

MFU FSP233

Interlocktexte programmieren

Adaptive Control Unit

ACU

Derek Schupp

Version vom: Mittwoch, 20. Januar 2021, 09:05:00

Aktuelle Versionen immer unter:

\\Winfilesvb\EET\$Root\EETCommon\ACU_System_Dokumentation

Inhaltsverzeichnis

1. Änderungsliste	1
2. Abkürzungen	2
3. Interlocktexte Textdatei	3
3.1. Struktureller Aufbau	3
3.1.1. Der Header	3
3.1.2. Die Daten	3
3.1.2.1. Einleitender Teil	3
3.1.2.2. Eigentliche Interlocktexte	3
3.1.3. Der Abschluss	4
3.1.3.1. Das Carriage Return und Line Feed	4
3.2. Aufbau der Textdatei	4
3.2.1. Aufbau der Textdatei für die Kommunikation über FSP233	4
3.2.1.1. Programmierung über FSP233	4
3.2.1.2. Lesen über FSP233	5
3.2.2. Aufbau der Textdatei für die Programmierung mittels USB Speicher	6
3.3. Die von PCA erstellten Interlocktexte	6
3.4. Identifikation baugleicher Module	7

1. Änderungsliste

<i>Datum</i>	<i>Name</i>	<i>Kommentar</i>
25.09.2008	D. Schupp	Dokument erstellt
17.11.2008	D. Schupp	Dokument umbenannt und Interlocktextlänge auf 25 Zeichen korrigiert.
10.11.2009	D. Schupp	'command' Byte ersatzlos gestrichen, weil obsolet
12.11.2009	D. Schupp	ACU in MFU und APC ² in ACU umbenannt
09.12.2010	D. Schupp	Interlocktexte sind nun im FP233 zu finden, die „Stringvariante“ zum Programmieren der Interlocktexte entfällt ersatzlos.
19.06.2013	D. Schupp	Die optionalen <CR> <LF> sind entfallen und nun zwingend erforderlich
16.09.2015	D. Schupp	Druckversion
08.02.2019	D. Schupp	Vollständig überarbeitet. Unterschied zwischen FSP233 und USB Speicher Programmierung.
20.01.2021	D. Schupp	Umstieg auf Office 2016, Schrift „Universe“ durch „Calibri“ ersetzt und die damit verbundenen Neuformatierungen durchgeführt. Diverse kleine Korrekturen (u.A. Rechtschreibung). Kapitel 3.3 „Die von PCA erstellten Interlocktexte“ hinzu gefügt.

2. Abkürzungen

Um den Text übersichtlich und kurz zu gestalten werden bestimmte Begriffe abgekürzt:

ACU	Adaptive Control Unit - alle zur Steuer- und Kontrolleinheit eines Netzgerätes gehörenden Komponenten
ADC	Analog Digital Converter
DCCT	DC Current Transformer
FPGA	Field Programmable Gate Array - ein programmierbarer Logikbaustein der die Funktionalitäten eines ACU Moduls beinhaltet
ICM	Interlock-and Control Module
MFU	Multi Function Unit - zentrale Grundeinheit eines ACU-Systems
NIOSII	Mikroprozessor in der MFU der die Benutzerschnittstelle zur Verfügung stellt
FW	Firmware - die notwendige Programmierung des FPGA. Es wird unterschieden zwischen Application und Factory Image. Das Application Image ist i.d.R. durch den Endanwender ersetzbar. Das Factory Image hingegen nur durch das Support-Team.
SW	Software - Betriebssoftware der NIOSII/VNC2 Mikroprozessoren innerhalb der MFU. Auch hier wird unterschieden zwischen Application und Factory Image. Das Application Image ist i.d.R. durch den Endanwender ersetzbar. Das Factory Image hingegen nur durch das Support-Team.
SVE	Strom-Versorgungs-Einheit (Netzgerät)
USI	Universal Serial Interface
PCA	Power Config Advanced (Software)
PPS	Personal Protection System
Quench	Sudden transition from superconducting to normal conditions

3. Interlocktexte Textdatei

Die Interlocktexte können mittels PowerConfigAdvanced oder als einfache Textdatei generiert und in die MFU des ACU Systems geladen werden.

Dabei ist zu unterscheiden, ob eine Interlocktexte Textdatei über den FSP233 der MFU oder einen USB Speicher programmiert werden soll.

