

FSPs des ADC DAC IO Ver. II Moduls (ab FW 7.0.x)

Version vom: Freitag, 17. März 2023, 14:53:00

Inhaltsverzeichnis

1.	Änderungsliste	1
2.	FSPs ADC DAC IO (Ver.II) Modul (FG660.46x)	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}	
	FSP006_InterlocksArrivalSequence	10
	0x06 _H /6 _D /0x3036 _{AS}	
	FSP009_ModuleSerialNumber	11
	0x09 _H /9 _D /0x3039 _{ASCII}	
	FSP010_ModuleCommands	12
	0x0A _H /10 _D /0x3041 _{ASCII}	
	FSP011_ModuleInterlocksMask_n	13
	0x0B _H /11 _D /0x3042 _{ASCII}	
	FSP012_USIConfig	15
	0x0C _H /12 _D /0x3043 _{ASCII}	
	FSP013_PeripheralConfig	16
	0x0D _H /13 _D /0x3044 _{ASCII}	
	FSP020_ActualValue_A	17
	0x14 _H /20 _D /0x3134 _{ASCII}	
	FSP030_SetValue_A	18
	0x1E _H /30 _D /0x3145 _{ASCII}	
	FSP031_SetValue_B	19
	0x1F _H /31 _D /0x3146 _{ASCII}	
	FSP033_SetValue_C	20
	0x20 _H /33 _D /0x3230 _{ASCII}	
	FSP034_SetValue_D	21
	0x21 _H /34 _D /0x3231 _{ASCII}	
	FSP040_RemoteUpdateStatus	22
	0x28 _H /40 _D /0x3238 _{ASCII}	
	FSP041_RemoteUpdateCommands	23
	0x29 _H /41 _D /0x3239 _{ASCII}	
	FSP042_RemoteUpdateData	24
	0x2A _H /42 _D /0x3241 _{ASCII}	
	FSP043_FSP_Image_Generator	26
	0x2B _H /43 _D /0x3242 _{ASCII}	
	FSP044_FSP_Image_Status	27
	0x2C _H /44 _D /0x3243 _{ASCII}	
	FSP045_AlteraRemoteUpdateCmd	28
	0x2D _H /45 _D /0x3244 _{ASCII}	
	FSP046_AlteraRemoteUpdateStatus	29
	0x2E _H /46 _D /0x3245 _{ASCII}	
	FSP050_ModuleSupplyValues	30
	0x32 _H /50 _D /0x3332 _{ASCII}	
	FSP051_ModulePotiValues	31
	0x33 _H /51 _D /0x3333 _{ASCII}	
	FSP052_ModuleComparatorValues	32
	0x34 _H /52 _D /0x3334 _{ASCII}	
	FSP053_ModuleTemperatures	33
	0x35 _H /53 _D /0x3335 _{ASCII}	
	FSP054_ModuleTemperaturesComparativeThresholds	34
	0x36 _H /54 _D /0x3336 _{ASCII}	
	FSP058_ParameterChecksumValue	35
	0x3A _H /58 _D /0x3341 _{ASCII}	
	FSP059_ParameterChecksumValueCalculated	36
	0x3B _H /59 _D /0x3342 _{ASCII}	
	FSP060_ADC_Configuration	37
	0x3C _H /60 _D /0x3343 _{ASCII}	
	FSP061_ADCCalibrationGAIN	38
	0x3D _H /61 _D /0x3344 _{ASCII}	
	FSP062_ADCCalibrationOFFSET	39

