



FSPs der 40-fach Digitalen Interlock Moduls (ab FW 7.0)

Version vom: Freitag, 17. Juni 2022, 09:29:00

Inhaltsverzeichnis

| | | |
|----|---|----|
| 1. | Änderungsliste | 1 |
| 2. | FSPs 40-fach Interlock Modul | 2 |
| | FSP001_ModuleStatus | 3 |
| | 0x01 _H /1 _D /0x3031 _{ASCII} | |
| | FSP002_ModuleWarnings | 5 |
| | 0x02 _H /2 _D /0x3032 _{ASCII} | |
| | FSP003_ModuleErrors | 6 |
| | 0x03 _H /3 _D /0x3033 _{ASCII} | |
| | FSP004_ModuleInterlocks | 7 |
| | 0x04 _H /4 _D /0x3034 _{ASCII} | |
| | FSP005_InterlocksEnable | 8 |
| | 0x05 _H /5 _D /0x3035 _{ASCII} | |
| | FSP009_ModuleSerialNumber | 9 |
| | 0x09 _H /9 _D /0x3039 _{ASCII} | |
| | FSP010_ModuleCommands | 10 |
| | 0x0A _H /10 _D /0x3041 _{ASCII} | |
| | FSP011_ModuleInterlocksMask_n | 11 |
| | 0x0B _H /11 _D /0x3042 _{ASCII} | |
| | FSP012_USIConfig | 13 |
| | 0x0C _H /12 _D /0x3043 _{ASCII} | |
| | FSP013_PeripheralConfig | 14 |
| | 0x0D _H /13 _D /0x3044 _{ASCII} | |
| | FSP040_RemoteUpdateStatus | 15 |
| | 0x28 _H /40 _D /0x3238 _{ASCII} | |
| | FSP041_RemoteUpdateCommands | 16 |
| | 0x29 _H /41 _D /0x3239 _{ASCII} | |
| | FSP042_RemoteUpdateData | 17 |
| | 0x2A _H /42 _D /0x3241 _{ASCII} | |
| | FSP045_AlteraRemoteUpdateCmd | 19 |
| | 0x2D _H /45 _D /0x3244 _{ASCII} | |
| | FSP046_AlteraRemoteUpdateStatus | 20 |
| | 0x2E _H /46 _D /0x3245 _{ASCII} | |
| | FSP050_ModuleSupplyValues | 21 |
| | 0x32 _H /50 _D /0x3332 _{ASCII} | |
| | FSP053_ModuleTemperatures | 22 |
| | 0x35 _H /53 _D /0x3335 _{ASCII} | |
| | FSP054_ModuleTemperaturesComparisonThresholds | 23 |
| | 0x36 _H /54 _D /0x3336 _{ASCII} | |
| | FSP058_ParameterChecksumValue | 24 |
| | 0x3A _H /58 _D /0x3341 _{ASCII} | |
| | FSP059_ParameterChecksumValueCalculated | 25 |
| | 0x3B _H /59 _D /0x3342 _{ASCII} | |
| | FSP060_ValCounter | 26 |
| | 0x3C _H /60 _D /0x3343 _{ASCII} | |
| | FSP061_InputFilterDelay | 27 |
| | 0x3D _H /61 _D /0x3344 _{ASCII} | |
| | FSP063_WaterFlow_Thresholds | 28 |
| | 0x3F _H /63 _D /0x3346 _{ASCII} | |
| | FSP064_InvertedWaterFlow_TimePeriodeBetweenTwoPulses_in_us | 30 |
| | 0x40 _H /64 _D /0x3430 _{ASCII} | |

1. Änderungsliste

| Datum | Name | Kommentar |
|------------|-----------|---|
| 01.07.2021 | D. Schupp | Dokument erstellt aus ACU-FSP mUSlc TFT |
| 17.06.2022 | D. Schupp | kleinere Korrekturen |
| | | |
| | | |
| | | |
| | | |

2. FSPs 40-fach Interlock Modul

Dieses Dokument behandelt modulspezifische FSPs des 40-fach Interlock Moduls.

| | |
|---------|--|
| Name | FSP001_ModuleStatus |
| Adresse | 0x01_H/1_D/0x3031_{ASCII} |
| Tiefe | 3 Byte / 24 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

[23..21] Reserviert
Reserviert für zukünftige Anwendungen

[20] wenn ,1', Regler freigegeben

[19..16] Status des Moduls

| [3..0] | Status |
|--------|---|
| 0x0 | Kein Status lesbar |
| 0x1 | <i>cSTATUSSetDefaults/ cSTATUSWaitForParameters</i> keine definierter Status |
| 0x2 | <i>cSTATUSUnitOff</i> Gerät ausgeschaltet |
| 0x3 | <i>cSTATUSLoadingBank</i> Bank laden |
| 0x4 | <i>cSTATUSSwitchingUnitOn</i> Gerät einschalten |
| 0x5 | <i>cSTATUSUnitOn</i> Gerät eingeschaltet |
| 0x6 | <i>cSTATUSControllerDisabledByFPGAInternalCause</i> FPGA interne Gründe (des Status erzeugenden Moduls) sperren den Regler |
| 0x7 | <i>cSTATUSControllerEnabled</i> Regler freigegeben |
| 0x8 | <i>cSTATUSSwitchingUnitOff</i> Gerät ausschalten |
| 0x9 | <i>cSTATUSControllerDisabledByCommand</i> Das Kommando <i>cCMDDisableController</i> sperrt den Regler |
| 0xA | <i>cSTATUSControllerDisabledByFPGAExternalCause</i> FPGA externe Gründe (des Status erzeugenden Moduls) sperren den Regler |
| 0xB | <i>cSTATUSResetInterlocks</i> |
| 0xC | <i>cSTATUSMachineProtection</i> |
| 0xD | n.u. |
| 0xE | <i>cSTATUSPowerOnReset</i> |
| 0xF | <i>cSTATUSWhenOthers</i> keine definierter Status |