3.1. Struktureller Aufbau

3.1.1. Der Header

Eingeleitet wird die Interlock Textdatei von einem Header, der folgendermaßen aufgebaut ist:

STX	PID	PID	GW	MA	FSP	FSP	[DATA...]	[PP PP]	ETX
--- Header ---								- Prüfsumme -	

Tabelle 3.1

STX	[1 Byte]	Start Of Text	immer	0x02
PID	[2 Bytes]	Lesen oder Schreiben	USI_PID_WR	0x5752
			USI_PID_RD	0x5244
GW	[1 Byte]	Gatewayadresse	immer	0x30
MA	[1 Byte]	Moduladresse	immer	0x30
FSP	[2 Byte]	FIFO Start Punkt	immer	0x4539
[DATA]	[n Bytes]	Interlocktexte	nur beim Schreiben	
[PP PP]	[2 Bytes]	Prüfsumme	nur beim Schreiben	
ETX	[1 Byte]	End Of Text		
[]	nur bei PID = USI_PID_WR			

3.1.2. Die Daten

Innerhalb der Daten befindet sich für jedes zu beschreibende Modul ein eigener Abschnitt. Jeder dieser Abschnitte startet mit einem einleitenden Teil, gefolgt von den eigentlichen Interlocktexten. Die Interlocktexte sind dabei aufsteigend nummeriert.

3.1.2.1. Einleitender Teil

Der einleitende Teil gibt die USI Nummer, die Modul Nummer und die Anzahl der nachfolgend beschriebenen Interlocks (max. 255) an. Gefolgt von 6 Dummies, die alle „0“ sind. Abgeschlossen wird der einleitende Teil von einem optionalen „Carriage Return“ und einem verbindlichen „Line Feed“.

USINr.	[1 Byte]
ModulNr.	[1 Byte]
AnzahlInterlocks	[2 Bytes]
DummyByte	[6 Bytes] für evtl. Erweiterungen
(CR)/LF	[1 Byte, bzw. 2 Bytes]

Bis hier sind es folglich 11, bzw. 12 Bytes.

Zu beachten ist, dass die Zählung der USIs und Module hexadezimal erfolgt und dabei von 0x1 bis 0xB (USI 1..11), bzw. 0x1 bis 0x8 (Modul 1 bis 8) gezählt wird.

3.1.2.2. Eigentliche Interlocktexte

Die eigentlichen Interlocktexte schließen sich an den einleitenden Teil an und beschreiben ausschließlich die Interlocks des Moduls, welches auch im einleitenden Teil genannt wird.

Jeder Eintrag der eigentlichen Interlocktexte beginnt dabei mit der Interlocknummer, deren aufsteigende Zählung mit „01“ startet. Direkt im Anschluss folgen maximal 50 Bytes Klartext zur Beschreibung

des Interlocks. Dieser Text wird später auf dem TFT der MFU angezeigt. Jeder Eintrag wird mit einem optionalen „Carriage Return“ und einem verbindlichen „Line Feed“ abgeschlossen.

```
InterlockNr.      [ 2 Byte]
InterlockText     [50 Bytes] maximale Anzahl, kann auch kürzer sein
                  (CR)/LF      [ 1 Byte, bzw. 2 Bytes]
```

Jeder Eintrag hat also eine maximale Länge von 53, bzw. 54 Bytes.

Zu beachten ist, dass die Nummerierung der Interlocks mit „01“ beginnt und auch hier Hexadezimal erfolgt.

3.1.3. *Der Abschluss*

Das Ende der Textdatei bildet eine Prüfsumme über alle Datenbytes und das ETX (0x03) um der MFU eindeutig mitzuteilen, dass nun das Ende der Übertragung erreicht ist.

3.1.3.1. *Das Carriage Return und Line Feed*

Jeder einleitende Teil und jeder Interlocktexteintrag muss zumindest mit einem Line Feed abgeschlossen werden. Wird die Interlocktextdatei auf einem Windowsrechner editiert, setzt Windows bei einem Zeilenumbruch automatisch ein „Carriage Return“ und ein „Line Feed“ ans Ende der Zeile. Linuxsysteme hingegen fügen lediglich ein „Line Feed“ ein.

3.2. *Aufbau der Textdatei*

Der unterschiedliche Aufbau der ist abhängig davon über welchen Weg die Daten in die MFU geschrieben, bzw. aus dieser gelesen werden.

Über das FSP233 ist es nötig zusätzlich Informationen in die Daten zu packen, im Hinblick auf das Ziel der übertragenen Daten und deren Prüfsumme. Das Gleiche gilt beim Auslesen über FSP233. Hierbei werden Informationen über den Ursprung der Daten hinzugefügt und ebenfalls eine Prüfsumme.