0x3E _H /62 _D /0x3345 _{ASCII}	
FSP063_ADCFAVResult	40
0x3F _H /63 _D /0x3346 _{ASCII}	
FSP064_InterlockSelectMUX	41
0x40 _H /64 _D /0x3440 _{ASCII}	
FSP066_ValCounter	42
0x42 _H /66 _D /0x3432 _{ASCII}	
FSP070_Controller_1_InputSourceSelectionMultiplexer	43
0x46 _H /70 _D /0x3436 _{ASCII}	
FSP071_Controller_1_DifferenceCalculatorMultiplier	45
0x47 _H /71 _D /0x3731 _{ASCII}	
FSP072_Controller_1_PI_Settings	46
0x48 _H /72 _D /0x3448 _{ASCII}	
FSP073_Controller1_Limits	47
0x49 _H /73 _D /0x3439 _{ASCII}	
FSP074_Controller_1_ComparatorLimits	48
0x4A _H /74 _D /0x3441 _{ASCII}	
FSP077_Controller_1_AdderSourceSelectionMultiplexer	49
0x4D _H /77 _D /0x3444 _{ASCII}	
FSP078_Controller_1_AdderLimits	50
0x4E _H /78 _D /0x3445 _{ASCII}	
FSP079_Controller_1_Values	51
0x4F _H /79 _D /0x3446 _{ASCII}	
FSP080_Controller_2_InputSourceSelectionMultiplexer	52
0x50 _H /80 _D /0x3530 _{ASCII}	
FSP081_Controller_2_DifferenceCalculatorMultiplier	54
0x51 _H /81 _D /0x3531 _{ASCII}	
FSP082_Controller_2_PI_Settings	55
0x52 _H /82 _D /0x3542 _{ASCII}	
FSP083_Controller_2_Limits	56
0x53 _H /83 _D /0x3533 _{ASCII}	
FSP084_Controller_2_ComparatorLimits	57
0x54 _H /84 _D /0x3544 _{ASCII}	
FSP089_Controller_2_Values	58
0x59 _H /89 _D /0x3539 _{ASCII}	
FSP090_ICJX_Port_Configuration	59
0x5A _H /90 _D /0x3541 _{ASCII}	
FSP091_ICJX_RD_Data	60
0x5B _H /90 _D /0x3542 _{ASCII}	
FSP100_DACxSourceSelectionMultiplexer	61
0x64 _H /100 _D /0x3634 _{ASCII}	
FSP101_DACGain_Offset	64
0x65 _H /101 _D /0x3635 _{ASCII}	
FSP109_PWM_PLLPhaseShift_ReConfig	65
0x6D _H /109 _D /0x3644 _{ASCII}	
FSP110_PWM_Config	66
0x6E _H /110 _D /0x3645 _{ASCII}	
FSP111_PWM_Limits	67
0x6F _H /111 _D /0x3646 _{ASCII}	
FSP112_PWM_InhibitValue	68
0x70 _H /112 _D /0x3730 _{ASCII}	
FSP113_HighSpeed_ReturnChannel_SourceSelectionMux	69
0x71 _H /113 _D /0x3731 _{ASCII}	
FSP114_ADCAverageValSel	74
0x72 _H /114 _D /0x3752 _{ASCII}	
FSP120_intFunctionGenerator	75
0x78 _H /120 _D /0x3738 _{ASCII}	
FSP121_Ext_ADC_ComparatorThresholds	77
0x79 _H /121 _D /0x3739 _{ASCII}	
FSP122_Ext_ADC_Data	78
0x7A _H /122 _D /0x3741 _{ASCII}	

1. Änderungsliste

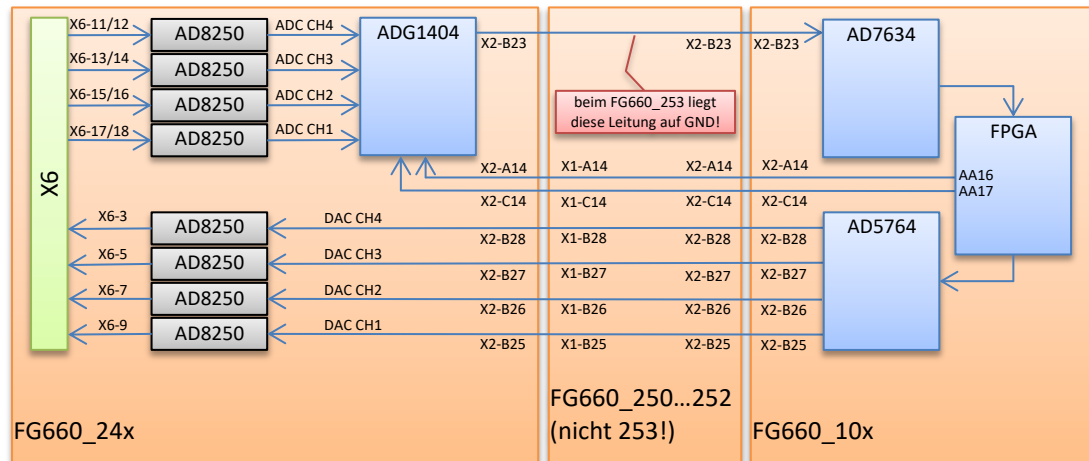
Datum	Name	Kommentar
31.03.2022	D. Schupp	Dokument erstellt aus ACU-FSP mUSIc TFT
23.09.2022	D. Schupp	Reduziert auf Ver.II FW 7.0.x