[15..12] Modul Kommando

| [3..0] | Kommando |
|--------|---|
| 0x0 | <i>cCMDNoAction</i> keine Aktion |
| 0x1 | <i>cCMDSwitchUnitOn</i> Gerät einschalten (wenn möglich) |
| 0x2 | <i>cCMDSwitchUnitOff</i> Gerät abschalten |
| 0x3 | <i>cCMDResetUnit</i> Reset durchführen (Interlocks) |
| 0x4 | <i>cCMDDisableController</i> |

- [11..9] Reserviert
Reserviert für zukünftige Anwendungen
- [8] USIsHighSpeed
wenn ,1' ist USI im Highspeed Mode
- [7..6] Reserviert
Reserviert für zukünftige Anwendungen
- [5] NoInterlocks
wenn ,1' stehen keine Interlocks an
Im Modul sind keine Interlocks gespeichert und es stehen auch keine Interlocks an.
- [4] NoErrors
wenn ,1' ist Modul fehlerfrei
Im Modul sind keine Fehler gespeichert die den Betrieb stören.
- [3] NoWarnings
wenn ,1' ist Modul ohne Warnungen
Im Modul sind keine Warnmeldungen vorhanden die den Betrieb zwar nicht stören aber
trotzdem überprüft werden müssten (Details im FSP für die Warnungsbits) z.B. Tempera-
tur zu hoch.
- [2] ModuleReady
wenn ,1' ist Modul betriebsbereit
Das Modul ist voll betriebsbereit
- [1] ChecksumOK
wenn ,1' Parameter Checksumme OK
Die Prüfsumme für die Modulparameter ist bestätigt.
- [0] ParametersLoaded
wenn ,1' sind die Parameter geladen
Das Modul hat seine Konfigurationsparameter geladen.

| | |
|---------|--|
| Name | FSP002_ModuleWarnings |
| Adresse | 0x02_H/2_D/0x3032_{ASCII} |
| Tiefe | modulabhängig |
| I/O | 3 Byte / 24 Bit |
| Reset | 0x(siehe Beschreibung) _H |

Im FSP werden alle Warnungen bitcodiert aufgelistet die den unmittelbaren Betrieb des Moduls nicht stören, aber trotzdem von einem Techniker untersucht werden müssen, dargestellt (z.B. Temperatur des Moduls zu hoch).

Liegt eine Warnung vor ist das korrespondierende Bit ,0' andernfalls ,1'. Außerdem ist Bit [3] das FSP001_ModuleStatus = ,0'.

[23..0] n.u., immer ,1'

| | |
|---------|--|
| Name | FSP003_ModuleErrors |
| Adresse | 0x03_H/3_D/0x3033_{ASCII} |
| Tiefe | 3 Byte / 24 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

Der FSP enthält alle Fehler die den unmittelbaren Betrieb des Moduls und damit des Übergeordneten Gerätes gefährdet und zu einer Abschaltung führt.

Liegt ein Fehler vor ist das korrespondierende Bit ,0' andernfalls ,1'. Außerdem ist Bit [4] das FSP001_ModuleStatus = ,0'.

[23..0] n.u., immer ,1'

| | |
|---------|--|
| Name | FSP004_ModuleInterlocks |
| Adresse | 0x04_H/4_D/0x3034_{ASCII} |
| Tiefe | 16 Byte / 128 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

Im FSP sind alle Interlocks des Moduls abgebildet sowohl die aktuell anstehenden wie auch die gespeicherten Interlockmeldungen.

Die Grenze liegt in der Mitte des FSP, wobei immer ganze Bytes für die Gruppen verwendet werden müssen. d.h. für 3 Interlocks müssen trotzdem 2 Bytes verwendet werden. 1 Byte für den aktuellen Status und 1 Byte für die gespeicherte Meldung.

Die unteren Bytes [n/2..0] des FSP sind für den aktuellen Status bestimmt und die oberen Bytes [n .. n/2] für die gespeicherten Interlocks.

Liegt ein Interlock vor ist das korrespondierende Bit ,0' andernfalls ,1'. Außerdem ist Bit [5] das FSP001_ModuleStatus = ,0'.

Nicht genutzte Interlockbits müssen ,1' sein!

