



FSPs des ADC 24Bits Class 2 Moduls (ab FW 7.1)

Version vom: Donnerstag, 22. August 2024, 07:53:00

Inhaltsverzeichnis

1.	Änderungsliste	1
2.	FSPs ADC 24Bits Class 2	2
	FSP000_MDS	3
	0x00 _H /0 _D /0x3030 _{ASCII}	
	FSP001_ModuleStatus	4
	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}	
	FSP006_InterlocksArrivalSequence	8
	0x06 _H /6 _D /0x3036 _{AS}	
	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	12
	0x0C _H /12 _D /0x3043 _{ASCII}	
	FSP013_PeripheralConfig	13
	0x0D _H /13 _D /0x3044 _{ASCII}	
	FSP017_DataStorageStatus	14
	0x11 _H /17 _D /0x3131 _{ASCII}	
	FSP018_DataStorageCfg	15
	0x12 _H /18 _D /0x3132 _{ASCII}	
	FSP030_SetValue_A	16
	0x1E _H /30 _D /0x3145 _{ASCII}	
	FSP040_RemoteUpdateStatus	17
	0x28 _H /40 _D /0x3238 _{ASCII}	
	FSP041_RemoteUpdateCommands	18
	0x29 _H /41 _D /0x3239 _{ASCII}	
	FSP042_RemoteUpdateData	19
	0x2A _H /42 _D /0x3241 _{ASCII}	
	FSP043_FSP_Image_Generator	21
	0x2B _H /43 _D /0x3242 _{ASCII}	
	FSP044_FSP_Image_Status	22
	0x2C _H /44 _D /0x3243 _{ASCII}	
	FSP045_AlteraRemoteUpdateCmd	23
	0x2D _H /45 _D /0x3244 _{ASCII}	
	FSP046_AlteraRemoteUpdateStatus	24
	0x2E _H /46 _D /0x3245 _{ASCII}	
	FSP051_ModulePotIValues	25
	0x33 _H /51 _D /0x3333 _{ASCII}	
	FSP052_ModuleComparatorValues	26
	0x34 _H /52 _D /0x3334 _{ASCII}	
	FSP053_ModuleTemperatures	27
	0x35 _H /53 _D /0x3335 _{ASCII}	
	FSP058_ParameterChecksumValue	28
	0x3A _H /58 _D /0x3341 _{ASCII}	
	FSP059_ParameterChecksumValueCalculated	29
	0x3B _H /59 _D /0x3342 _{ASCII}	
	FSP060_ADC_Configuration	30
	0x3C _H /60 _D /0x3343 _{ASCII}	
	FSP061_ADCCalibrationGAIN	31
	0x3D _H /61 _D /0x3344 _{ASCII}	
	FSP062_ADCCalibrationOFFSET	32
	0x3E _H /62 _D /0x3345 _{ASCII}	
	FSP063_ADCFAVResult	33
	0x3F _H /63 _D /0x3346 _{ASCII}	
	FSP064_DCCT_CorrectionFactor	34
	0x40 _H /64 _D /0x3430 _{ASCII}	
	FSP065_OnChipADCScale	35

0x41 _H /65 _D /0x3431 _{ASCII}	
FSP066_ADC_SyncTB.....	36
0x42 _H /66 _D /0x3432 _{ASCII}	
FSP073_Controller_Limits.....	37
0x49 _H /73 _D /0x3439 _{ASCII}	
FSP074_Controller_PI_Settings.....	38
0x4A _H /60 _D /0x3441 _{ASCII}	
FSP075_Controller_I_Part_ComparatorLimits.....	39
0x4B _H /75 _D /0x3442 _{ASCII}	
FSP076_Controller_SetValueDeviation.....	40
0x4C _H /76 _D /0x3443 _{ASCII}	
FSP077_Controller_PI_Output.....	41
0x4D _H /77 _D /0x3444 _{ASCII}	
FSP078_SerialComDt2SendSel.....	42
0x4E _H /78 _D /0x3445 _{ASCII}	
FSP079_SerialCom_1_ErrorGenCfg.....	43
0x4F _H /79 _D /0x3446 _{ASCII}	
FSP080_SerialCom_1_Status.....	44
0x50 _H /80 _D /0x3530 _{ASCII}	
FSP081_SerialCom_2_ErrorGenCfg.....	45
0x51 _H /81 _D /0x3531 _{ASCII}	
FSP082_SerialCom_2_Status.....	46
0x52 _H /82 _D /0x3532 _{ASCII}	
FSP083_Redundant_DCCT_Thrs.....	47
0x53 _H /83 _D /0x3533 _{ASCII}	
FSP084_Redundant_DCCT_Cfg.....	48
0x54 _H /84 _D /0x3534 _{ASCII}	
FSP085_Redundant_DCCT_Status.....	49
0x55 _H /85 _D /0x3535 _{ASCII}	

1. Änderungsliste

Datum	Name	Kommentar
16.08.2024	D. Schupp	Dokument erstellt
22.08.2024	D. Schupp	FSP079_SerialCom_1_ErrorGenCfg FSP081_SerialCom_2_ErrorGenCfg Inhalte ergänzt

2. FSPs ADC 24Bits Class 2

Dieses Dokument behandelt modulspezifische FSPs des ADC 24Bits Class 2 FG660_424 und höher.

Name	FSP000_MDS
Adresse	0x00_H/0_D/0x3030_{ASCII}
Tiefe	n Byte / m Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Die hier **rot** hervorgehobenen Werte sind die Erkennungsmerkmale für PowerConfigAdvanced (PCA).