2. FSPs ADC DAC IO (Ver.II) Modul (FG660.46x)

Dieses Kapitel behandelt modulspezifische FSPs des ADC_DAC_IO Moduls Ver.II ab FG660.460.

3. Erweiterungen

In Verbindung mit weiteren Modulen ist folgende Erweiterung möglich.



Die bis zu 4 sequentiell gemultiplexten und vom 18-Bit ADC gewandelten ADC Messwerte (ADC_FAV_Result_CH1...CH4) können anschließend wahlweise über weitere Multiplexer als HighSpeed-Messwert über USI verschickt werden (ADC_FAV_Result_A...D).

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

- [23] wenn ,1', Hauptschütz geschlossen
- [22] n.u., immer ,1'
- [21] n.u., immer ,1'
- [20] wenn ,1', Regler über Switching Operations 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
→ siehe FSP010_ModuleCommands
- [11..9] Reserviert
Reserviert für zukünftige Anwendungen
- [8] USIIsHighSpeed
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. Temperatur 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	8 Byte / 64 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

- [63...59] n.u., immer ,1'
- [58] USI Slave 2 HighSpeed Abbruch (Fehler in der FW, muss ,3' lauten)
- [57] USI Slave 2 HighSpeed Abbruch
- [56] USI Slave 1 HighSpeed Abbruch
- [55] wenn ,0', Komparator-Schwellen des FPGA internen Komparators vom Kanals D des generischen 13 Bit ADC überschritten
- [54] wenn ,0', Komparator-Schwellen des FPGA internen Komparators vom Kanals D des generischen 13 Bit ADC unterschritten
- [53] wenn ,0', Komparator-Schwellen des FPGA internen Komparators vom Kanals C des generischen 13 Bit ADC überschritten
- [52] wenn ,0', Komparator-Schwellen des FPGA internen Komparators vom Kanals C des generischen 13 Bit ADC unterschritten
- [51] wenn ,0', Komparator-Schwellen des FPGA internen Komparators vom Kanals B des generischen 13 Bit ADC überschritten
- [50] wenn ,0', Komparator-Schwellen des FPGA internen Komparators vom Kanals B des generischen 13 Bit ADC unterschritten
- [49] wenn ,0', Komparator-Schwellen des FPGA internen Komparators vom Kanals A des generischen 13 Bit ADC überschritten
- [48] wenn ,0', Komparator-Schwellen des FPGA internen Komparators vom Kanals A des generischen 13 Bit ADC unterschritten
- [47] wenn ,0', Komparator-Schwellen des HW-Komparators des 18-Bit Haupt ADC überschritten
- [46] wenn ,0', Komparator-Schwellen des HW-Komparators des 18-Bit Haupt ADC unterschritten
- [45] n.u., immer ,1'
- [44] wenn ,0', Hauptschütz nicht geschlossen (festgelegt via PCA'Main contactot interlock bit)
- [43] wenn ,0', Interlock kommend vom ICJX, Nibble 2, Bit 3
- [42] wenn ,0', Interlock kommend vom ICJX, Nibble 2, Bit 2
- [41] wenn ,0', Interlock kommend vom ICJX, Nibble 2, Bit 1
- [40] wenn ,0', Interlock kommend vom ICJX, Nibble 2, Bit 0

- [39] wenn ,0', Interlock kommend vom ICJX, Nibble 1, Bit 3
- [38] wenn ,0', Interlock kommend vom ICJX, Nibble 1, Bit 2
- [37] wenn ,0', Interlock kommend vom ICJX, Nibble 1, Bit 1
- [36] wenn ,0', Interlock kommend vom ICJX, Nibble 1, Bit 0
- [35] wenn ,0', Interlock kommend vom ICJX, Nibble 0, Bit 3
- [34] wenn ,0', Interlock kommend vom ICJX, Nibble 0, Bit 2
- [22] wenn ,0', Interlock kommend vom ICJX, Nibble 0, Bit 1
- [32] wenn ,0', Interlock kommend vom ICJX, Nibble 0, Bit 0

