Ausgang auf Frontplatte, TTL bis 50 Ohm, 1-polig Lemo
- TIF2: Eventtrigger-Ausgang auf Frontplatte, TTL bis 50 Ohm, 1-polig Lemo

Start/Stop-Zähler 2 unabhängige 32-Bit-Zähler, mit 50 MHz (Standard) getaktet (Auflösung 20 ns)

Dynamischer Status Zustandsanzeige der Ablaufsteuerung, 2 externe Statusanzeigen

Statischer Status 4 externe Statusanzeigen, Anzeige des Taktgebers (Standard oder andere)

2.2 Erweiterungen ab V01.05

Event-Dekoder-A, B und C 3 zusätzliche Eventdekoder

Burstmode während der Dauer des Gatepulses kann ein alternierendes Signal programmiert werden

Ausgangsbelegung alle 8 Signalausgänge können unabhängig voneinander mit einem der bis zu 14 verschiedenen Signale belegt werden

Polarität die Polarität der Ausgangssignale ist kanalweise einstellbar

2.3 Funktionen

Betriebsmodus: Folgende Betriebsmodi sind vorgesehen:

independent mode: Beide Zähler werden mit der ersten Flanke des Eingangssignals gestartet.

transparent mode: Der Startzähler wird mit der ersten Flanke des Eingangssignals gestartet, der Stopzähler mit der zweiten Flanke desselben.

Triggerarten: Folgende Triggerarten sind vorgesehen:

Extern: Als Trigger wird 1 der 8 externen Trigger-Eingänge verwendet. Dabei muß festgelegt werden, nach welcher Logik (*high-* oder *low-active*) das Eingangssignal ausgewertet werden soll. Ebenso ist anzugeben, ob die Entprellung für den Triggereingang aktiviert werden soll oder nicht. Bei eingeschalteter Entprellung muß das Triggersignal mindestens für 300 ns anstehen, bevor es als gültiger Trigger akzeptiert wird. Ohne Entprellung reichen 20 ns aus.

Software: Als Trigger wird das Beschreiben einer festgelegten Subadresse über den Modulus per Software erwartet. Diese Triggerart ist nur im *independent mode* möglich.

Event: Als Trigger wird der Empfang eines einstellbaren Eventcodes (Start-Event) erwartet. Im *transparent mode* muß auch ein Stop-Event programmiert werden. Die Eventdecoder (Start- und Stop-Event) können als 8-, 12- oder 16-Bit Komparatoren verwendet werden.

Abbildung 1 auf Seite 9 zeigt schematisch die Unterschiede zwischen *independent-* und *transparent-mode* bei verschiedenen Triggerarten.

Single-/Repeat-Modus: Im *single-mode* wird nur der eingestellte Trigger nur einmal zugelassen und entsprechend auch nur ein Gatepuls erzeugt. Nach jedem Trigger muß also erneut vorbereitet und *enabled* werden.

Im *repeat-mode* wird zu jedem Trigger auch ein Gatepuls erzeugt und zwar so lange, bis per Software explizit *disabled* wird.

Data-Ready-Interrupt: Die 8 Low-Byte-Bits des dynamischen Status können zur Generierung eines DRDY-Interrupts maskiert werden.

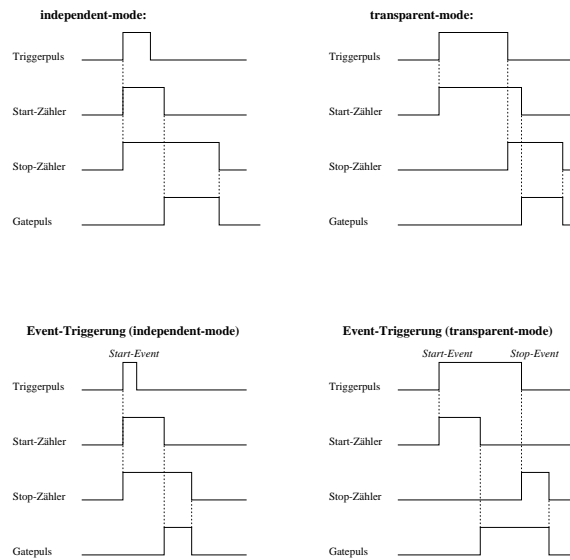


Abbildung 1: Vergleich der Betriebsmodi *independent* und *transparent*, Schaubild zum *Burstmode*

3 Die Schnittstelle zum Gerät

Der Rahmenpulsgenerator wurde als Modulbus-Karte entwickelt und muß über eine entsprechende Modulbus-Interfacekarte FG 450 012 angesteuert werden.

Der Modulbus ist eine Weiterentwicklung des *alten GSI-Bus-Systems* und wurde ca 1995 für den Umbau der Penning-Quellen entwickelt. Die Modulbuskomponenten werden über einen Adreßbus (13-Bit) selektiert und über einen Datenbus (16-Bit) gesteuert. Der Adreßbus ist folgendermaßen aufgeteilt:

Bit	Bedeutung
13...15	–
8...12	Modul-Adresse (5 Bit)
0...7	Sub-Adresse (8 Bit)

D. h. über die *Modul-Adresse* wird die Modulbuskomponente ausgewählt und über die *Sub-Adresse* ein Register der Modulbuskomponente. Mit einem anschließenden Lesen bzw. Schreiben vom bzw. auf den *Datenbus* wird dann ein Datum (16 Bit) vom bzw. zum Register der Modulbuskomponente gelesen bzw. geschrieben. In den folgenden Beschreibungen der Modulbuskomponenten werden die *Sub-Adressen* der einzelnen Modulbuskomponenten in ihrer Bedeutung und Funktion erläutert. Einige Sub-Adressen sind von zentraler Bedeutung und stehen deshalb für alle Modulbuskomponenten gleichermaßen zur Verfügung:

Sub-Adresse (Hex)	Zugriff (write/read)	Bedeutung	
		Bit 8...15	Bit 0...7
<i>FE</i>	r	Modul-ID (Karte)	Modul-ID (Steckplatz)
<i>FC</i>	r	Modulskalierung	Modul-Adresse
<i>FA</i>	r	EPLD-ID (Karte)	frei (<i>FF</i>)

Diese *Standard-Sub-Adressen* werden zur Ermittlung und Überprüfung der Modulbus-Konfiguration (im Sinne der *Autokonfiguration*) verwendet.