Parameter (Generisch)	Inhalt/Wert	Typ
wModuleClass	1	signed integer
wModuleSubClass	8	signed integer
wVendorID	1 (GSI)	signed integer
lProductID	86	signed integer
wUSI	17	signed integer
bDeviceMaxSpeed	1 (20MBit)	signed integer
wAttributes	2 (HighSpeed support)	signed integer
gAttrib_Bit7_nFac_Applic_Image	1 (Application image)	enumerated
lMaxPower	0	signed integer
sDescription	ADC 24Bits Class2	string
Parameter (Ports)		
wHWMajorRelease	0x nn _H	8 Bit hex.
lHWMInorRelease	0x nnnn _H	16 Bit hex.
sHWDate	0x 010119 _H ^{*)}	24 Bit hex.
wFWMajorRelease	0x 07 _H	8 Bit hex.
lFWMinorRelease	0x xyyy _H ^{**)}	16 Bit hex.
sFWDDate	0x ddmmyy _H ^{*)}	24 Bit hex.

^{*)} hexadezimaler Wert entspricht dem 01.01.19.

Dies ist das Mindestdatum für die Modulunterstützung in PCA

^{**)} die oberen 8 Bit (xx) entscheiden darüber, welche Version des Forms in PCA geladen wird. Die unteren 8 Bit (yy) können sich ändern, ohne dass es eine Änderung des PCA Forms bedarf.

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

- [23..9] *Reserviert*
Reserviert für zukünftige Anwendungen
n.u. immer ,0'
- [8] *USIsHighSpeed*
wenn ,1' ist USI im Highspeed Mode
- [7..6] *Reserviert*
Reserviert für zukünftige Anwendungen
n.u. immer ,0'
- [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.
z. Zt. immer ,1'
- [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. Temperatur zu hoch.
z. Zt. immer ,1'
- [2] *ModuleReady*
wenn ,1' ist Modul betriebsbereit
Das Modul ist voll betriebsbereit
z. Zt. immer ,1'
- [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.
z. Zt. immer ,1'

Name	FSP002_ModuleWarnings
Adresse	0x02_H/2_D/0x3032_{ASCII}
Tiefe	1 Byte / 8 Bit
I/O	lesen
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‘.

[7..0] *n.u.*, immer ‚1‘

Name	FSP003_ModuleErrors
Adresse	0x03_H/3_D/0x3033_{ASCII}
Tiefe	1 Byte / 8 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'.

[7..0] *n.u.*, immer ,1'

Name	FSP004_ModuleInterlocks
Adresse	0x04_H/4_D/0x3034_{ASCII}
Tiefe	2 Byte / 16 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

[15..13] *n.u.*, immer ,1'

[12] wenn '0', *Steckerüberwachung DCCT*, wird auch ADC stirnseitig durch Wechselblinken der roten LEDs signalisiert.

[11] wenn '0', *Komparator 1 Positiv-Überwachung*

[10] wenn '0', *Komparator 2 Positiv-Überwachung Mittelwert*

[9] wenn '0', *Komparator 3 Negativ-Überwachung*

[8] wenn '0', *Komparator 4 Negativ-Überwachung Mittelwert*

Aktuell anstehende Interlocks

[7..5] *n.u.*, immer ,1'

[4] wenn '0', *Steckerüberwachung DCCT*, wird auch ADC stirnseitig durch Wechselblinken der roten LEDs signalisiert.

[3] wenn '0', *Komparator 1 Positiv-Überwachung*

[2] wenn '0', *Komparator 2 Positiv-Überwachung Mittelwert*

[1] wenn '0', *Komparator 3 Negativ-Überwachung*

[0] wenn '0', *Komparator 4 Negativ-Überwachung Mittelwert*

Name	FSP006_InterlocksArrivalSequence
Adresse	0x06_H/6_D/0x3036_{ASCII}
Tiefe	4 Byte / 32 Byte
I/O	lesen
Reset	Reset:0x(siehe Beschreibung) _H

Dieser FSP liefert die zeitliche Abfolge auftretender Interlocks. Liegen mehrere Interlocks an, kann über diesen FSP die zeitliche Abfolge von deren Auftreten gelesen werden.

Zu beachten ist, treten Interlocks zeitgleich (also im selben Taktzyklus) auf, wird nur das hochwertigste Interlockbit erfasst und in diesem FSP abgelegt.

Werden also z.B. die Interlocks Bit[1], Bit[3] und Bit[7] im selben Takt erfasst, wird nur Bit[7] im FSP abgelegt.

Das FSP erfasst die 3 zuerst auftretenden Interlocks dieses Moduls.

- [31..24] Das Interlockbit des zuerst aufgetretenen Interlocks (i)
- [23..16] Das Interlockbit des vor (i) aufgetretenen Interlocks (i-1)
- [15..8] Das Interlockbit des zuletzt aufgetretenen Interlocks (i-2)
- [7..0] Anzahl der erfassten Interlock im FSP006 insgesamt (0..3)

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 Nummer

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

[47..0] *Seriennummer* des Moduls (48 Bit)

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.

Die Kommandos dieses FSPs werden nur ausgeführt, wenn das Modul mittels Standard-USI angebunden ist. Wird die USI HighSpeed Verbindung verwendet, dann werden die Kommandos aus diesem FSP ignoriert. Stattdessen erfolgt die Kommandoübertragung über die HighSpeed Anbindung.

[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 (z.B. Interlocks)
0x4	<i>cCMDDisableController</i>
0x5	<i>cCMDTriggerSomething</i> hiermit lassen sich Sonderfunktionen in Modulen auslösen

Für das ADC Modul ist nur das Kommando „0x03 (Reset)“ von Belangen um evtl. angefallene und nicht mehr anstehende Interlocks zu löschen.