Aktuell anstehende Interlocks

- [31...27] n.u., immer ,1'
- [26] USI Slave 2 HighSpeed Abbruch (Fehler in der FW, muss ,3' lauten)
- [25] USI Slave 2 HighSpeed Abbruch
- [24] USI Slave 1 HighSpeed Abbruch
- [23] wenn ,0', pos. Schwellen des FPGA internen Komparators vom Kanals D des generischen 13 Bit ADC überschritten
- [22] wenn ,0', neg. Schwellen des FPGA internen Komparators vom Kanals D des generischen 13 Bit ADC unterschritten
- [21] wenn ,0', pos. Schwellen des FPGA internen Komparators vom Kanals C des generischen 13 Bit ADC überschritten
- [20] wenn ,0', neg. Schwellen des FPGA internen Komparators vom Kanals C des generischen 13 Bit ADC unterschritten
- [19] wenn ,0', pos. Schwellen des FPGA internen Komparators vom Kanals B des generischen 13 Bit ADC überschritten
- [18] wenn ,0', neg. Schwellen des FPGA internen Komparators vom Kanals B des generischen 13 Bit ADC unterschritten
- [17] wenn ,0', pos. Schwellen des FPGA internen Komparators vom Kanals A des generischen 13 Bit ADC überschritten
- [16] wenn ,0', neg. Schwellen des FPGA internen Komparators vom Kanals A des generischen 13 Bit ADC unterschritten
- [15] wenn ,0', pos. Schwellen des HW-Komparators des 18-Bit Haupt ADC überschritten
- [14] wenn ,0', neg. Schwellen des HW-Komparators des 18-Bit Haupt ADC unterschritten
- [13] n.u., immer ,1'
- [12] wenn ,0', Hauptschütz nicht geschlossen (festgelegt via PCA'Main contactot interlock bit)
- [11] wenn ,0', Interlock kommend vom ICJX, Nibble 2, Bit 3
- [10] wenn ,0', Interlock kommend vom ICJX, Nibble 2, Bit 2
- [9] wenn ,0', Interlock kommend vom ICJX, Nibble 2, Bit 1
- [8] wenn ,0', Interlock kommend vom ICJX, Nibble 2, Bit 0
- [7] wenn ,0', Interlock kommend vom ICJX, Nibble 1, Bit 3
- [6] wenn ,0', Interlock kommend vom ICJX, Nibble 1, Bit 2
- [5] wenn ,0', Interlock kommend vom ICJX, Nibble 1, Bit 1
- [4] wenn ,0', Interlock kommend vom ICJX, Nibble 1, Bit 0
- [3] wenn ,0', Interlock kommend vom ICJX, Nibble 0, Bit 3

-
- | | |
|-----|---|
| [2] | wenn ,0', Interlock kommend vom ICJX, Nibble 0, Bit 2 |
| [1] | wenn ,0', Interlock kommend vom ICJX, Nibble 0, Bit 1 |
| [0] | wenn ,0', Interlock kommend vom ICJX, Nibble 0, Bit 0 |

Name	FSP006_InterlocksArrivalSequence
Adresse	0x06_H/6_D/0x3036_{ASCII}
Tiefe	6 Byte / 48 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 5 zuerst auftretenden Interlocks dieses Moduls.

- [47..40] Das Interlockbit des zuerst aufgetretenen Interlocks (i)
- [39..32] Das Interlockbit des vor (i) aufgetretenen Interlocks (i-1)
- [31..24] Das Interlockbit des vor (i-1) aufgetretenen Interlocks (i-2)
- [23..16] Das Interlockbit des vor (i-2) aufgetretenen Interlocks (i-3)
- [15..8] Das Interlockbit des zuletzt aufgetretenen Interlocks (i-4)
- [7..0] Anzahl der erfassten Interlock im FSP006 insgesamt (0..5)

Name	FSP009_ModuleSerialNumber
Adresse	0x09_H/9_D/0x3039_{ASCII}
Tiefe	12 Byte / 96 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.