Erläuterung der Begriffe:

Modul-ID Jede Modulbuskomponente hat eine eindeutige **ID**entifikationsnummer. Diese wird zentral im Labor **BELAB** vergeben und ist auf der Modulbuskomponente fest verdrahtet. Auf der VG-Leiste des Modulbus-Steckplatzes muß beim Aufbau eines Modulbus-Einschubs die gleiche ID verdrahtet werden. Einige Modulbuskomponenten haben auf der Frontplatte eine *ID-Led*, die anzeigt, ob die ID der Karte mit der ID des Steckplatzes übereinstimmt.

Modulskalierung Einige Modulbuskomponenten sind sehr flexibel verwendbar deshalb wird über die *Modulskalierung* (fest verdrahtet auf der VG-Leiste des Steckplatzes) angezeigt wie diese Komponente auf diesem Steckplatz verwendet werden kann (z. B. ob die I/O-Karte **input** oder **output** kann).

Modul-Adresse Jede Modulbuskomponente muß innerhalb eines Modulbus-Einschubs über eine eindeutige Modul-Adresse verfügen.

EPLD-ID Versionskennung des auf der Modulbuskomponente integrierten EPLD-Bausteins.

3.1 Funktionscodes der Modulbus-Interfacekarte

Die für die Kommunikation mit der Modulbus-Interfacekarte 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 die Modulbus-Interfacekarte nur Treiberfunktion für den Modulbus hat, beschränken sich die verwendbaren 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

3.2 Modulbus-Subadressen des Rahmenpulsgenerator

Der Rahmenpulsgenerator (FG 450 380, *Modul-ID* 26_{hex}) stellt über den Modulbus folgende Funktionen zur Verfügung:

Sub-Adresse (Hex)	Zugriff oder Bit	Bedeutung
02	w	Sollwert Start-Zähler <i>low word</i> (Bits 0...15)
04	w	Sollwert Start-Zähler <i>high word</i> (Bits 16...31)
06	w	Sollwert Stop-Zähler <i>low word</i> (Bits 0...15)
08	w	Sollwert Stop-Zähler <i>high word</i> (Bits 16...31)
0A	w	Auswahl externe Trigger-Eingänge und externe Ausgänge
	0...2	Triggereingang 1...8 (kodierte)
	8...15	Auswahl externe Ausgänge 1...8
	8	externer Ausgang 1
	⋮	⋮
0C	15	externer Ausgang 8
	w	Auswahl der Bits im dynamischen Status, die DRDY-INR auslösen
	0	Gatepuls <i>active</i>
	1	Fehleranzeige <i>Trigger während Gatepuls</i>
	2	Fehleranzeige <i>Stoppuls vor Startpuls empfangen</i>
	3	<i>powerup detected</i> (nicht maskierbar)
0E	6	externer Status-Eingang 4
	7	externer Status-Eingang 5
	w/r	Einstellung der Triggerbedingung und Polarität
	0...1	Triggerbedingung (kodierte) 0: keine Triggerung 1: externe Triggerung 2: Software Triggerung

Forts. auf nächster Seite

Forts. von letzter Seite

Sub-Adresse (Hex)	Zugriff oder Bit	Bedeutung
	2	3: Event Triggerung Betriebsmodus 0: independent mode 1: transparent mode
	3	Single/Repeat-mode 0: Single mode 1: Repeat mode
	4	Logik für Triggereingang (0: positiv, 1: negativ)
	5	Logik für Ausgänge (0: positiv, 1: negativ)
	6	Entprellung der externen Triggereingänge: 0: Entprellung aus (Trigger muß mindestens 20 ns lang anstehen) 1: Entprellung ein (Trigger muß mindestens 300 ns lang anstehen)
	7	unbenutzt
	8...9	Bitbreite für Start-Event-Decoder 0: 16 Bit 1: 12 Bit 2: 8 Bit
	10...11	Bitbreite für Stop-Event-Decoder
	12...13	Bitbreite für TIF1-Event-Decoder
	14...15	Bitbreite für TIF2-Event-Decoder
10	w	<i>Enable-Register</i>
	0	enable Gatepuls <i>active</i>
	2	enable DRDY-Interrupt
	4	enable Start-Event
	5	enable Stop-Event
	6	enable TIF1-Event
	7	enable TIF2-Event
12	w	Eventmaske für Start-Event-Decoder
14	w	Eventmaske für Stop-Event-Decoder
16	w	Eventmaske für TIF1-Event-Decoder
18	w	Eventmaske für TIF2-Event-Decoder
1A	w	Software-Trigger (Daten sind dummy)
1C	w	Software-Reset (Daten sind dummy)
20	r	<i>statischer</i> Status
	0	externer Statuseingang 0
	1	externer Statuseingang 1
	2	externer Statuseingang 2
	3	externer Statuseingang 3
	4	1: 50 MHz Oszillator, 0: anderer (z. Zt. nicht möglich)
	5...15	unbenutzt
22	r	<i>dynamischer</i> Status
	0	Gatepuls <i>active</i>