Name	FSP011_ModuleInterlocksMask_n
Adresse	0x0B_H/11_D/0x3042_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	Lesen / schreiben
Reset	0xE0_E0

Dieser FSP enthält eine Bitmaske, die nicht verwendete Interlocks ausmaskiert, d.h. alle nicht zu benutzenden Interlocks sind mit 1 zu setzen.

Wie beim FSP004_ModuleInterlocks liegt die Grenze in der Mitte des FSP, wobei immer ganze Bytes für die Gruppen verwendet werden müssen. d.h. für 5 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.

Gespeicherte Interlocks

[15..13] n.u., immer ,1'

[12] wenn ,1' → *Steckerüberwachung DCCT* deaktiviert, wird auch ADC stirnseitig durch Wechselblinken der roten LEDs signalisiert.

[11] wenn ,1' → *Komparator 1 Positiv-Überwachung* deaktiviert

[10] wenn ,1' → *Komparator 2 Positiv-Überwachung Mittelwert* deaktiviert

[9] wenn ,1' → *Komparator 3 Negativ-Überwachung* deaktiviert

[8] wenn ,1' → *Komparator 4 Negativ-Überwachung Mittelwert* deaktiviert

Aktuell anstehende Interlocks

[7..5] n.u., immer ,1'

[4] wenn ,1' → *Steckerüberwachung DCCT* deaktiviert, wird auch ADC stirnseitig durch Wechselblinken der roten LEDs signalisiert.

[3] wenn ,1' → *Komparator 1 Positiv-Überwachung* deaktiviert

[2] wenn ,1' → *Komparator 2 Positiv-Überwachung Mittelwert* deaktiviert

[1] wenn ,1' → *Komparator 3 Negativ-Überwachung* deaktiviert

[0] wenn ,1' → *Komparator 4 Negativ-Überwachung Mittelwert* deaktiviert

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	FSP017_DataStorageStatus
Adresse	0x11_H/17_D/0x3131_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Enthält den DataStorage Status

[15..8] Wert n der Adressweite der Erfassung (2^n), mit $13 \Rightarrow 2^{13}$ = mögliche Speicheradressen
Die oberen 8 Bit des FSP017 verraten wieviel Data das DataStorage-Modul später ggf. senden wird. Der darin befindliche Wert gibt die Anzahl der Adressleitungen an. D.h. wenn der Wert "&H1" um diese Anzahl nach links verschoben wird ($2^{\text{dieser Wert}}$), ergibt das die Speichertiefe des DataStorage-Moduls.

Beispiel:

"&H0D" \Rightarrow 13(d) \Rightarrow 0x0_0000_0000_0001

um 13 geschoben = 0x1_0000_0000_0000 = 2^{13}

DataStorage hat 13 Adressleitungen für 32 Bit Werte \Rightarrow 8192 mögl. Adressen für 32 Bit Werte. Es werden bei "jedem Schuss" aber 64 Bit Daten (also 2×32 Bit) geschrieben, d.h. es bleiben somit eigentlich nur 12 Adressleitungen für 32 Bit Werte \Rightarrow 4096 mögl. Adressen für 64 Bit Werte. Da jede Speicherstelle 32 Bit tief ist, werden 8192×4 Bytes = 32768 Bytes gespeichert. Wird DataStorage also mit 13 Adressleitungen gelesen, werden insgesamt 32kByte Daten empfangen.

Es werden also insgesamt nur $(2^{13})/2$ Bytes empfangen

Welche Daten erfasst werden und anschließend zur Verfügung stehen bitte dem „FSP018_DataStorageCfg“ auf Seite 15 entnehmen.

[7..3] *n.u.*

[2] *Trigger condition detected*

[1] *Read complete*

[0] *Write complete*

Name	FSP018_DataStorageCfg
Adresse	0x12_H/18_D/0x3132_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen /schreiben
Reset	0_0000_0000_0_0_0 _H

Enthält die DataStorage Konfiguration

[47..44] *n.u.*

[43..28] *preTriggerLocationNumber* : it can be left to default 0 value (16b) in “internal oscilloscope” mode, this parameter defines how many RAM location have to be not overwritten before the trigger condition detection.
It is useful to implement a pre-trigger, center or post trigger acquisition mode.

[27..12] *Number of skippable values* : it can be left to default 0 value(16b) it defines the number of ADC values that have not to be written in the RAM. When 0, all the received values are written.

[11..9] *n.u.*

[8] *ExtTiggerCommand* : it is used to trigger the write action
the rising edge of this signal stops the write action in “loop” mode. The ACU_DataStorage module continues to write the RAM till the location where the external trigger condition was detected. After that the write actions are stopped. This functionality is useful when the ACU_Data storage module is used as “internal oscilloscope”.

[7..5] *n.u.*

[4] *RdEnable* : it is used to arm the read action when the write is completed it has to be set to 1 (level) during the read operation when the write complete status is reported on the FSP 123.

[3..2] *n.u.*

[0] *Reset command*, necessary before start to record data

Name	FSP030_SetValue_A
Adresse	0x1E_H/30_D/0x3145_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x000000 _H

Über diesen FSP wird der Sollwert_A für die Temperatur-Regler PWM der ADC Heizung angegeben.

[23..0] Vorzeichenfreier Sollwert (0x000000 bis 0xFFFF00)

Die Sollwerttemperatur steht in Korrelation zum gemessenen Istwert des HDC1080.

Das Temperaturregister des HDC1080 ist ein 16-Bit-Ergebnisregister im Binärformat.
(Die 2 LSBs D1 und D0 sind immer 0).

Das Ergebnis der Erfassung ist immer ein 14-Bit-Wert (UNSIGNED).