Gespeicherte Interlocks

[127..104] wenn ,0', Electrical Interlock[40..21]

[103..64] Eingänge werden als normale Interlocks benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind = 0):
es werden nur die geraden Bits (38, 36, .. 2, 0) für die Kanäle[20..1] verwendet, die ungeraden müssen ausmaskiert werden.
wenn '0', Electrical Interlock[20..1]

Eingänge werden für Wasserwächter benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind <> 0):
gerade Bit: zu viel Wasser,
ungerade Bit: zu wenig Wasser
wenn ,0', Wert über, bzw. unterschritten

Aktuell anstehende Interlocks

[63..40] wenn ,0', Electrical Interlock[40..21]

[39..0] Eingänge werden als normale Interlocks benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind = 0):
es werden nur die geraden Bits (38, 36, .. 2, 0) für die Kanäle[20..1] verwendet, die ungeraden müssen ausmaskiert werden.
wenn '0', Electrical Interlock[20..1]

Eingänge werden für Wasserwächter benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind <> 0):
gerade Bit: zu viel Wasser,
ungerade Bit: zu wenig Wasser
wenn ,0', Wert über, bzw. unterschritten

| | |
|---------|--|
| Name | FSP005_InterlocksEnable |
| Adresse | 0x05_H/5_D/0x3035_{ASCII} |
| Tiefe | 5 Byte / 40 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

Im FSP sind alle Interlock Enable Informationen des Moduls abgebildet sofern darüber Informationen zur Verfügung gestellt werden. I.d.R. werden auf Modulen z.B. Jumperstellungen bzgl. der Zulässigkeit von Interlocks abgefragt und deren Einstellungen hier abgelegt.

Die Interlocks des Moduls lassen sich mittels Jumpers (de)aktivieren.

Ist ein Interlock aktiviert (also zugelassen) ist das korrespondierende Bit in diesem FSP gesetzt, nicht zugelassene, also dauerhaft gesperrte Interlocks werden durch eine ,0' dargestellt.

[39..0] wenn '1' Interlock des zugehörigen elektrischen Eingangs [40..1] aktiv

| | |
|---------|--|
| Name | FSP009_ModuleSerialNumber |
| Adresse | 0x09_H/9_D/0x3039_{ASCII} |
| Tiefe | 6 Byte / 48 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

Der FSP enthält die Modul Serien Nummern

Die Serien Nummer ist über einen One Wire Chip von Dallas/Maxim zu erzeugen, da gewährleistet sein muss das die Serien Nummer weltweit nur einmal vergeben ist.

[47..0] Modul Seriennummer

| | |
|---------|---|
| Name | FSP010_ModuleCommands |
| Adresse | 0x0A_H/10_D/0x3041_{ASCII} |
| Tiefe | 1 Byte / 8 Bit |
| I/O | lesen / schreiben |
| Reset | 0x00 _H |

Wenn ein Modul Kommandos unterstützt (Einschalten, Ausschalten, Reset usw.) dann werden diese über diesen FSP gesetzt.

[7..4] n.u.

[3..0] Diese Kommandos werden von der MFU oder PowerConfigAdvanced gesetzt und steuern die Module-/Gerätefunktionen

| [3..0] | Kommando |
|--------|---|
| 0x0 | <i>cCMDNoAction</i> keine Aktion |
| 0x1 | <i>cCMDSwitchUnitOn</i> Gerät einschalten (wenn möglich) |
| 0x2 | <i>cCMDSwitchUnitOff</i> Gerät abschalten |
| 0x3 | <i>cCMDResetUnit</i> Reset durchführen (Interlocks) |
| 0x4 | <i>cCMDDisableController</i> |

| | |
|---------|---|
| Name | FSP011_ModuleInterlocksMask_n |
| Adresse | 0x0B_H/11_D/0x3042_{ASCII} |
| Tiefe | 24 Byte / 192 Bit |
| I/O | Lesen / schreiben |
| Reset | 0x0000000000000000_0000000000000000_0000000000000000 _H |

Dieser FSP enthält Bitmasken, die zum einen nicht verwendete Interlocks vollständig ausmaskiert, d.h. alle nicht zu benutzenden Interlocks sind mit ‚1‘ zu setzen. Zum anderen lassen sich Interlocks mit diesem FSP so maskieren, dass diese erst nach Freigabe des Reglers aktiviert werden. Deren Erfassung wird also während einer Reglersperre ignoriert. Hierbei ist das zugehörige Bit für verzögerte Freigabe des Interlocks (zusammen mit der Reglerfreigabe) mit ‚1‘ zu setzen.

Zugelassene Interlocks nach der Reglerfreigabe

Diese Maske beeinflusst welche Interlocks erst nach der Reglerfreigabe zugelassen werden und ob diese dann vom FPGA erkannt und bearbeitet werden dürfen oder nicht.

Interlocks mit gesetztem Bit (‚1‘) werden erst verzögert mit der erteilten Reglerfreigabe erfasst.

(Beispiel: Der Hauptschutz darf erst zugelassen werden, wenn die Reglerfreigabe erteilt ist. Andernfalls würde das anliegende Hauptschutz Interlock das Einschalten der SVE dauerhaft verhindern. Daher ist das zugehörige Bit auf ‚1‘ zu setzen)

[191..188] n.u.