Forts. auf nächster Seite

Forts. von letzter Seite

Sub-Adresse (Hex)	Zugriff oder Bit	Bedeutung
	1	Fehleranzeige <i>Trigger während Gatepuls</i>
	2	Fehleranzeige <i>Stoppuls vor Startpuls empfangen</i>
	3	<i>powerup detected</i> (nicht maskierbar)
	4..5	unbenutzt
	6	externer Status-Eingang 1
	7	externer Status-Eingang 2
	8	Gatepuls <i>enabled</i>
	9	<i>waiting for trigger</i>
	10	<i>waiting for gatepuls</i>
	11	<i>DRDY-INR enabled</i>
	12	Start-Event <i>enabled</i>
	13	Stop-Event <i>enabled</i>
	14	TIF1-Event <i>enabled</i>
	15	TIF2-Event <i>enabled</i>
24	w	Burstmode: High-Count <i>low word</i> (Bits 0..15)
26	w	Burstmode: High-Count <i>high word</i> (Bits 16..21)
28	w	Burstmode: Low-Count <i>low word</i> (Bits 0..15)
2A	w	Burstmode: Low-Count <i>high word</i> (Bits 16..21)
30	w	Eventmaske für Event-Decoder-A
32	w	Eventmaske für Event-Decoder-B
34	w	Eventmaske für Event-Decoder-C
36	w	Eventmaske für Event-Decoder-D (aus Platzgründen nicht implementiert)
38	w	<i>Event-Control-register</i>
	0..3	Event-Decoder-A
	0	enable
	1..2	Bitbreite für Event-Decoder
		0: 16 Bit
		1: 12 Bit
		2: 8 Bit
	3	Pulsbetrieb (0: disable, 1: enable)
	4..7	Event-Decoder-B
	4	enable
	5..6	Bitbreite für Event-Decoder
	7	Pulsbetrieb (0: disable, 1: enable)
	8..11	Event-Decoder-C
	8	enable
	9..10	Bitbreite für Event-Decoder
	11	Pulsbetrieb (0: disable, 1: enable)
	12..15	Event-Decoder-D (aus Platzgründen nicht implementiert)
3A	w	<i>Mux-Output-3-0</i>
		0: Gate-Puls
		1: Start-Event (nur wenn Triggerart auf Event steht)

Forts. auf nächster Seite

Forts. von letzter Seite

Sub-Adresse (Hex)	Zugriff oder Bit	Bedeutung
		2: Stop-Event (nur wenn Triggerart auf Event steht) 3: TIF1-Event (aktiviert Pulse-2) 4: TIF2-Event (deaktiviert Pulse-2) 5: A-Event (aktiviert Pulse-3) 6: B-Event (deaktiviert Pulse-3) 7: C-Event 8: D-Event (aus Platzgründen nicht implementiert) 9: Pulse-1 (von Start- bis Stop-Event) 10: Pulse-2 (von TIF1- bis TIF2-Event) 11: Pulse-3 (von A- bis B-Event) 12: Pulse-4 (aus Platzgründen nicht implementiert) 13: Burstmode (nur während des Gate-Pulses)
3C	0...3	Belegung Ausgangskanal 0
	4...7	Belegung Ausgangskanal 1
	8...11	Belegung Ausgangskanal 2
	12...15	Belegung Ausgangskanal 3
	w	<i>Mux-Output-7-4</i>
3E	0...3	Belegung Ausgangskanal 4
	4...7	Belegung Ausgangskanal 5
	8...11	Belegung Ausgangskanal 6
	12...15	Belegung Ausgangskanal 7
	w	<i>Out-Polarity</i>
	0	Polarität Ausgangskanal-0 (0: positiv, 1: negativ)

	7	Polarität Ausgangskanal-7 (0: positiv, 1: negativ)
	8..15	unbenutzt
40	r	Gatepulse-Length <i>low word</i> (Bits 0...15)
42	r	Gatepulse-Length <i>high word</i> (Bits 16...24)
	25	counter-overflow
	26	gate already active

3.3 Interlock Interrupt

Ein Summeninterlock wird vom Gerät nicht generiert.

3.4 Data Request (DRQ) Interrupts

Ein DRQ-Interrupt wird vom Gerät nicht generiert.

3.5 Data Ready (DRD) Interrupts

Der Rahmenpulsgenerator kann so eingestellt werden (über Subadresse $0C_{Hex}$), daß er bestimmte Änderungen im *dynamischen Status* als DRD Interrupt anzeigt. S. hierzu Abs. 2 auf Seite 11.

3.6 Umfang eines logischen Gerätes

Jede Rahmenpulsgenerator-Modulbus-Karte repräsentiert ein logisches Gerät im Sinne des Kontrollsystems.

3.7 Definition der Bits des Hardwarestatus

Das Gerät liefert 1 Wort (16 Bit) Statusinformation und ein Wort *dynamischen* Status (Informationen, die sich von Puls zu Puls ändern können).

Nur der *statische* Status wird im Hardwarestatus des Gerätes abgebildet (Bits 8...11), während der *dynamische* Status nur zur Überwachung des Timings von Puls zu Puls dient.

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

Die Statusbits des Hardwarestatus 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	externer Statuseingang 0		
9	externer Statuseingang 1		
10	externer Statuseingang 2		
11	externer Statuseingang 3		
12	unbenutzt	immer high	–
⋮	⋮	⋮	⋮
15	unbenutzt	immer high	–
16	unbenutzt	immer high	–
⋮	⋮	⋮	⋮
31	unbenutzt	immer high	–

3.8 Konfigurationsabfrage

Ein logisches Gerät ist ansprechbar und damit im Kontrollsystem vorhanden, wenn an einem Modulbus eine Karte mit der ID des Rahmenpulsgenerators (26_{Hex}) gefunden wird. Dabei müssen folgende Randbedingungen eingehalten werden:

- Die Adresse der Modulbus-Interfacekarte muß ein Vielfaches von 16 sein.
- Als Moduladresse der Rahmenpulsgeneratoren innerhalb eines Modulbusses sind Werte von 1 bis 16 erlaubt.