Werden die Daten mittels USB Speicher übertragen, handelt es sich hingegen nur um die reinen Daten, also den wirklichen Inhalt des Flash-Speichers de MFU.

3.2.1. *Aufbau der Textdatei für die Kommunikation über FSP233*

3.2.1.1. *Programmierung über FSP233*

Soll die Interlocktexte Datei über den FSP 233 in die MFU geschrieben werden, ist diese nach dem nachfolgenden Muster aufzubauen.

Wird die Datei von PowerConfigAdvanced (PCA) erzeugt und in die MFU gespeichert, geschieht dies nach genau diesem Schema.

```
WR00E91118000000 [(CR) LF]
01Analog.Interlock.1..... [(CR) LF]
02Analog.Interlock.2..... [(CR) LF]
03Analog.Interlock.3..... [(CR) LF]
04Analog.Interlock.4..... [(CR) LF]
(...) 15 weitere Einträge
14Electric.Interlock.10..... [(CR) LF]
15not.used.number.21..... [(CR) LF]
16not.used.number.22..... [(CR) LF]
17not.used.number.23..... [(CR) LF]
18not.used.number.24..... [(CR) LF]
2118000000 [(CR) LF]
01InterlockNummer00001..... [(CR) LF]
02InterlockNummer00002..... [(CR) LF]
03InterlockNummer00003..... [(CR) LF]
04InterlockNummer00004..... [(CR) LF]
```

```
(...) 15 weitere Einträge
14InterlockNummer00020 ..... [ (CR) LF]
15InterlockNummer00021 ..... [ (CR) LF]
16InterlockNummer00022 ..... [ (CR) LF]
17InterlockNummer00023 ..... [ (CR) LF]
18InterlockNummer00024 ..... [ (CR) LF]
PP□
```

Header, inkl. STX (0x02)

USI Nummer, Modul Nummer, mit insgesamt (im Beispiel) 24 (Hex. 18) Interlocktexten

6 Dummybytes (z. Zt. immer 0)

ggf. 'Carrige Return' (Windows) und 'LineFeed' (Windows/Linux) müssen sein

2 Byte Interlocknummer, beginnend mit 0x01

50 Zeichen Interlocktext. Nicht genutzte Textstellen müssen NICHT mit Leerzeichen gefüllt sein, die Zeile muss aber mit [CRLF] abgeschlossen werden und der Interlocktext inkl. Interlocknummer darf 52 Zeichen NICHT überschreiten

Diese Leerzeichen sind optional und füllen den Text auf die zulässigen 50 Zeichen Interlocktext bevor das [CRLF] erfolgt. Das [CRLF] kann auch unmittelbar nach dem eigentlichen Text erfolgen, die MFU fügt die notwendigen Leerzeichen dann automatisch ein

Prüfsumme PP (2 Byte ASCII) und ETX (0x03)

3.2.1.2. Lesen über FSP233

Mit einem Lesekommando (siehe 3.1.1) über ein Terminalprogramm kann FSP233 gelesen werden.

Wird die Datei über den FSP233 gelesen sehen die zuvor unter Kapitel 3.2.1.1 über FSP233 programmierten Daten wie folgt aus:

```
□00E9111800000001Analog.Interlock.1.....02A
analog.Interlock.2.....03Analog.Interlock.3.
.....04Analog.Interlock.4.....
..... (...) 15 weitere Einträge 14Electric.Interlock.10.....
.....15not.used.number.21.....16not.
used.number.22.....17not.used.number.23...
.....18not.used.number.24.....
.....211800000001InterlockNummer00001.....
..02InterlockNummer00002.....03InterlockNumme
r00003.....04InterlockNummer00004.....
..... (...) 15 weitere Einträge 14InterlockNummer00020.....
.....15InterlockNummer00021.....1
6InterlockNummer00022.....17InterlockNummer00
023.....18InterlockNummer00024.....
..... PP□
```

Header, inkl. STX (0x02)

USI Nummer, Modul Nummer, mit insgesamt (im Beispiel) 24 (Hex. 18) Interlocktexten

6 Dummybytes (z. Zt. immer 0)

2 Byte Interlocknummer, beginnend mit 0x01

50 Zeichen Interlocktext. Nicht genutzte Textstellen sind mit Leerzeichen gefüllt. Jeder Interlocktext inkl. Interlocknummer ist also 52 Zeichen lang.