[187..168] wenn ‚0‘, werden Electrical Interlock[40..21] erfasst

[167..128] Eingänge werden als normale Interlocks benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind = 0):
es werden nur die geraden Bits (38, 36, .. 2, 0) für die Kanäle[20..1] verwendet, die ungeraden müssen ausmaskiert werden.
wenn ‚0‘, werden Electrical Interlock[20..1] erfasst

Eingänge werden für Wasserwächter benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind <> 0):
gerade Bit: zu viel Wasser,
ungerade Bit: zu wenig Wasser
wenn ‚0‘, Wert über, bzw. unterschritten

Gespeicherte Interlocks

Diese Maske beeinflusst gespeicherte Interlocks und ob diese vom FPGA erkannt und bearbeitet werden sollen oder nicht.

[127..124] n.u.

[123..104] wenn ‚0‘, werden Electrical Interlock[40..21] gespeichert

[103..64] Eingänge werden als normale Interlocks benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind = 0):
es werden nur die geraden Bits (38, 36, .. 2, 0) für die Kanäle[20..1] verwendet, die ungeraden müssen ausmaskiert werden.
wenn ‚0‘, werden Electrical Interlock[20..1] gespeichert

Eingänge werden für Wasserwächter benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind <> 0):
gerade Bit: zu viel Wasser,
ungerade Bit: zu wenig Wasser
wenn ‚0‘, Wert über, bzw. unterschritten

Aktuell anstehende Interlocks

Diese Maske beeinflusst aktuell anstehende Interlocks und ob diese vom FPGA erkannt und bearbeitet werden sollen oder nicht

[63..60] n.u.

[59..40] wenn ,0', werden Electrical Interlock[40..21] gespeichert

[39..0] Eingänge werden als normale Interlocks benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind = 0):
es werden nur die geraden Bits (38, 36, .. 2, 0) für die Kanäle[20..1] verwendet, die ungeraden müssen ausmaskiert werden.
wenn '0', werden Electrical Interlock[20..1] gespeichert

Eingänge werden für Wasserwächter benutzt (zugehörige Bits des „FSP063_WaterFlow_Thresholds“ sind $\neq 0$):
gerade Bit: zu viel Wasser,
ungerade Bit: zu wenig Wasser
wenn ,0', Wert über, bzw. unterschritten

| | |
|---------|---|
| Name | FSP012_USIConfig |
| Adresse | 0x0C_H/12_D/0x3043_{ASCII} |
| Tiefe | 1 Byte / 8 Bit |
| I/O | Lesen / schreiben |
| Reset | 0x00 _H |

Dieser FSP definiert die USI Konfiguration

[7] wenn ,1' USI im HighSpeed Modus, wenn ,0' USI im normalen Modus

[4..3] n.u.

[2..0] USI Bitrate

| [2..0] | Bitrate |
|--------|----------------------|
| 111 | 115,2 kBit (default) |
| 110 | 1 MBit |
| 101 | 2 MBit |
| 100 | 5 MBit |
| 011 | 10 MBit |
| 010 | 16,6 MBit |
| 001 | 20 MBit |
| 000 | 25 MBit (Test only!) |

| | |
|---------|---|
| Name | FSP013_PeripheralConfig |
| Adresse | 0x0D_H/13_D/0x3044_{ASCII} |
| Tiefe | 1 Byte / 8 Bit |
| I/O | lesen / schreiben |
| Reset | 0x80 _H |

Auf dem Modul befindliche Peripherie kann mit diesem FSP konfiguriert werden

- [7] Nach einschalten der Spannungsversorgung wird dieses Bit automatisch auf ,1' gesetzt. Werden Parameter durch die MFU oder per PC geladen, die zur Prüfsummenbildung beitragen sollen, muss dieses Bit gelöscht werden, bevor der erste Parameter übertragen wird. Ist das Laden der Parameter beendet, muss dieses Bit wieder auf ,1' gesetzt werden. Im Anschluss daran wird die Vergleichs-Prüfsumme an „FSP058_ParameterChecksumValue“ gesendet. Die Modul-Freigabe erfolgt aber nur, wenn die Vergleichs-Prüfsumme auch zu der aus den restlichen Parametern gebildeten Prüfsumme passt. Das Löschen dieses Bit löscht die zuvor errechnete Prüfsumme.
- [6..0] n.u.

| | |
|---------|---|
| Name | FSP040_RemoteUpdateStatus |
| Adresse | 0x28_H/40_D/0x3238_{ASCII} |
| Tiefe | 1 Byte / 8 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

Beim Fernupdate wird dieser FSP für das Rücklesen der Statusinformationen des Fernupdates verwendet.

[7..3] n.u.

[2] FSP042_Busy
wenn ,1' ist FSP42 beschäftigt (z.B. weil gerade Flashsektoren gelöscht oder programmiert werden) und es sollten KEIN Zugriffe darauf erfolgen

[1] FSP042_ReadyToSendData,
wenn ,1' können Daten vom Host an FSP42 abgeholt werden

[0] FSP042_ReadyToReceiveData,
wenn ,1' können Daten vom Host an FSP42 gesendet werden

| | |
|---------|---|
| Name | FSP041_RemoteUpdateCommands |
| Adresse | 0x29_H/41_D/0x3239_{ASCII} |
| Tiefe | 1 Byte / 8 Bit |
| I/O | lesen / schreiben |
| Reset | 0x00 _H |

Dieser FSP überträgt die Kommandos für das Fernupdate

[7..3] n.u.