[95..48] Seriennummer eines evtl. angeschlossenen Erweiterungsmoduls

[47..0] ADC DAC IO 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	12 Byte / 96 Bit
I/O	Lesen / schreiben
Reset	0xFF002000_ FF002000_ FF002000H

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 Hauptschütz darf erst zugelassen werden, wenn die Reglerfreigabe erteilt ist. Andernfalls würde das anliegende Hauptschütz Interlock das Einschalten der SVE dauerhaft verhindern. Daher ist das zugehörige Bit auf ,1' zu setzen)

- [95..91] n.u., immer ,1'
- [90] wenn ,0', wird Interlock USI Slave 3 HighSpeed Abbruch dauerhaft erfasst
- [89] wenn ,0', wird Interlock USI Slave 2 HighSpeed Abbruch dauerhaft erfasst
- [88] wenn ,0', wird Interlock USI Slave 1 HighSpeed Abbruch dauerhaft erfasst
- [87..86] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals D des 13 Bit ADC überschritten [87], bzw. unterschritten [86] dauerhaft erfasst
- [85..84] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals C des 13 Bit ADC überschritten [85], bzw. unterschritten [84] dauerhaft erfasst
- [83..82] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals B des 13 Bit ADC überschritten [83], bzw. unterschritten [82] dauerhaft erfasst
- [81..80] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals A des 13 Bit ADC überschritten [81], bzw. unterschritten [80] dauerhaft erfasst
- [79..78] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des HW-Komparators des 18-Bit ADC überschritten [79], bzw. unterschritten [78] dauerhaft erfasst
- [76] wenn ,0', wird Interlock ,MainContactorClosedFailure_n' - Hauptschütz wurde nicht geschlossen dauerhaft erfasst
- [75..64] wenn ,0', werden zugehörige Interlock kommend vom ICJX dauerhaft erfasst

Gespeicherte Interlocks

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

- [63..59] n.u., immer ,1'
- [58] wenn ,0', wird Interlock USI Slave 3 HighSpeed Abbruch gespeichert
- [57] wenn ,0', wird Interlock USI Slave 2 HighSpeed Abbruch gespeichert
- [56] wenn ,0', wird Interlock USI Slave 1 HighSpeed Abbruch gespeichert

- [55..54] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals D des 13 Bit ADC überschritten [55], bzw. unterschritten [54] gespeichert
- [53..52] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals C des 13 Bit ADC überschritten [53], bzw. unterschritten [52] gespeichert
- [51..50] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals B des 13 Bit ADC überschritten [51], bzw. unterschritten [50] gespeichert
- [49..48] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals A des 13 Bit ADC überschritten [49], bzw. unterschritten [48] gespeichert
- [47..46] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des HW-Komparators des 18-Bit ADC überschritten [47], bzw. unterschritten [46] angezeigt
- [45] n.u., immer ,1'
- [44] wenn ,0', wird Interlock ,MainContactorClosedFailure_n' - Hauptschütz wurde nicht geschlossen gespeichert
- [43..32] wenn ,0', werden zugehörige Interlock kommend vom ICJX gespeichert

Aktuell anstehende Interlocks

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

- [31..27] n.u., immer ,1'
- [26] wenn ,0', wird Interlock USI Slave 3 HighSpeed Abbruch angezeigt
- [25] wenn ,0', wird Interlock USI Slave 2 HighSpeed Abbruch angezeigt
- [24] wenn ,0', wird Interlock USI Slave 1 HighSpeed Abbruch angezeigt
- [23..22] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals D des 13 Bit ADC überschritten [23], bzw. unterschritten [22] angezeigt
- [21..20] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals C des 13 Bit ADC überschritten [21], bzw. unterschritten [20] angezeigt
- [19..18] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals B des 13 Bit ADC überschritten [19], bzw. unterschritten [18] angezeigt
- [17..16] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des FPGA internen Komparators vom Kanals A des 13 Bit ADC überschritten [17]}, bzw. unterschritten [16] angezeigt
- [15..14] wenn ,0', werden Interlock bezüglich der Komparator-Schwellen des HW-Komparators des 18-Bit ADC überschritten [15], bzw. unterschritten [14] angezeigt
- [13] n.u., immer ,1'
- [12] wenn ,0', wird Interlock ,MainContactorClosedFailure_n' - Hauptschütz wurde nicht geschlossen angezeigt
- [11..0] wenn ,0', werden zugehörige Interlock kommend vom ICJX angezeigt

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 „alte“ im Modul errechnete Prüfsumme.
- [6..0] n.u.