Die Genauigkeit des Ergebnisses hängt von der gewählten Wandlungszeit ab.

Die Temperatur kann aus den Ausgabedaten berechnet werden mit:

$$\text{Temperatur (}^\circ\text{C)} = (\text{code}[15..0] / 2^{16}) * 165^\circ\text{C} - 40^\circ\text{C}$$

code[15..0] entspricht in diesem Fall [23..8] des vorzeichenfreien Sollwert.

Im Umkehrschluss erfolgt die Generierung des FSP Inhalts mit:

$$\text{code}[15..0] = ((\text{Temperature (}^\circ\text{C)} + 40^\circ\text{C}) / 165^\circ\text{C}) * 2^{16}$$

$$\text{FSP030_SetValue_A} = \text{code}[15..0] \& 0x00$$

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 bytes, 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 ADCII 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	FSP043_FSP_Image_Generator
Adresse	0x2B_H/43_D/0x3242_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x00_00_00 _H

Über diesen FSP wird ein FSP Image erzeugt und im seriellen Flash gespeichert, bzw. die Images der FSPs aus dem seriellen Flash geladen. Der serielle Flash ist hierdurch auch löscher. Statusrückmeldungen gibt das „FSP044_FSP_Image_Status“.

Hinweis

Zwischen den Funktionen „FSPImgLoadImage“, „FSPImgMakelImage“ und „FSPImgEraseAll“ dürfen die Bitwechsel nicht unmittelbar erfolgen. D.h. sofern eine der Funktion benutzt wurde, muss das zugehörige Bit zuerst gelöscht werden, bevor das Bit einer anderen Funktion gesetzt wird.

- [23..16] *FSPImgSectorAddress*, sofern *FSPImgUseSectorAddress*; = ,1' wird der hier angegebene Sektor im ext. seriellen Flash gelöscht, zur Sicherung, bzw. zum Laden von FSP Daten genutzt.
- [15..8] *FSPImgFSPNumber*, gibt die FSP Nummer an, die mittels ‚FSPImgMakelImage‘ ins ext. serielle Flash gesichert werden soll.
- [7..4] *n.u.*
- [3] *FSPImgUseSectorAddress*, wenn ‚1‘ wird das Image in den ‚FSPImgSectorAddress‘ Sektor geschrieben, wenn ‚0‘ wird die Startadresse aus dem ‚M25P_Access‘ Modulgeneric ‚gFSPImgStartAddress‘ verwendet.
- [2] *FSPImgEraseAll*, wenn ‚1‘ wird ein Sektor im ext. seriellen Flash gelöscht. Dieser ist abhängig von ‚FSPImgUseSectorAddress‘ und ‚FSPImgSectorAddress‘ oder ‚gFSPImgStartAddress‘. Zum Starten des Löschvorgangs muss ein ‚0‘ → ‚1‘ Übergang erfolgen.
- [1] *FSPImgMakelImage*, wenn ‚1‘ wird der Inhalt des FSP ‚FSPImgFSPNumber‘ ins ext. serielle Flash gesichert. Zum Starten des Sicherungsvorgangs muss ein ‚0‘ → ‚1‘ Übergang erfolgen. Sollen mehrere FSP gesichert werden ist ‚FSPImgMakelImage‘ mehrfach zu aktivieren und die jeweilige FSP Nummer in ‚FSPImgFSPNumber‘ anzugeben.
- [0] *FSPImgLoadImage*, wenn ‚1‘ wird der Inhalt des ext. seriellen Flashs in die FSPs geladen. Sind zuvor mehrere FSP gesichert worden, werden alle Sicherungen geladen. Zum Starten des Ladevorgangs muss ein ‚0‘ → ‚1‘ Übergang erfolgen.

Name	FSP044_FSP_Image_Status
Adresse	0x2C_H/44_D/0x3243_{ASCII}
Tiefe	1 Byte / 8 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Liefert Statusrückmeldungen bzgl. des Sicherungsvorgangs von FSP Daten. Die Sicherung wird über FSP043 gesteuert.

[7..1] *n.u.*, immer ,0'

[0] *FSPImgReady*, ist ,1' wenn das ext. Flash Zugriffe zum sichern/laden von FSP Daten gestattet, ist ,0' wenn Daten gesichert oder gelesen werden, bzw. ein Löschvorgang durchgeführt wird. Ist diese Bit ,0' und mittels FSP043 werden Zugriffe auf das ext. Flash gestartet ist das Ergebnis undefiniert.

Name	FSP045_AlteraRemoteUpdateCmd
Adresse	0x2D_H/45_D/0x3244_{ASCII}
Tiefe	7 Byte / 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

Imagetyp lesen

Bit[4] = ,0' (Read)

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

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

Imagetyp wechseln

Bit[4] = ,1' (Write)

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

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

[23..17] *n.u.*

[16] *Reset WD Disable* (only for debug)

[15..13] *n.u.*

[12] *Start Write* (steigende Flanke an diesem Bit startet die FSM zum Imagetyp-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	FSP051_ModulePotiValues
Adresse	0x33_H/51_D/0x3333_{ASCII}
Tiefe	8 Byte / 64 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Liefert die vorzeichenbehafteten Poti-Spannungen des Moduls. Immer 2 Byte stehen für eine Spannung. Bei 4 Spannungen ist dieses FSP 8 Byte tief. Die Spannungen sind dabei wie folgt sortiert.