[2..0] Kommandos für den RemoteUpdateHandler

| [2..0] | Kommando |
|--------|---|
| 000 | NOP |
| 001 | Erase Bulk, das gesamte Flash löschen |
| 010 | Erase Sector, nur den an 'DataAddress' angegeben Sektor löschen |
| 011 | Write single bytes, ein einzelnes Bytes ins Flash schreiben |
| 100 | Write continuously, beliebige Anzahl Bytes ins Flash schreiben |
| 101 | Read single byte, ein einzelnes Bytes aus dem Flash lesen |
| 110 | Read continuously, beliebige Anzahl Bytes aus dem Flash lesen |
| 111 | Init |

| | |
|---------|---|
| Name | FSP042_RemoteUpdateData |
| Adresse | 0x2A_H/42_D/0x3241_{ASCII} |
| Tiefe | 256 Byte / 2048 Bit |
| I/O | lesen / schreiben |
| Reset | 0x(siehe Beschreibung) _H |

Dieser FSP überträgt die Daten für das Fernupdate entweder vom Host zum Modul oder umgekehrt.

Achtung: Dieses FSP ist besonders im Hinblick auf Lesen und Schreiben. Da das FSP ein nachgeschalteter serieller Flash bedient und die empfangenen Daten direkt in diesen Flash programmiert, bzw. aus dem Flash ausgelesene Daten direkt an den Host versendet werden

Zugriffe einleiten

Zugriffe auf FSP042 müssen generell über FSP041 eingeleitet werden.

Das erste „Kommando“ an FSP041 lautet immer „000“ NOP. Der FSP041 muss mit ACK antworten. Darauf erfolgt das Kommando „111“ (Init). FSP041 muss auch hier mit ACK antworten.

Lesen

Bevor Daten aus dem FSP042 gelesen werden, muss das Lesen mit FSP041 eingeleitet werden.

Zum Lesen eines einzelnen Byte wird das Kommando: „101“ (Read single byte) an FSP041 gesendet. FSP041 muss mit ACK antworten.

Durch lesen von FSP040 lässt sich kontrollieren, ob FSP042 prinzipiell bereit ist Daten zu senden (Bit[1]).

Anschließend wird FSP042 einmalig gelesen. Dabei wird das erste Byte gesendet. Der Ausleseprozess beginnt an Adresse 0x0 und wird automatisch inkrementiert. D.h. wird ein weiteres Lesekommando an FSP042 geschickt, wird das folgende Byte ausgegeben.

Sollen hingegen die Daten seitenweise (jeweils 256 Byte) gelesen werden, erfolgt dies mit dem Kommando: „110“ an FSP041.

Anschließend wird mit jedem Lesebefehl an FSP042 jeweils eine Seite Daten übertragen. Die Seiten werden dabei automatisch inkrementiert.

Schreiben

Bevor Daten sinnvoll ins Flash geschrieben werden können, muss dieses gelöscht werden.

Das Kommando „001“ an FSP041 löscht dieses komplett, das Kommando „010“ an FSP041 hingegen nur die aktuell adressierte Page. Da ein direktes Adressieren der Page im ASCII nicht möglich ist, entfällt die Verwendung dieses Kommandos. In jedem Fall muss FSP042 ACK antworten. Der EPCS Controller beginnt dann unmittelbar mit dem Löschen des Flashs.

Jetzt kann sofort ein erneutes Init-Kommando („111“) an FSP041 gesendet werden. Dieser muss mit ACK antworten.

Jetzt erfolgt die Einleitung des Schreibkommandos.

Zum Schreiben eines einzelnen Byte wird das Kommando: „011“ (Write single bytes) an FSP041 gesendet. FSP041 muss mit ACK antworten.

Sollen hingegen die Daten seitenweise (jeweils 256 Byte) geschrieben werden, erfolgt dies mit dem Kommando: „100“.

Durch lesen von FSP040 lässt sich kontrollieren, ob FSP042 prinzipiell bereit ist Daten zu empfangen, sobald das Bit[0] gesetzt wird. Dieses wird gesetzt, wenn der Löschvorgang abgeschlossen und ein Schreibkommando geschickt wurde. Der Löschvorgang kann bis zu 20 Sekunden dauern.

Anschließend wird abhängig vom Schreibkommando mit dem Schreibbefehl an FSP042 jeweils entweder ein Byte oder jeweils eine Seite Daten ins Flash übertragen. Die Adressen, bzw. Seiten werden dabei automatisch inkrementiert.

Der Schreibvorgang beginnt dabei in jedem Fall bei Adresse 0x0.

Abbrechen/Beenden

Alle Zugriffe (schreiben/lesen) auf den Flash über FSP042 lassen sich mit einem „111“ (Init) an FSP041 abrechnen/beenden.

| | |
|---------|---|
| Name | FSP045_AlteraRemoteUpdateCmd |
| Adresse | 0x2D_H/45_D/0x3244_{ASCII} |
| Tiefe | 6/7 Byte / 48/56 Bit |
| I/O | lesen / schreiben |
| Reset | Reset:0x00100000_00_0_0_0_0 _H |

Dieser FSP dient als Kommando FSP für die Altera Remote Update Funktion

Imagetyyp lesen

Bit[4] = ,0' (Read)

Bit[8] = ,1' (steigende Flanke startet lesen des Imagetyps)

FSP046[1..0] enthält nun den aktuellen Imagetyyp.