Name	FSP020_ActualValue_A
Adresse	0x14_H/20_D/0x3134_{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 bis zu 16fach gemittelt (siehe: „FSP060_ADC_Configuration“)

[23..0] Vorzeichenbehafteter ADC Wert (20 Bit), bis zu 16fach gemittelt

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

Dieser FSP dient zur Übermittlung eines 16 Bit Sollwertes_A an diverse Ziele.

[23..0] 20 Bit Sollwert_A, wird über verschiedene Multiplexer wahlweise diversen Zielen als endgültiger Sollwert_A vorgegeben:

- Als Sollwert für den PI Regler 1 mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für den PI Regler 2 mittels des Sollwert-Multiplexer, welcher über „FSP080_Controller_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für den PI Regler 1 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für den PI Regler 2 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Summand 1 und/oder Summand 2 für den internen Addierer mittels des Addierer-Summand-1-2-Multiplexers, welcher über „FSP077_Controller_1_AdderSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für die PWM mittels des PWM-Multiplexer welcher über „FSP110_PWM_Config“ konfiguriert wird.
- Als DAC_1 Quelle mittels des DAC Quellen Multiplexers, der über „FSP100_DACxSourceSelectionMultiplexer“ konfiguriert wird.
- Als DAC_2 Quelle mittels des DAC Quellen Multiplexers, der über „FSP100_DACxSourceSelectionMultiplexer“ konfiguriert wird.
- Als DAC_3 Quelle mittels des DAC Quellen Multiplexers, der über „FSP100_DACxSourceSelectionMultiplexer“ konfiguriert wird.
- Als DAC_4 Quelle mittels des DAC Quellen Multiplexers, der über „FSP100_DACxSourceSelectionMultiplexer“ konfiguriert wird.
- Als Quelle für den ICJX im Freilaufmodus

Name	FSP031_SetValue_B
Adresse	0x1F_H/31_D/0x3146_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x000000 _H

Dieser FSP dient zur Übermittlung eines 16 Bit Sollwertes_B an diverse Ziele.

[23..0] 20 Bit Sollwert_B, wird über verschiedene Multiplexer wahlweise diversen Zielen als endgültiger Sollwert_B vorgegeben:

- Als Sollwert für den PI Regler 1 mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für den PI Regler 2 mittels des Sollwert-Multiplexer, welcher über „FSP080_Controller_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für den PI Regler 1 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für den PI Regler 2 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Summand 1 und/oder Summand 2 für den internen Addierer mittels des Addierer-Summand-1-2-Multiplexers, welcher über „FSP077_Controller_1_AdderSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für die PWM mittels des PWM-Multiplexer welcher über „FSP110_PWM_Config“ konfiguriert wird.
- Als DAC_1 Quelle mittels des DAC Quellen Multiplexers, der über „FSP100_DACxSourceSelectionMultiplexer“ konfiguriert wird.
- Als DAC_2 Quelle mittels des DAC Quellen Multiplexers, der über „FSP100_DACxSourceSelectionMultiplexer“ konfiguriert wird.
- Als DAC_3 Quelle mittels des DAC Quellen Multiplexers, der über „FSP100_DACxSourceSelectionMultiplexer“ konfiguriert wird.
- Als DAC_4 Quelle mittels des DAC Quellen Multiplexers, der über „FSP100_DACxSourceSelectionMultiplexer“ konfiguriert wird.

Name	FSP033_SetValue_C
Adresse	0x20_H/33_D/0x3230_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x000000 _H

Dieser FSP dient zur Übermittlung eines 16 Bit Sollwertes_C an diverse Ziele.