- [63..48] vorzeichenbehaftete *Poti-Spannung Komparator 4* (13 Bit)
- [47..32] vorzeichenbehaftete *Poti-Spannung Komparator 3* (13 Bit)
- [31..16] vorzeichenbehaftete *Poti-Spannung Komparator 2* (13 Bit)
- [15..0] vorzeichenbehaftete *Poti-Spannung Komparator 1* (13 Bit)

Name	FSP052_ModuleComparatorValues
Adresse	0x34_H/52_D/0x3334_{ASCII}
Tiefe	8 Byte / 64 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Liefert die vorzeichenbehafteten Komparatorspannungen des Moduls. Immer 2 Byte stehen für eine Spannung. Bei 4 Spannungen ist dieses FSP 8 Byte tief. Die Spannungen sind dabei wie folgt sortiert.

[63..48] vorzeichenbehaftete *Spannung Komparator 4*

[47..32] vorzeichenbehaftete *Spannung Komparator 3*

[31..16] vorzeichenbehaftete *Spannung Komparator 2*

[15..0] vorzeichenbehaftete *Spannung Komparator 1*

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

Humidity and temperature sensor read back values

[31..16] *Humidity (16b)*

Das Feuchteregister ist ein 16-Bit-Ergebnisregister im Binärformat.

(Die 2 LSBs D1 und D0 sind immer 0).

Das Ergebnis der Erfassung ist immer ein 14-Bit-Wert (UNSIGNED), wobei die Genauigkeit von der gewählten Wandlungszeit abhängt.

Die Luftfeuchtigkeit kann aus den Ausgabedaten berechnet werden mit:

$$\text{Relative Humidity (\%RH)} = (\text{HUMIDITY}[15..0] / 2^{16}) * 100\%RH$$

[15..0] *Temperature (16b)*

Das Temperaturregister ist ein 16-Bit-Ergebnisregister im Binärformat.

(Die 2 LSBs D1 und D0 sind immer 0).

Das Ergebnis der Erfassung ist immer ein 14-Bit-Wert (UNSIGNED).

Die Genauigkeit des Ergebnisses hängt von der gewählten Wandlungszeit ab.

Die Temperatur kann aus den Ausgabedaten berechnet werden mit:

$$\text{Temperatur (}^{\circ}\text{C)} = (\text{TEMPERATURE}[15..0] / 2^{16}) * 165^{\circ}\text{C} - 40^{\circ}\text{C}$$

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 Module 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 „FSP058_ParameterChecksumValue“ verglichen.

[23..0] *Checksumme der Datenübertragung zum ADC.*

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_ADC_Configuration
Adresse	0x3C_H/60_D/0x3343_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x0000_0000_00 _H

Repräsentiert verschiedene Konfigurationszustände des ADC Moduls.

[39..24] *Terminal counter external time-base:*
it can be any value from 100 to $(2^{16})-1$.
ex:100 => the external time-base period is 1us (clock period=10ns)
→SyncTimebase(0..11)

[23..8] *ADC Mittelwertbildung innerhalb des LTC2380 24b ADCs*

[7..0]	Mittelwertbildung
0x00	Mittelwertbildung aus
0x01	1fach Mittelung
0x02	2fach Mittelung
0x03	3fach Mittelung
...	
0x64	64fach Mittelung

[7..6] *n.u.*

[5] *External time-base enable*

[4] wenn ,1', *FlushFIFO* der Mittelwertbildung innerhalb der Firmware

[3..1] *n.u.*

[0] wenn ,1', *ADC Mittelwertbildung* innerhalb der Firmware eingeschaltet

Name	FSP061_ADCCalibrationGAIN
Adresse	0x3D_H/61_D/0x3344_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x80_0000 _H

Wird als Hilfs-FSP zur Kalibrierung des ADC Moduls verwendet. Enthält die Verstärkungskorrektur (Skalierungsfaktor) des ADC, der über PowerConfigAdvanced beim Kalibriervorgang ermittelt und dann nicht flüchtig im ADC Moduls gespeichert wird. Nach dem Einschalten der Spannungsversorgung wird dieses FSP im Falle einer durchgeführten Kalibrierung mit dem gespeicherten Kalibrierwert geladen. Der Resetwert wird dabei überschrieben.

[23..0] Vorzeichenfreier 24 Bit *Skalierungswert zur Korrektur der ADC Verstärkung*
 Der Gain-Wert ist als vorzeichenloser Festkommawert zu interpretieren, wobei das MSB für den Integer Teil und die restlichen Bits für den Dezimalwert stehen.
 Der Verstärkungswert kann 1,9...V nicht überschreiten.

Beispiel

Gain = 0,5V → 0,5 * 2²³ = 4194304 = 0x40_0000
 1,0V → 1 * 2²³ = 8388608 = 0x80_0000
 1,999999999V → 1,9.. * 2²³ = 16777215 = 0xFF_FFFF (Max value)

Name	FSP062_ADCCalibrationOFFSET
Adresse	0x3E_H/62_D/0x3345_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x00_0000 _H

Wird als Hilfs-FSP zur Kalibrierung des ADC Moduls verwendet. Enthält die Offsetkorrektur des ADC, der über PowerConfigAdvanced beim Kalibriervorgang ermittelt und dann nicht flüchtig im ADC Modul gespeichert wird. Nach dem Einschalten der Spannungsversorgung wird dieses FSP im Falle einer durchgeführten Kalibrierung mit dem gespeicherten Kalibrierwert geladen. Der Resetwert wird dabei überschrieben.

[23..0] Vorzeichenbehaftete 24 Bit Offsetkorrektur des ADC
Der Offsetwert ist als vorzeichenbehafteter Wert zu interpretieren.
(V/LSB = V/(Vpp/2^{AnzahlBits})).