Imagetyyp wechseln

Bit[4] = ,1' (Write)

Bit[12] = ,1' (steigende Flanke wechselt das Image)

Mit Einführung des CycloneV änderte sich dieses FSP inhaltlich leicht.

Die Startadresse wanderte um 4 Bits nach links (beginnt nicht mehr bei Bit 20, sondern erst bei Bit 24) und wird um 4 weitere Bits ergänzt (hat also nun die Breite 32 Bits). (18.12.19 – DS)

Altes Format

[47..44] n.u.

[43..20] Flash Start Address (ab dieser Adresse wird das Image geschrieben)

[19..17] n.u.

Neues Format

[55..24] Flash Start Address (ab dieser Adresse wird das Image geschrieben)

[23..17] n.u.

Gemeinsam unverändert

[16] Reset WD Disable (only for debug)

[15..13] n.u.

[12] Start Write (steigende Flanke an diesem Bit startet die FSM zum Imagetyyp-Wechsel)

[11..9] n.u.

[8] Start Read (steigende Flanke an diesem Bit startet die FSM zum lesen des Image-Type)

[7..5] n.u.

[4] Read_n_Write_Enable (muss ,0' sein damit ,Start Read' überhaupt ausgeführt wird, muss ,1' sein damit ,Start Write' überhaupt ausgeführt wird)

[3..2] n.u.

[1..0] Read Source

| | |
|---------|---|
| Name | FSP046_AlteraRemoteUpdateStatus |
| Adresse | 0x2E_H/46_D/0x3245_{ASCII} |
| Tiefe | 10 Byte / 80 Bit |
| I/O | lesen |
| Reset | Reset:0x(siehe Beschreibung) _H |

Dieser FSP dient als Status FSP für die Altera Remote Update Funktion

- [79..72] ReconfTriggerCondition
- [71..69] Force Osc_int n.u.
- [68] Force Osc_int
- [67..44] Boot Address
- [43..41] Wachdog Enable n.u.
- [40] Wachdog Enable
- [39..8] Wachdog timeout
- [7..5] Cd_early n.u.
- [4] Cd_early, wenn ,1' ist ein gültiges Application-Image an der Bootadresse zu finden
- [3..2] MSM State n.u.
- [1..0] MSM State ('00' = Factory Image, '11' = Application Image)

| | |
|---------|---|
| Name | FSP050_ModuleSupplyValues |
| Adresse | 0x32_H/50_D/0x3332_{ASCII} |
| Tiefe | 16 Byte / 128 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

Liefert die vorzeichenbehafteten Betriebsspannungen des Moduls. Immer 2 Byte stehen für eine Spannung. Bei 8 Spannungen ist dieses FSP 16 Byte tief Die Spannungen sind dabei wie folgt sortiert.

- [127..112] Spannung an ADCIN (13 Bit)
- [111..96] vorzeichenbehaftete -12 Volt (13 Bit)
- [95..80] vorzeichenbehaftete 12 Volt (13 Bit)
- [79..64] vorzeichenbehaftete 5 Volt Analog (13 Bit)
- [63..48] vorzeichenbehaftete 5 Volt Digital (13 Bit)
- [47..32] vorzeichenbehaftete 3,3 Volt (13 Bit)
- [31..16] vorzeichenbehaftete 2,5 Volt (13 Bit)
- [15..0] vorzeichenbehaftete 1,2 Volt (13 Bit)

| | |
|---------|---|
| Name | FSP053_ModuleTemperatures |
| Adresse | 0x35_H/53_D/0x3335_{ASCII} |
| Tiefe | 4 Byte / 32 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

Je Temperatur 2 Byte ASCII, also je 1 Byte Vorzeichen behaftetes HEX Zeichen.

[31..24] Alarm_Info

| [7..0] | Bedeutung |
|--------|--|
| 0x00 | n.u., Resetzustand |
| 0x01 | Device 1 nicht bereit |
| 0x02 | Device 2 nicht bereit |
| 0x04 | Device 3 nicht bereit |
| 0x08 | DeviceSearchRunDone |
| 0x10 | Device 1 Grenze überschritten |
| 0x20 | Device 2 Grenze überschritten |
| 0x40 | Device 3 Grenze überschritten |
| 0x80 | Alarm Interrupt wenn Grenze bei einem Device überschritten |

[23..16] Sensor 3: Temperatur Modul Mitte (8 Bit)

[15..8] Sensor 2: Temperatur FPGA (8 Bit)

[7..0] Sensor 1: Temperatur DC-DC Wandler (8 Bit)

Die Schwellen der Temperaturgrenzen werden im „**Fehler! Verweisquelle konnte nicht gefunden werden.**“ festgelegt.

| | |
|---------|---|
| Name | FSP054_ModuleTemperaturesComparisonThresholds |
| Adresse | 0x36_H/54_D/0x3336_{ASCII} |
| Tiefe | 3 Byte / 24 Bit |
| I/O | lesen / schreiben |
| Reset | 0x3C_3C_3C _H |

Stellt die vorzeichenbehafteten Vergleichswerte zur Verfügung bei denen die Temperatursensoren Alarm auslösen sollen, sofern die Temperatur überschritten wurde.