Beide Adressen zusammen (abzgl. 1) ergeben dann die tatsächliche physikalische Geräteadresse im Sinne des Kontrollsystems.

4 Die Bedienung des Gerätes

4.1 Aufgaben im Normalbetrieb

4.1.1 Sollwerte

Als Sollwerte müssen zu Beginn eines Zyklus die Zählersollwerte für den Start- und den Stop-Zähler (jeweils 32 Bit) gesetzt werden. Die Werte müssen dem vorhandenen Taktgeber entsprechend normiert werden.

4.1.2 Dynamischer Status

Anhand des dynamischen Geratestatus können während der Bedienung evtl. aufgetretene Fehler festgestellt werden.

4.2 Genauigkeitsanforderungen

Die Zählerauflösung erreicht bei 50 MHz eine Auflösung von 20 ns. Mit der verfügbaren Datenbreite von 32 Bit sind Verzögerungszeiten bis 42 s möglich. Dies erscheint für alle bisher bekannten Anwendungen ausreichend.

4.3 Zeitkritische Anforderungen

Die Zählersollwerte müssen vor Eintreffen des Startsignals an die Elektronik geschickt werden. Damit evtl. auftretende Timingfehler zuverlässig erkannt werden können, muß nach dem Sollwert setzen der *dynamische* Status ausgewertet werden.

4.4 Einordnung in das Timing

Je nach Einsatzort des Gerätes (UNILAC, SIS, ESR, ...) sind mehrere Betriebsvarianten nötig. Die erste Variante gilt für den Einsatz am UNILAC. Die zweite Variante für Geräte, die am SIS betrieben werden.

Ein Beispiel eines UNILAC Standardzyklus mit relevanten Events für den Rahmenpulsgenerator ist in Abbildung 4.4 angegeben.

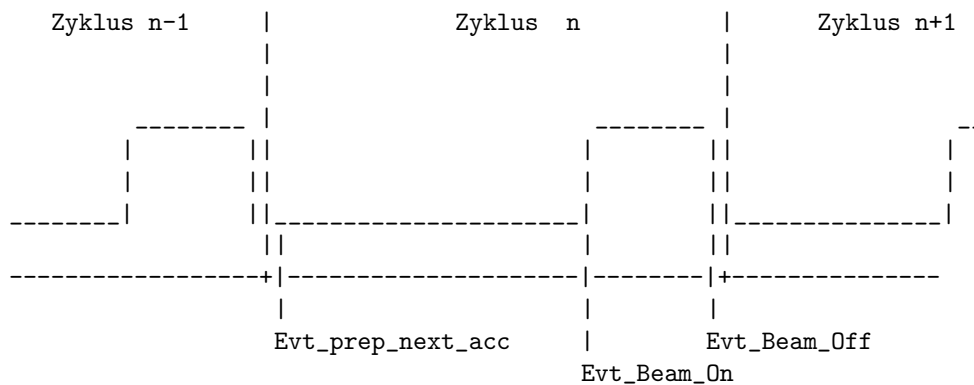


Abbildung 2: Standardzyklus UNILAC mit relevanten Events für Rahmenpulsgenerator

Die Konnektierungen sind in der folgenden Tabelle zusammengefaßt.

Aktion	Event
Sollwerte für die Zähler setzen	Evt_prep_next_acc
dynamischen Status lesen und auswerten	Evt_Beam_Off

Tabelle 4: Standard-Eventkonnektierungen für Rahmenpulsgenerator am UNILAC

Ein Beispiel eines SIS Standardzyklus mit relevanten Events für den Rahmenpulsgenerator ist in Abbildung 4.4 angegeben.

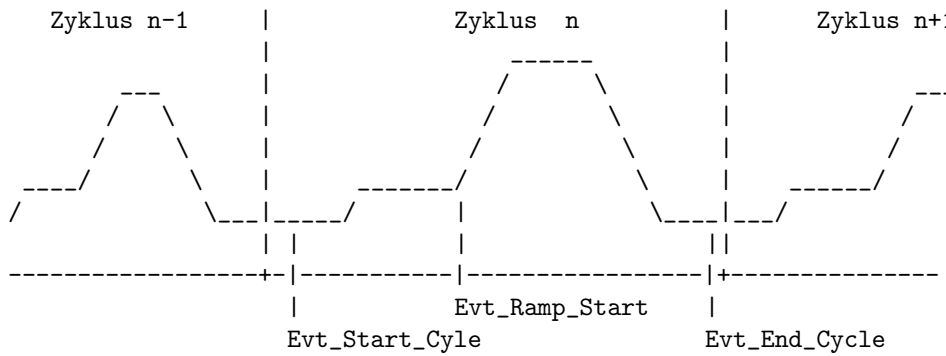


Abbildung 3: Standardzyklus SIS mit relevanten Events für Rahmenpulsgenerator

Die Konnektierungen sind in der folgenden Tabelle zusammengefaßt.

Aktion	Event
Sollwerte für die Zähler setzen	Evt_Start_Cycle
Zähler per Funktionscode starten	Evt_Ramp_Start
dynamischen Status lesen und auswerten	Evt_End_Cycle

Tabelle 5: Standard-Eventkonnektierungen für Rahmenpulsgenerator SIS

4.5 Festlegung von Startwerten

4.5.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset durchgeführt.
- Alle Sollwerte werden für alle virtuellen Beschleuniger auf 0 gesetzt und am Gerät eingestellt.
- Die SE wird in den Eventmode-Betrieb geschaltet (nur bei Kaltstart der SE).
- Die Standard-Eventkonnektierungen werden der Betriebsvariante entsprechend gesetzt.