Diese Leerzeichen wurden ggf. automatisch von der MFU beim Programmieren der Interlocktexte aufgefüllt.

Prüfsumme PP (2 Byte ASCII) und ETX (0x03)

3.2.2. Aufbau der Textdatei für die Programmierung mittels USB Speicher

Der Aufbau der Textdatei entspricht dem Muster von Kapitel 3.2.1 mit ein paar Änderungen.

Die Datei beinhaltet am Anfang KEINEN Header, am Ende keine Prüfsumme und KEIN ETX und ebenfalls KEIN Carriage Return und KEIN Line Feed.

Im Weiteren müssen die 50 Zeichen des eigentlichen Interlocktexteintrags vollständig sein, d.h. nicht benutzte Zeichen am Ende des Texteintrags müssen mit Leerzeichen aufgefüllt werden.

```
111800000001Analog.Interlock.1.....02Analog
.Interlock.2.....03Analog.Interlock.3.....
.....04Analog.Interlock.4.....
.....(...)15 weitere Einträge 14Electric.Interlock.10.....
.....15not.used.number.21.....16not.used.n
umber.22.....17not.used.number.23.....
.....18not.used.number.24.....
....211800000001InterlockNummer00001.....02In
terlockNummer00002.....03InterlockNummer00003
.....04InterlockNummer00004.....
.....(...)15 weitere Einträge 14InterlockNummer00020.....
.....15InterlockNummer00021.....16Interl
ockNummer00022.....17InterlockNummer00023....
.....18InterlockNummer00024.....
.....
.....
```

USI Nummer, Modul Nummer, mit insgesamt (im Beispiel) 24 (Hex. 18) Interlocktexten

6 Dummybytes (z. Zt. immer 0)

2 Byte Interlocknummer, beginnend mit 0x01

50 Zeichen Interlocktext. Nicht genutzte Textstellen müssen mit Leerzeichen gefüllt sein. Jeder Interlocktext inkl. Interlocknummer muss 52 Zeichen lang sein.

Diese Leerzeichen sind verpflichtend und füllen den Text auf die zulässigen 50 Zeichen Interlocktext.

3.3. Die von PCA erstellten Interlocktexte

Werden Interlocktexte mittels PCA erstellt, werden der eigentlichen Interlockbeschreibung optional noch zusätzliche Informationen vorangestellt.

Auf dem TFT der MFU werden, sofern Interlocktexte für ein Modul verfügbar sind, ausschließlich die bis zu 50 Zeichen Interlocktext angezeigt. Dadurch fällt es mitunter etwas schwer herauszufinden, zu welchem Modul ein angezeigter Text gehört.

Daher ergänzt PCA diesen Interlocktext um zusätzliche Informationen, die das Lesen vereinfachen sollen.

Nachteilig ist hierbei, dass diese zusätzlichen Informationen den eigentlichen Interlocktext um die gleiche Anzahl Zeichen kürzt, da der Eintrag auch weiterhin nur 50 Zeichen haben kann.

Ein von PCA erstellter Eintrag sieht dabei z.B. wie folgt aus:

```
01ICM 1100 Analog.Interlock.1.....
```

ICM – dies ist der Modulname Er lässt sich innerhalb PCA ändern. Dafür besitzt das modulbeschreibende Form i.d.R. ein Benutzerfeld „Module name“.

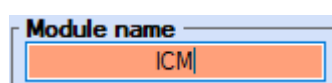


Abbildung 3.1: „Modul name“ Eingabefeld in PCA

Gefolgt von einem Leerzeichen.

Ist das „Module name“ Eingabefeld leer, entfallen sowohl diese Information, als auch das nachfolgende Leerzeichen.

1100 - Gibt die USI-Nummer, Modulnummer und die Interlockbitnummer an.

3.4. Identifikation baugleicher Module

Prinzipiell soll an jede USI jedes beliebige Modul anschließbar sein, die Interlocktexte sind aber bestimmten Modulen an bestimmten USIs zugeordnet.

Ein Modul kann nicht mitteilen für welche Funktionen es verantwortlich ist, z.B. Strom- oder Spannungsmessung, dies muss in den Systemparametern festgelegt werden.

Da auch mehrere Module gleicher Bauart an der ACU angeschlossen sein können. (2xADC usw.), muss klar sein welches Modul zu welchen Interlocktexten gehört.

Dies muss der Benutzer festlegen, ebenso die eindeutigen Konfigurationsparameter.