Beispiel

Vpp = +/- 11V

Bits = 24

Offset = 5,5V → $5,5 / (11 / 2^{24}) = 4194304 = 0x40_0000$

1,0V → $1,0 / (11 / 2^{24}) = 762600 = 0x0B_A2E8$

-1,0V → $-1,0 / (11 / 2^{24}) = -762600 = 0xF4_5D18$

Name	FSP063_ADCFAVResult
Adresse	0x3F_H/63_D/0x3346_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Gibt den gleitenden Mittelwert des ADC zurück (FAV: Floating Average Value). Der ADC läuft frei und erfasst permanent Werte. Diese werden ggf. gemittelt (siehe: „FSP060_ADC_Configuration“)

[23..0] Vorzeichenbehafteter ADC Wert (24 Bit), ggf. gemittelt

Name	FSP064_DCCT_CorrectionFactor
Adresse	0x40_H/64_D/0x3430_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen / schreiben
Reset	0x00000000 _H

Korrekturwerte für den DCCT Abgleich des gemessenen ADC Wertes (FSP063_ADCFAVResult)

[31..0] FSP64_DCCT_CorrectionFactor
Vorzeichenfrei. Die oberen 8 Bits stehen für den Integer Teil und die restlichen Bits für den Dezimalwert.

Name	FSP065_OnChipADCScale
Adresse	0x41_H/65_D/0x3431_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x800000 _H

Wenn der OnChip-Mittelungsmodus des LTC2380 24-Bit ADC ausgewählt ist, bezieht der ADC-Chip seinen integrierten FIR-Digitalfilter mit ein, um die gemittelten Daten bereitzustellen. Basierend auf der gewählten Abtastfrequenz und der Anzahl der Mittelungen kann eine bestimmte Eingangsfrequenz mit ihren Harmonischen unterdrückt/abgeschnitten werden. Die zu unterdrückende Grundfrequenz berechnet sich wie folgt:

$$f_{REJECT} = \frac{f_{SAMPLE}}{N}$$

Die Grundfrequenz f_{REJECT} und ihre Harmonischen bis zur Frequenz $f_{SAMPLE} - f_{REJECT}$ werden verworfen (Verstärkung=0). Die höchste Abtastfrequenz beträgt 1 MSps, während die Ausgangsdatenrate vom gewählten Betriebsmodus abhängig ist. Im konventionellen Betriebsmodus ist die Ausgangsdatenrate gleich der Abtastrate, im Mittelungsmodus ist sie $\frac{1}{(N*1\mu s)}$.

Bei $N = 2$ beträgt die Ausgangsdatenrate 500 KSps (jedes Sample ist 24 Bit lang).

Zu beachten ist, dass für N-Werte, die keine Potenz von 2 sind (wie zum Beispiel 5), der konvertierte Wert am Ausgang mit einem Korrekturfaktor gleich $\frac{N}{M}$ skaliert werden muss, wobei M die Potenz von 2 näher und höher ist als N.

$$(M = 2^{\text{ceil}(\log_2 N)})$$

Im Fall $N=5$ ist der Skalierungsfaktor $5/8$.

$$M = 2^{\text{ceil}(\log_2 5)} = 2^{\text{ceil}(2.321\dots)} = 2^3 = 8$$

Daraus folgt: OnChipADCScale = $5/8 = 0,625$

Die SCLK-Taktfrequenz beträgt 50 MHz mit 50 % Arbeitszyklus, die SDI-Leitung wird auf der SCLK-Anstiegsflanke abgetastet und der SCLK-Ruhezustand ist niedrig.

- [23..0] Vorzeichenfreier 24 Bit Skalierungswert zur Korrektur der ADC Verstärkung
 Der Wert ist als vorzeichenloser Festkommawert zu interpretieren, wobei das MSB für den Integer Teil und die restlichen Bits für den Dezimalwert stehen.
 Der Wert kann 1,9...V nicht über- und 0 nicht unterschreiten.

Beispiel

OnChipADCScale = 0,0V	→ 0	*2 ²³ =	0 = 0x00_0000
0,5V	→ 0,5	*2 ²³ =	4194304 = 0x40_0000
0,625	→ 0,625	*2 ²³ =	5242880 = 0x50_0000
1,0V	→ 1	*2 ²³ =	8388608 = 0x80_0000
1,9999999999V	→ 1,9..	*2 ²³ =	16777215 = 0xFF_FFFF (Max value)

Name	FSP066_ADC_SyncTB
Adresse	0x42_H/66_D/0x3432_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x0_0_0000 _H

[22..20} SelSyncOut[2..0] – ADC_SyncTimeBase(0..11)

[18..17] SelSyncIn[1..0] – ADC_SyncTimeBase(0..11)

[16] M_nS – ADC_SyncTimeBase(0..11)

[15..0] TprSet[15..0] – ADC_SyncTimeBase(0..11)

Name	FSP073_Controller_Limits
Adresse	0x49_H/73_D/0x3439_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Über diesen FSP können die Bereichsgrenzen des PI Reglers festgelegt werden. Der Regler dient der Temperaturreglung des ADCs.

[47..24] Controller_MaxVal (20 Bit), repräsentiert den oberen (maximalen) Grenzwert des PI Regler

[23..0] Controller_MinVal (20 Bit), repräsentiert den unteren (minimalen) Grenzwert des PI Regler

Name	FSP074_Controller_PI_Settings
Adresse	0x4A_H/60_D/0x3441_{ASCII}
Tiefe	13 Byte / 104 Bit
I/O	lesen / schreiben
Reset	0x00_00000000_00000000_00000000 _H

Über diesen FSP können die I und P1, P2 Anteile des PI Reglers gesetzt werden. Der Regler dient der Temperaturreglung des ADCs.

[103..97] n.u

[98] Wenn ,1' ist der P2 Anteil des Reglers aktiv.