4.5.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset durchgeführt.

4.6 Handbetrieb

Ein Handbetrieb des Gerätes ist nicht vorgesehen.

4.7 Verhalten bei Störungen

4.7.1 Event-Sequenzfehler

Kommen die relevanten Events nicht in der richtigen Reihenfolge (Sollwert setzen, Status lesen), so ist der aktuelle Zyklus abzubrechen und ein Fehler zu melden.

4.7.2 Event-Overrun

Overrun-Fehler müssen nicht berücksichtigt werden, da evtl. daraus resultierende Timingfehler von der Gerätehardware selbständig erkannt und im dynamischen Status angezeigt werden.

4.7.3 Ausfall der Kommunikation EC – Gerät

Der Ausfall der Kommunikation zwischen EC und Gerät führt zu Timeouts. Tritt ein Timeout auf, so ist der aktuelle Zyklus mit einem entsprechenden Fehler abzubrechen.

4.8 Bedienungsfehler vom Operating

Die Sollwerte für die Zähler, die vom Operating gesetzt werden, müssen auf ihre Plausibilität hin überprüft werden. Z. B. Sollwert auf Taktfrequenz der Zähler normieren und überprüfen, ob das Ergebnis in ein 32 Bit Register paßt.

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 3.6 definierten logischen Gerätes.

5.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **RPG**.

Die Gerätemodellnummer ist 27_{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

Master-Properties							
Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
CONSTANT	R	0	–	1	BitSet32	1	0
SWITCH	R/W	1	BitSet16	1	BitSet16	1	0

5.2.1 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: Beim Lesen immer 1, d.h. 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 3.7 auf Seite 15 und in der Tabelle 3.7 auf Seite 15 erklärt sind.

5.2.3 INIT

Bedeutung: Initialisierung des Gerätes (Kaltstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt 4.5.1 auf Seite 17.

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 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 einiger gerätespezifischer Konstanten.

Parameter: Keine.

Daten: Das bitSet32 Datum gibt die Frequenz an (in Hz), mit der die Zählerbausteine betrieben werden.

5.2.8 SWITCH

Bedeutung: Setzen bzw. Lesen der Kanalanwahl an den Multiplexern. Achtung! Diese Property existiert nur in der Software-Variante *SIS_DIAG*.

Parameter: Der 16-Bit Integer Wert gibt die Nummer des Multiplexers an. 0 bedeutet Multiplexer A und 1 bedeutet Multiplexer B.

Daten: Der 16-Bit Integer Wert gibt die Kanalnummer am Multiplexer an. Folgende Kombinationen aus Parameter und Daten sind möglich:

Parameter	Datum	Kanal	Signal
0	1	A,1	S07DP3
0	2	A,2	Mischer
0	3	A,3	Schottky
0	4	A,4	AUX 4
0	5	A,5	AUX 5
0	6	A,6	AUX 6
1	1	B,1	S02BE1
1	2	B,2	S08BE2
1	3	B,3	HF 3
1	4	B,4	HF 4
1	5	B,5	HF 5
1	6	B,6	HF 6

5.3 Die Slave-Properties

Slave-Properties							
Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
ACTIV	R/W	0	–	1	BitSet16	1	0
COPYSET	W	0	–	1	BitSet16	1	0
EQMERROR	RA	0	–	139	Integer32	1	0
DYNSTAT	RA	0	–	2	BitSet16	1	0
COUNTVAL	RA/WA	0	–	2	RealF	-6	s
WORKMODE	R/W	0	–	1	BitSet16	1	0
TRIGDESC	RA/WA	0	–	6	Integer16	1	0
OUTCHAN	RA/WA	0	–	8	Integer16	1	0
TIFDESC	RA/WA	0	–	4	BitSet16	1	0
OUTACTIV	R/W	1	BitSet16	1	Integer16	1	0
OUTTYPE	R/W	1	BitSet16	1	Integer16	1	0
BURST	RA/WA	0	–	2	RealF	-6	s
GPLENGTH	R	0	–	1	Integer32	-6	s

5.3.1 ACTIV

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

Parameter: Keine.

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

5.3.2 EQMERROR

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

Parameter: 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 DYNSTAT

Bedeutung: Lesen des *dynamischen* Gerätestatus.

Parameter: Keine.

Daten: Das erste Datum ist der *dynamische* Status am Beginn des Zyklus (aber nach dem Laden der Zählersollwerte). Das zweite Datum enthält den *dynamischen* Status am Ende des Zyklus. Zur Bedeutung der einzelnen Bits des *dynamischen* Status siehe Abschnitt 2 auf Seite 12.

5.3.5 COUNTVAL

Bedeutung: Setzen bzw. Lesen der Zählersollwerte.

Parameter: Keine.

Daten: Das erste Datum ist der Sollwert des Startzählers, das zweite Datum entspricht dem Sollwert des Stopzählers. Die Einheit ist μs .

5.3.6 WORKMODE

Bedeutung: Setzen bzw. Lesen des *working-mode*.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Null heißt, das Gerät arbeitet im *independent*-Mode bzw. soll im *independent*-Mode arbeiten. Eins heißt, das Gerät arbeitet im *transparent*-Mode bzw. soll im *transparent*-Mode arbeiten.

5.3.7 TRIGDESC

Bedeutung: Setzen bzw. Lesen der Triggerbedingungen.

Parameter: Keine.