Je Temperatur 2 Byte ASCII, also je 1 Byte Vorzeichen behaftetes HEX Zeichen.

Als Standardwert ist 60° Celsius (60_D = 3C_H) gewählt.

[23..16] Sensor 3: Temperatur Modul Mitte (8 Bit)

[15..8] Sensor 2: Temperatur FPGA (8 Bit)

[7..0] Sensor 1: Temperatur DC-DC Wandler (8 Bit)

| | |
|---------|---|
| Name | FSP058_ParameterChecksumValue |
| Adresse | 0x3A_H/58_D/0x3341_{ASCII} |
| Tiefe | 3 Byte / 24 Bit |
| I/O | lesen / schreiben |
| Reset | 0x000000 _H |

Repräsentiert die Vergleichs-Prüfsumme der vom Modul empfangenen Parameter. Dieser Wert dient zum Vergleich der im Modul errechneten Prüfsumme.

Die Modul-Prüfsumme wird dabei aus den empfangenen Datenbytes durch aufaddieren gebildet und abschließend mit dem Eintrag von „FSP059_ParameterChecksumValueCalculated“ verglichen.

[23..0] Checksumme der Datenübertragung

Die Prüfsumme wird im Modul ChecksumBuilder der Teil von mUISC (modular-USI-control) ist aus den Daten der beschriebenen FSP gebildet und abschließend mit dem Wert dieses FSP verglichen.

| | |
|---------|---|
| Name | FSP059_ParameterChecksumValueCalculated |
| Adresse | 0x3B_H/59_D/0x3342_{ASCII} |
| Tiefe | 3 Byte / 24 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

Repräsentiert die errechnete Prüfsumme innerhalb des Moduls. Gibt die aktuell im Modul errechnete Prüfsumme zurück. Dadurch kann der Fortschritt der Prüfsummenbildung jederzeit verifiziert werden.

[23..0] errechnete Checksumme der Datenübertragung vom PC, bzw. MFU.

| | |
|---------|---|
| Name | FSP060_ValCounter |
| Adresse | 0x3C_H/60_D/0x3343_{ASCII} |
| Tiefe | 4 Byte / 32 Bit |
| I/O | lesen / schreiben |
| Reset | 0x01_00_00_01 _H |

Gibt Zeiten vor, die das Modul für bestimmte Aktionen warten muss.

[31..24] n.u., immer Int(1)

[23..16] Zeitdauer in Sekunden nach dem ,ON' Kommandos, bis die Interlockfassung aktiviert wird, sofern das Interlock im ,Conditional' Modus ist

[15..8] n.u., immer Int(0)

[7..0] n.u., immer Int(1)

| | |
|---------|---|
| Name | FSP061_InputFilterDelay |
| Adresse | 0x3D_H/61_D/0x3344_{ASCII} |
| Tiefe | 10 Byte / 80 Bit |
| I/O | lesen / schreiben |
| Reset | 0x0_FFFFFFFFFFFFFFFF_0000 _H |

It contains the Digital electric interlock filter settings in terms of mask bits and filter delay time. In the module the electric interlocks are 40, so only 60 of 64 mask bits are used for this project.

[79..76] n.u.

[75..16] Mask bit active high: when the interlock is masked and the delay value(see below) is different from zero, the delay between input and output is maximum 10µs (see ACU_InputFilter.docx for more details)

[15..0] Delay value: it defines how long an input interlock has to be ignored (not reported to the output) after its activation. The minimum delay value is 1=>10µs. When it is set to zero, the output will follow the input immediately (no filtering action).

| | |
|---------|--|
| Name | FSP063_WaterFlow_Thresholds |
| Adresse | 0x3F_H/63_D/0x3346_{ASCII} |
| Tiefe | 120 Byte / 960 Bit |
| I/O | lesen / schreiben |
| Reset | 0x000000_000000_000000_000000_000000_000000_000000_000000_ 000000_000000_000000_000000_000000_000000_000000_000000_ 000000_000000_000000_000000_000000_000000_000000_000000_ 000000_000000_000000_000000_000000_000000_000000_000000_ 000000_000000_000000_000000_000000_000000_000000_000000 _H |

Gibt die die Zeiten in μ -Sekunden als Integer-Wert an, die maximal oder und minimal zwischen zwei detektierten Wasserflusswächterpulsen liegen darf.

Bsp.: 100 Pulse bei 1 Liter Durchfluss
und
20 Liter/Minute Durchfluss maximal
→ 2000 Pulse/Minute
→ 33,3 Pulse/Sekunde
→ mit $1/33,3 = 0,03$ Sekunden = 30 mSekunden = 30.000 μ s Zeitspanne zwischen 2 Pulsen

15 Liter/Minute Durchfluss minimal
→ 1500 Pulse/Minute
→ 25 Pulse/Sekunde
→ mit $1/25 = 0,04$ Sekunden = 40 mSekunden = 40.000 μ s Zeitspanne zwischen 2 Pulsen

Die min. Zeit zwischen 2 Pulsen liegt also bei 30.000 μ s (bei 20 Litern Durchfluss), die max. Zeit bei 40.000 μ s (bei 15 Litern Durchfluss).