[97] Wenn ,1' wird der Komparator zur Deaktivierung des I-Anteils aktiviert.
Die Schwellwerte für diesen Komparator stellt
„FSP075_Controller_I_Part_ComparatorLimits“ zur Verfügung.

[96] Wenn ,1' wird der I-Anteil des Reglers um den Faktor 1000 verlangsamt.

[95..64] Controller1_I_Part (32 Bit), repräsentiert den I Anteil des PI Reglers.

[63..32] Controller1_P2_Part (32 Bit), repräsentiert den 2. P Anteil des PI Reglers.

[31..0] Controller1_P1_Part (32 Bit), repräsentiert den 1. P Anteil des PI Reglers.

Name	FSP075_Controller_I_Part_ComparatorLimits
Adresse	0x4B_H/75_D/0x3442_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Repräsentiert die Bereichsgrenzen in denen der I Anteil bei der Regelung berücksichtigt werden soll. Der Regler dient der Temperaturreglung des ADCs.

[47..24] Controller_I_Part_ComparatorOFFThreshold (20 Bit)

[23..0] Controller_I_Part_ComparatorONThreshold (20 Bit)

Name	FSP076_Controller_SetValueDeviation
Adresse	0x4C_H/76_D/0x3443_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann die Regeldifferenz (Delta I) des PI Reglers gelesen werden. Der Regler dient der Temperaturregung des ADCs. Die Regeldifferenz wird dabei im Differenzbildner des Reglers aus dem Minuend SetValue und dem Subtrahend ActualValue gebildet

[23..0] SetValueDeviation

Der Differenzbildner erwartet 2er-Komplementdaten an den Eingängen Soll-, bzw. Istwert.

Das Temperaturregister des HDC1080 ist ein 16-Bit-Ergebnisregister im Binärformat.

(Die 2 LSBs D1 und D0 sind immer 0).

Das Ergebnis der Erfassung ist immer ein 14-Bit-Wert (UNSIGNED).

Die Genauigkeit des Ergebnisses hängt von der gewählten Wandlungszeit ab.

Die Temperatur kann aus den Ausgabedaten berechnet werden mit:

$$\text{Temperatur (}^{\circ}\text{C)} = (\text{code}[15..0] / 2^{16}) * 165^{\circ}\text{C} - 40^{\circ}\text{C}$$

Da hierdurch ein UNSIGNED Wert für den Istwert und über den „FSP030_SetValue_A“ ebenfalls ein UNSIGNED Wert für den Sollwert vorliegen, werden diese beiden Werte vor dem Analgen an die Eingänge um jeweils ein Bit nach rechts (/2) geschoben. Dadurch ist sichergestellt, dass das führende Bit immer „0“ ist.

Bei der Umrechnung der berechneten Differenz entfallen die -40°C, da die Differenz selbst nur eine Temperatur darstellt, die frei ist von jedwedem Bezug zu den Eingangswerten.

'Die Differenz muss aber um eins nach links (*2) geschoben werden, da die Eingangswerte zuvor um eins nach rechts (/2) geschoben wurden.

$$\text{Temperatur (}^{\circ}\text{C)} = (\text{code}[15..0] / 2^{16}) * 165^{\circ}\text{C} * 2$$

code[15..0] entspricht in diesem Fall [23..8] des vorzeichenbehafteten Ausgabewerts.

Name	FSP077_Controller_PI_Output
Adresse	0x4D_H/77_D/0x3444_{ASCII}
Tiefe	9 Byte / 72 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann der P, I und PI-Anteil des PI Regler gelesen werden. Der Regler dient der Temperaturregung des ADCs.

[71..48] Controller_P_Output (20 Bit)

[47..24] Controller_I_Output (20 Bit)

[23..0] Controller_PI_Output (20 Bit)

Name	FSP078_SerialComDt2SendSel
Adresse	0x4E_H/78_D/0x3445_{ASCII}
Tiefe	1 Byte / 8 Bit
I/O	lesen/schreiben
Reset	0x00 _H

Mit diesem FSP lassen sich die Daten auswählen, die über die beiden LWL Kanäle verschickt werden.

Das ErrGenCfg FSP für SerialCom_1 ist „FSP079_SerialCom_1_ErrorGenCfg“.

Das ErrGenCfg FSP für SerialCom_2 ist „FSP081_SerialCom_2_ErrorGenCfg“.

Der Status für SerialCom_1 ist „FSP080_SerialCom_1_Status“.

Der Status für SerialCom_2 ist „FSP082_SerialCom_2_Status“.

[7..4] Quellenwahl für i_Data2EncoderMux2-Ausgangssignal (4 Bit)
NewDt2_8b10bEncoder_2, Dt2_8b10bEncoder_2[23..0]

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	NewCntDt2MHz, CntDt2MHz[23..0]
0x2	NewCntDt1MHz, CntDt1MHz[23..0]
0x3	NewCntDt500KHz, CntDt500KHz[23..0]
0x4	NewCntDt250KHz, CntDt250KHz[23..0]
0x5	ADC_NewCorrectedValue, ADC_CorrectedValue[23..0]
0x6	PayloadRx_1_valid, PayloadRx_1_data[23..0]
0x7	
..	n.u.
0xF	

[3..0] Quellenwahl für i_Data2EncoderMux1-Ausgangssignal (4 Bit)
NewDt2_8b10bEncoder_1, Dt2_8b10bEncoder_1[23..0]

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	NewCntDt2MHz, CntDt2MHz[23..0]
0x2	NewCntDt1MHz, CntDt1MHz[23..0]
0x3	NewCntDt500KHz, CntDt500KHz[23..0]
0x4	NewCntDt250KHz, CntDt250KHz[23..0]
0x5	ADC_NewCorrectedValue, ADC_CorrectedValue[23..0]
0x6	PayloadRx_0_valid, PayloadRx_0_data[23..0]
0x7	
..	n.u.
0xF	

Name	FSP079_SerialCom_1_ErrorGenCfg
Adresse	0x4F_H/79_D/0x3446_{ASCII}
Tiefe	5 Byte / 40 Bit
I/O	lesen/schreiben
Reset	0x00_06_000000 _H

Konfiguriert die Fehlerinjektion, die Bitrate und setzt die Fehler für den LWL-Kanal **1** zurück.