Daten: Die bis zu 6 Daten haben folgende Bedeutung:

1: Repeat-Mode

0: *single*

1: *repeat*

2: Trigger-Mode

0: *kein Trigger*

1: *Externer Trigger*

2: *Interner Trigger*

3: *Event-Trigger*

3 ... 6: Triggerbeschreibung für *externe*- und *Event*-Triggerung:

Externer Trigger:

3: Nummer (1 ... 8) des Triggerkanaleingangs. Positiver Wert bedeutet *positive* Logik (Trigger mit steigender Flanke), negativer Wert bedeutet *negative* Logik (Trigger mit fallender Flanke).

Event-Trigger:

3: zu dekodierender Eventcode als *Start-Event*

4: Bitbreite mit der das *Start-Event* dekodiert werden soll

Wert	Bedeutung
0	16-Bit-Dekodierung
1	12-Bit-Dekodierung
2	8-Bit-Dekodierung

5: zu dekodierender Eventcode als *Stop-Event*

6: Bitbreite mit der das *Stop-Event* dekodiert werden soll

Wert	Bedeutung
0	16-Bit-Dekodierung
1	12-Bit-Dekodierung
2	8-Bit-Dekodierung

5.3.8 OUTCHAN

Bedeutung: Setzen bzw. Lesen der Ausgangskanal-Auswahl.

Parameter: Keine.

Daten: Die 1 bis 8 Daten geben die zu versorgenden Ausgangskanalnummern (1 ... 8) an. Negative Werte bedeuten *low-active* Ausgangspegel, Positive Werte bedeuten *high-active* Ausgangspegel.

5.3.9 TIFDESC

Bedeutung: Setzen bzw. Lesen der TIF-Programmierung.

Parameter: Keine.

Daten: Die 4 BitSet16-Werte haben folgende Bedeutung:

1. Zu dekodierender Eventcode auf TIF-1.
2. Bitbreite mit der an TIF-1 dekodiert werden soll.

Wert	Bedeutung
0	16-Bit-Dekodierung
1	12-Bit-Dekodierung
2	8-Bit-Dekodierung

3. Zu dekodierender Eventcode auf TIF-2.
4. Bitbreite mit der an TIF-2 dekodiert werden soll.

Wert	Bedeutung
0	16-Bit-Dekodierung
1	12-Bit-Dekodierung
2	8-Bit-Dekodierung

5.3.10 OUTACTIV

Bedeutung: Setzen bzw. Lesen der Einstellung eines Ausgangskanals.

Parameter: 1 BitSet16-Wert gibt die Ausgangskanal-Nummer an.

Daten: Der Integer16-Wert gibt die Einstellung des Ausgangskanals an:

Wert	Bedeutung
-1	enabled, <i>active low</i>
0	disabled
1	enabled, <i>active high</i>

5.3.11 OUTTYPE

Bedeutung: Setzen bzw. Lesen der Belegung eines Ausgangskanals.

Parameter: 1 BitSet16-Wert gibt die Ausgangskanal-Nummer an.

Daten: Der Integer16-Wert gibt die Belegung des Ausgangskanals an:

Wert	Bedeutung
0	Gate-Puls
1	Start-Event (nur wenn Triggerart auf Event steht)
2	Stop-Event (nur wenn Triggerart auf Event steht)
3	TIF1-Event (aktiviert Pulse-2)
4	TIF2-Event (deaktiviert Pulse-2)
5	A-Event (aktiviert Pulse-3)
6	B-Event (deaktiviert Pulse-3)
7	C-Event
9	Pulse-1 (von Start- bis Stop-Event)
10	Pulse-2 (von TIF1- bis TIF2-Event)
11	Pulse-3 (von A- bis B-Event)
13	Burstmode (nur während des Gate-Pulses)

5.3.12 BURST

Bedeutung: Setzen bzw. Lesen der Parameter des Burstmode. Im Burstmode wechselt das Signal am Ausgang während des Gatepulses abwechselnd von *high* nach *low* und umgekehrt. Die Dauer sowohl der high- als auch der low-Phase ist in μs einstellbar.

Parameter: Keine.

Daten: Die beiden Real-Werte haben folgenden Bedeutung:

1. Länge des *high*-Signals in μs .
2. Länge des *low*-Signals in μs .

5.3.13 GPLLENGTH

Bedeutung: Lesen der zuletzt gemessenen Gatepulslänge.

Parameter: Keine.

Daten: Gatepulslänge in μs .

Teil II

Der Entwurf der Software

6 Softwareentwurf

Die Software zur Bedienung und Überwachung des Rahmenpulsgenerators erfordert keine Besonderheiten in der Programmierung, die an dieser Stelle einer expliziten Erklärung bedürften. Verwendung finden ausschließlich Standardmethoden, die vom Kontrollsystem zur Verfügung gestellt werden.

7 Lokale Datenbasis

In der lokalen Datenbasis sind (außer den Standards) keine zusätzlichen Einträge nötig.

8 Dualport RAM

Hierher kommt die Beschreibung des gerätespezifischen Teils des Dualport RAM (`m_data_type`, `s_data_type`, `Dev_Common_Buf_Type`, ...).

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 Rahmenpulsgenerators folgende gerätespezifischen USRs benötigt:

9.2.1 `R_Constant`

Lesen der Gerätespezifischen Konstanten (Taktfrequenz, Betriebsmodus und Triggerart)

9.2.2 W_Countval

Umrechnung der Zählersollwerte in μs auf 32 Bit Zählersetzwerte normiert auf die Taktfrequenz der Zähler.

9.2.3 R_Countval

Umrechnung der Zählersetzwerte (normiert auf die Taktfrequenz der Zähler) auf Zählersollwerte in μs .

9.2.4 R_DynStat

Lesen des *dynamischen* Gerätestatus.

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

Erläuterung, welche Übergänge zwischen den internen Zuständen vorgesehen sind und wodurch sie ausgelöst werden sollen.