[959..936] Electrical_In[20]-WaterFlow_20_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[935..912] Electrical_In[20]-WaterFlow_20_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[911..888] Electrical_In[19]-WaterFlow_19_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[887..864] Electrical_In[19]-WaterFlow_19_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[863..840] Electrical_In[18]-WaterFlow_18_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[839..816] Electrical_In[18]-WaterFlow_18_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[815..792] Electrical_In[17]-WaterFlow_17_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[791..768] Electrical_In[17]-WaterFlow_17_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[767..744] Electrical_In[16]-WaterFlow_16_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[743..720] Electrical_In[16]-WaterFlow_16_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[719..696] Electrical_In[15]-WaterFlow_15_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[695..672] Electrical_In[15]-WaterFlow_15_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[671..648] Electrical_In[14]-WaterFlow_14_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[647..624] Electrical_In[14]-WaterFlow_14_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[623..600] Electrical_In[13]-WaterFlow_13_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[599..576] Electrical_In[13]-WaterFlow_13_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[575..552] Electrical_In[12]-WaterFlow_12_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[551..528] Electrical_In[12]-WaterFlow_12_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[527..504] Electrical_In[11]-WaterFlow_11_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)
[503..480] Electrical_In[11]-WaterFlow_11_ValidMinTimeBetweenTwoPulses_in_us (24 Bit)
[479..456] Electrical_In[10]-WaterFlow_10_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit)

| | |
|------------|--|
| [455..432] | Electrical_In[10]-WaterFlow_10_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |
| [431..408] | Electrical_In[09]-WaterFlow_09_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit) |
| [407..384] | Electrical_In[09]-WaterFlow_09_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |
| [383..260] | Electrical_In[08]-WaterFlow_08_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit) |
| [359..336] | Electrical_In[08]-WaterFlow_08_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |
| [335..312] | Electrical_In[07]-WaterFlow_07_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit) |
| [311..288] | Electrical_In[07]-WaterFlow_07_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |
| [287..264] | Electrical_In[06]-WaterFlow_06_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit) |
| [263..240] | Electrical_In[06]-WaterFlow_06_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |
| [239..216] | Electrical_In[05]-WaterFlow_05_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit) |
| [215..192] | Electrical_In[05]-WaterFlow_05_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |
| [191..168] | Electrical_In[04]-WaterFlow_04_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit) |
| [167..144] | Electrical_In[04]-WaterFlow_04_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |
| [143..120] | Electrical_In[03]-WaterFlow_03_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit) |
| [119..96] | Electrical_In[03]-WaterFlow_03_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |
| [95..72] | Electrical_In[02]-WaterFlow_02_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit) |
| [71..48] | Electrical_In[02]-WaterFlow_02_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |
| [47..24] | Electrical_In[01]-WaterFlow_01_ValidMaxTimeBetweenTwoPulses_in_us (24 Bit) |
| [23..0] | Electrical_In[01]-WaterFlow_01_ValidMinTimeBetweenTwoPulses_in_us (24 Bit) |

| | |
|---------|---|
| Name | FSP064_InvertedWaterFlow_TimePeriodeBetweenTwoPulses_in_us |
| Adresse | 0x40_H/64_D/0x3430_{ASCII} |
| Tiefe | 60 Byte / 480 Bit |
| I/O | lesen |
| Reset | 0x(siehe Beschreibung) _H |

Gibt die die Zeiten in μ -Sekunden zwischen zwei detektierten Wasserflusswächterpulsen als invertierte, vorzeichenbehaftete Integer-Wert an.

Beispiel: 100 μ s => 0x64_H wird im 2er Komplement als 0x7FFF9C erfasst und ausgegeben.

[479..456] Electrical_In[20]-WaterFlow_20_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [455..432] Electrical_In[19]-WaterFlow_19_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [431..408] Electrical_In[18]-WaterFlow_18_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [407..384] Electrical_In[17]-WaterFlow_17_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [383..260] Electrical_In[16]-WaterFlow_16_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [359..336] Electrical_In[15]-WaterFlow_15_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [335..312] Electrical_In[14]-WaterFlow_14_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [311..288] Electrical_In[13]-WaterFlow_13_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [287..264] Electrical_In[12]-WaterFlow_12_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [263..240] Electrical_In[11]-WaterFlow_11_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [239..216] Electrical_In[10]-WaterFlow_10_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [215..192] Electrical_In[09]-WaterFlow_09_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [191..168] Electrical_In[08]-WaterFlow_08_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [167..144] Electrical_In[07]-WaterFlow_07_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [143..120] Electrical_In[06]-WaterFlow_06_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [119..96] Electrical_In[05]-WaterFlow_05_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [95..72] Electrical_In[04]-WaterFlow_04_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [71..48] Electrical_In[03]-WaterFlow_03_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [47..24] Electrical_In[02]-WaterFlow_02_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden
 [23..0] Electrical_In[01]-WaterFlow_01_TimePeriode, Dauer zwischen zwei Pulsen in μ -Sekunden