Das Äquivalent für SerialCom_2 ist „FSP081_SerialCom_2_ErrorGenCfg“.

Der Status für SerialCom_1 ist im „FSP080_SerialCom_1_Status“.

Den Multiplexer für die zu sendenden Daten bedient „FSP078_SerialComDt2SendSel“.

[39..36] Selected received payload

[3..0]	Ausgang des Multiplexers
0x0	Disabled
0x1	PayloadRx_1_data[23..0]
0x2	PayloadRx_2_data[23..0]
0x3 .. 0xF	n.u.

[35..33] n.u.

[32] Reset errors

[31..24] Bit length: it defines how many 120MHz clock cycles are used to code 1 bit information.

[23..21] n.u.

[20] Error injection enable(active high)

[19..17] n.u.

[16] Change symbol flag

[15..13] n.u.

[12..8] Symbol mask

[7..0] Symbol number to corrupt

Name	FSP080_SerialCom_1_Status
Adresse	0x50_H/80_D/0x3530_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Sammelt die Alarme für den LWL-Kanal 1.

Das Äquivalent für SerialCom_2 ist „FSP082_SerialCom_2_Status“.

Das „Debugging“ der SerialCom_1 erfolgt im „FSP079_SerialCom_1_ErrorGenCfg“.

Den Multiplexer für die zu sendenden Daten bedient „FSP078_SerialComDt2SendSel“.

[23..17] n.u.

[16] PayloadAlarm_1

[15] LostConnection_1

[14] IllSymbComFailed_1

[13] PckgDis_1

[12] ParityComFailed_1

[11..8] ParityErrorCnt_1[3..0]

[7..0] CntError_1[7..0]

Name	FSP081_SerialCom_2_ErrorGenCfg
Adresse	0x51_H/81_D/0x3531_{ASCII}
Tiefe	5 Byte / 40 Bit
I/O	lesen/schreiben
Reset	0x00_06_000000 _H

Konfiguriert die Fehlerinjektion, die Bitrate und setzt die Fehler für den LWL-Kanal **1** zurück.

Das Äquivalent für SerialCom_2 ist „FSP079_SerialCom_1_ErrorGenCfg“.

Der Status für SerialCom_1 ist im „FSP080_SerialCom_1_Status“.

Den Multiplexer für die zu sendenden Daten bedient „FSP078_SerialComDt2SendSel“.

[39..36] Selected received payload

[3..0]	Ausgang des Multiplexers
0x0	Disabled
0x1	PayloadRx_1_data[23..0]
0x2	PayloadRx_2_data[23..0]
0x3 .. 0xF	n.u.

[35..33] n.u.

[32] Reset errors

[31..24] Bit length: it defines how many 120MHz clock cycles are used to code 1 bit information.

[23..21] n.u.

[20] Error injection enable(active high)

[19..17] n.u.

[16] Change symbol flag

[15..13] n.u.

[12..8] Symbol mask

[7..0] Symbol number to corrupt

Name	FSP082_SerialCom_2_Status
Adresse	0x52_H/82_D/0x3532_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Sammelt die Alarme für den LWL-Kanal **2**.

Das Äquivalent für SerialCom_1 ist „FSP080_SerialCom_1_Status“.

Das „Debugging“ der SerialCom_2 erfolgt im „FSP079_SerialCom_1_ErrorGenCfg“.

Den Multiplexer für die zu sendenden Daten bedient „FSP078_SerialComDt2SendSel“.

[23..17] n.u.

[16] PayloadAlarm_2

[15] LostConnection_2

[14] IllSymbComFailed_2

[13] PckgDis_2

[12] ParityComFailed_2

[11..8] ParityErrorCnt_2[3..0]

[7..0] CntError_2[7..0]

Name	FSP083_Redundant_DCCT_Thrs
Adresse	0x53_H/83_D/0x3533_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen/schreiben
Reset	0x000000_000000 _H

[47..24] Comparator**OFF**Threshold[23..0] – inst_Hysteresis_Comparator_DCCT_Redundant

[23..0] Comparator**ON**Threshold[23..0] – inst_Hysteresis_Comparator_DCCT_Redundant

Name	FSP084_Redundant_DCCT_Cfg
Adresse	0x54_H/84_D/0x3534_{ASCII}
Tiefe	1 Byte / 8 Bit
I/O	lesen/schreiben
Reset	0x00 _H

- [7] Datenauswahl für [inst_Difference_Calculator_Redundant_Check]
[0] = PayloadRx_1_data[23..0]
[1] = PayloadRx_2_data[23..0]
- [6..1] n.u.
- [0] Enable – [inst_Hystersis_Comparator_DCCT_Redundant]

Name	FSP085_Redundant_DCCT_Status
Adresse	0x55_H/85_D/0x3535_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

[31..25] n.u. (immer 0)

[24] RedundantDCCT (Ausgang [inst_Hystersis_Comparator_DCCT_Redundant])

[23..0] DeltaAct[23..0] (Ausgang [inst_Difference_Calculator_Redundant_Check])