Die Zustände und die Übergänge zwischen denselben sind in den folgenden Tabellen zusammengefaßt. Ebenso befindet sich eine Legende der Tabellen weiter unten.

10.1.3 Standard-Zustandsübergänge

Das Gerät nimmt an der Puls-zu-Puls-Modulation teil. Innerhalb eines virtuellen Beschleunigers wird zum Sollwert setzen der Zustand **ready** oder **error** erwartet und zum Zustand **busy** weitergeschaltet. Zum Istwert lesen wird der Zustand **busy** erwartet und zum Zustand **ready** weitergeschaltet.

ready (oder error) -> busy -> ready

10.2 Eventkonnektierte EQMs

10.2.1 Prep_EQM

Event: Evt_prep_next_acc (UNILAC) oder Evt_Start_Cycle (SIS, ESR)

Aktion: Sollwerte an die Zähler schicken. Zählerzustand ermitteln (über *dynamischen* Status), um erkennen zu können, ob evtl. die Zähler schon frühzeitig losgelaufen sind.

Tabelle der Zustandsübergänge				
von↓ nach→		ready	error	busy
ready	U:	–	overrun etc.	Zyklus Start
	B:	–	–	–
	A:	–	div. EQMs	Prep_EQM
error	U:	RESET, Zyklus Ende	–	Zyklus Start
	B:	–	–	–
	A:	RESET_EQM End_EQM	–	Prep_EQM
busy	U:	Zyklus Ende	–	–
	B:	–	–	–
	A:	End_EQM	–	–

Tabelle 9: Zustandsübergangsdiagramm

Legende

- Die Priorität der Zustände (höchste Priorität zuerst): error, ready, busy.
Liegen mehrere Bedingungen für verschiedene Zustände gleichzeitig vor (z.B. Netz aus und Gerät auf Handbetrieb), muß der jeweils wichtigste Zustand eingenommen werden.
- U: Auslösende Ursache.
RESET Reset wird per Kommando oder Knöpfchendrücken ausgelöst.
- B: Abzuprüfende Bedingung.
R Remotebit des Status steht auf Remote.
r Remotebit des Status steht auf Local.
P Powerbit des Status steht auf Power on.
p Powerbit des Status steht auf Power off.
- A: Ausführende Stelle des Zustandsübergangs.
Status lesen (period.) Beim periodischen (oder zumindest regelmäßigen) Lesen des Status.
..._EQM Innerhalb des EQMs ..._EQM.

Tabelle 10: Legende zu den Zustandsübergangsdiagrammen

10.2.2 End_EQM

Event: Evt_Beam_Off (UNILAC) oder Evt_End_Cycle (SIS, ESR).

Aktion: Auslesen des *dynamischen* Status und Untersuchung desselben auf Timingfehler.

10.2.3 Start_EQM

Event: Evt_Ramp_Start (SIS, ESR).

Aktion: Nur bei Triggerart *intern!*

Zählerzustand feststellen und auswerten (über *dynamischen* Status, um evtl. Timingfehler zu erkennen). Zähler per Funktionscode starten.

10.3 Periodisch konnektierte EQMs

10.3.1 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.

10.4 An externe Interrupts konnektierte EQMs

10.4.1 Interlock_EQM

Interrupt: Summen-Interlock.

Aktion: Keine Aktion ausführen.

10.4.2 DRD_EQM

Interrupt: Data Ready Interrupt.

Aktion: Keine Aktion ausführen.

10.4.3 DRQ_EQM

Interrupt: Data Request Interrupt.

Aktion: Keine Aktion ausführen.

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 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 16) 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

Lesen des Gerätestatus und Aktualisierung desselben im DPRAM.

10.8.2 Do_Intr_Service_Prep

Keine Aktion.

11 Varianten

Die Varianten im Einzelnen:

Variant 1 (RPG-UNI.PIN) Die EQMs sind an das UNILAC-Timing konnektiert.

Variant 2 (RPG-SIS.PIN) Die EQMs sind an das SIS-Timing konnektiert.

12 Besonderheiten

Bisher sind keine Besonderheiten (Abweichungen von Standardverfahren der Gerätebehandlung durch das Kontrollsystem) zu vermerken.

Index

—A—

Abriß 2
Active_EQM 26
Änderungsprotokoll 2
An externe Interrupts konnektierte EQMs 25
Aufgabe des Gerätes 7

—B—

Bedienung des Gerätes 14
Bedienungsfehler 16
Besonderheiten 27

—D—

Datenbasis 23
Dev_Init_EQM 26
Dev_Reset_EQM 26
Display_DevErr_EQM 26
Display_DPR_EQM 26
DRD Interrupt 13
DRD_EQM 26
DRQ Interrupt 12
DRQ_EQM 26
Dualport RAM 23
Dynamischer Status 14

—E—

End_EQM 25
EQMs 24
EQMs; An externe Interrupts konnektierte 25
EQMs; An externe Interrupts konnektier-
te; DRD_EQM 26
EQMs; An externe Interrupts konnektier-
te; DRQ_EQM 26
EQMs; An externe Interrupts konnektier-
te; Interlock_EQM 25
EQMs; Eventkonnektierte 25
EQMs; Eventkonnektierte; End_EQM 25
EQMs; Eventkonnektierte; Prep_EQM 25
EQMs; Eventkonnektierte; Start_EQM 25
EQMs; für die Diagnose vor Ort 26
EQMs; für die Diagnose vor Ort; Display_DevErr_EQM
26
EQMs; für die Diagnose vor Ort; Display_DPR_EQM
26
EQMs; Globale Routinen 27

EQMs; Globale Routinen; Do_Intr_Service_Prep
27
EQMs; Globale Routinen; Read_and_Update_Status
27
EQMs; Kommandogetriggerte 26
EQMs; Kommandogetriggerte; Active_EQM
26
EQMs; Kommandogetriggerte; Dev_Init_EQM
26
EQMs; Kommandogetriggerte; Dev_Reset_EQM
26
EQMs; Kommandogetriggerte; Power_EQM
26
EQMs; Kommandogetriggerte; Status_EQM
26
EQMs; Periodisch konnektierte 25
EQMs; Periodisch konnektierte; Update_Config_EQM
25
EQMs; Sonstige 26
EQMs; Sonstige; Startup_EQM 26
Event-Overrun 16
Event-Sequenzfehler 16
Eventkonnektierte EQMs 25
Eventkonnektierungen 14

—F—

Funktionen 8
Funktionscodes 10

—G—

Genauigkeitsanforderungen 14
Gerät; Aufgabe 7
Gerät; Bedienung 14
Gerät; Funktionen 8
Gerät; Funktionen; Data-Ready-Interrupt 8
Gerät; Funktionen; Entprellung 8
Gerät; Funktionen; Independent-Mode 8
Gerät; Funktionen; Repeat-Mode 8
Gerät; Funktionen; Single-Mode 8
Gerät; Funktionen; Transparent-Mode 8
Gerät; Funktionen; Trigger-Moden 9
Gerät; Funktionen; Triggerarten 8
Gerät; Funktionen; Triggerarten; Event 8
Gerät; Funktionen; Triggerarten; Extern 8
Gerät; Funktionen; Triggerarten; Software 8
Gerät; Hardware 7
Gerät; logisches 13

Gerät;Repräsentation 16
Gerät;Schnittstelle 10
Gerät;Schnittstelle;Subadressen 11
Gerät;Technische Daten 7
Gerätemodell 7
Gerätemodell;Kennzeichnung 16
Gerätemodell;Master-Properties 16
Gerätemodell;Slave-Properties 19
Globale Routinen 27

—H—

Handbetrieb 15
Hardware des Gerätes 7
Hardware;Modulbus 10
Hardwarestatus 13

—I—

Init 15
Interfacekarte 10
Interlock 12
Interlock_EQM 25
Interne Zustände 24
Interrupt;DRD Interrupt 13
Interrupt;DRQ Interrupt 12
Interrupt;Interlock 12

—K—

Kaltstarts 15
Kommandogetriggerte EQMs 26
Konfigurationsabfrage 13

—L—

logisches Gerät 13
Lokale Datenbasis 23

—M—

Master-Properties 16
Modulbus 10
Modulbus;EPLD-ID 10
Modulbus;Modul-Adresse 10
Modulbus;Modul-ID 10
Modulbus;Modulskalierung 10
Modulbus;Standard-Sub-Adressen 10

—N—

N_Init 23
N_Reset 23
Normalbetrieb 14

—O—

Overrun 16

—P—

Periodisch konnektierte EQMs 25
Power_EQM 26
Prep_EQM 25
Properties;ACTIV 19
Properties;CONSTANT 19
Properties;COPYSET 20
Properties;COUNTVAL 21
Properties;DYNSTAT 20
Properties;EQMERROR 19
Properties;INFOSTAT 17
Properties;INIT 17
Properties;Master- 16
Properties;OUTCHAN 22
Properties;POWER 16
Properties;RESET 17
Properties;Slave- 19
Properties;STATUS 17
Properties;TIFDESC 22
Properties;TRIGDESC 21
Properties;VERSION 17
Properties;WORKMODE 21

—R—

R_Active 23
R_Constant 23
R_Countval 24
R_DynStat 24
R_EQMErr 23
R_InfoStat 23
R_Power 23
R_Status 23
R_Version 23
Repräsentation des Gerätes 16
Reset 15

—S—

Schnittstelle zum Gerät 10
Sequenzfehler 16
Slave-Properties 19

Softwareentwurf 23
Softwarestatus 13
Sollwerte 14
Sonstige EQMs 26
Standardzyklus SIS 14
Standardzyklus UNILAC 14
Start_EQM 25
Startup_EQM 26
Startwerte 15
Status_EQM 26
Statusbits 13
Störungen 16
Störungen_;Event-Overrun 16
Störungen_;Event-Sequenzfehler 16
Störungen_;Kommunikation EC – Gerät 16

—T—

Technische Daten 7
Timing 14
Timingfehler 14

—U—

Update_Config_EQM 25
USRs 23
USRs_;gerätespezifische 23
USRs_;gerätespezifische_;R_Constant 23
USRs_;gerätespezifische_;R_Countval 24
USRs_;gerätespezifische_;R_DynStat 24
USRs_;gerätespezifische_;W_Countval 24
USRs_;obligatorische 23
USRs_;obligatorische_;N_Init 23
USRs_;obligatorische_;N_Reset 23
USRs_;obligatorische_;R_Active 23
USRs_;obligatorische_;R_EQMErr 23
USRs_;obligatorische_;R_InfoStat 23
USRs_;obligatorische_;R_Power 23
USRs_;obligatorische_;R_Status 23
USRs_;obligatorische_;R_Version 23
USRs_;obligatorische_;W_Active 23
USRs_;obligatorische_;W_CopySet 23
USRs_;obligatorische_;W_Power 23

—V—

Varianten 27
Varianten_;Betriebs- 14
Varianten_;Software- 27

—W—

W_Active 23
W_CopySet 23
W_Countval 24
W_Power 23
Warmstarts 15

—Z—

Zeitkritische Anforderungen 14
Zustände_;Interne 24
Zustände_;Interne_;Bedeutung 24
Zustände_;Interne_;Standard-Übergänge 24
Zustände_;Interne_;Übergänge 24