[23..0] 20 Bit Sollwert_C, wird über verschiedene Multiplexer wahlweise diversen Zielen als endgültiger Sollwert_C vorgegeben:

- Als Sollwert für den PI Regler 1 mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für den PI Regler 2 mittels des Sollwert-Multiplexer, welcher über „FSP080_Controller_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für den PI Regler 1 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für den PI Regler 2 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für die PWM mittels des PWM-Multiplexer welcher über „FSP110_PWM_Configkonfiguriert wird.

Name	FSP034_SetValue_D
Adresse	0x21_H/34_D/0x3231_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x000000 _H

Dieser FSP dient zur Übermittlung eines 16 Bit Sollwertes_D an diverse Ziele.

[23..0] 20 Bit Sollwert_D, wird über verschiedene Multiplexer wahlweise diversen Zielen als endgültiger Sollwert_D vorgegeben:

- Als Sollwert für den PI Regler 1 mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für den PI Regler 2 mittels des Sollwert-Multiplexer, welcher über „FSP080_Controller_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für den PI Regler 1 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für den PI Regler 2 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für die PWM mittels des PWM-Multiplexer welcher über „FSP110_PWM_Configkonfiguriert wird.

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 abbrechen/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 löschtbar. 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	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	Reset:0x0100000_0_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)

[47..44] n.u.

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

[19..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] Watchdog Enable n.u.
- [40] Watchdog Enable
- [39..8] Watchdog 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] vorzeichenbehaftete VREF, 2,75 Volt (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	FSP051_ModulePotiValues
Adresse	0x33_H/51_D/0x3333_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

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

[31..16] vorzeichenbehaftete Poti-Spannung der positiven Schweller des 18 Bit ADC HW Komparators (13 Bit)

[15..0] vorzeichenbehaftete Poti-Spannung der negativen Schweller des 18 Bit ADC HW Komparators (13 Bit)

Name	FSP052_ModuleComparatorValues
Adresse	0x34_H/52_D/0x3334_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

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

[31..16] vorzeichenbehaftete positive Spannung des 18 Bit ADC HW Komparators (13 Bit)

[15..0] vorzeichenbehaftete negativen Spannung des 18 Bit ADC HW Komparators (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

Liefert verschiedene Temperaturen des Moduls.

3 Sensoren vom Typ TC74 liefern Statustemperaturen des Moduls mit 8 Bit Auflösung (inkl. Vorzeichen). Diese Sensoren können für Alarmzwecke verwendet werden. Die Alarmschwellen sind zur Laufzeit änderbar und im Standard FSP 054 hinterlegt.

1 Sensor vom Typ ADT7410 liefert die Temperatur im ADC Blechgehäuse mit 16 Bit Auflösung (inkl. Vorzeichen)

[31..24] TC74_Control_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] TC74 Sensor 3: Temperatur Modul Mitte (8 Bit inkl. Vorzeichen)

[15..8] TC74 Sensor 2: Temperatur FPGA (8 Bit inkl. Vorzeichen)

[7..0] TC74 Sensor 1: Temperatur DC-DC Wandler (8 Bit inkl. Vorzeichen)

Name	FSP054_ModuleTemperaturesComparativeThresholds
Adresse	0x36_H/54_D/0x3336_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	Lesen / schreiben
Reset	0x46_46_46 _H

Stellt die vorzeichenbehafteten Vergleichswerte zur Verfügung bei denen die TC74 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 70° Celsius (70_D = 46_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 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 „FSP059_ParameterChecksumValueCalculated“ 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	1 Byte / 8 Bit
I/O	Lesen / schreiben
Reset	0x00 _H

Repräsentiert verschiedene Konfigurationszustände des ADC Moduls.

[7..4] n.u.

[5] ADC_ExternalConStartSignal_Enable, wenn ,1' wird der ADC über den HighSpeedReceiver getriggert, andernfalls läuft er frei. D.h. jedes Mal, wenn vom Host ein neuer HighSpeed Wert am ADC DAC IO Module ankommt und erfolgreich ausdekodiert wurde, also im Modul zur Verfügung steht, startet eine neue ADC Messung.

[4] Remove Outliers, wenn ,1' werden der größte und der kleinste Wert bei der Mittelwertbildung ausgeblendet. Wenn ,0' werden alle Werte zur Mittelwertbildung (sofern diese nicht abgeschaltet ist) herangezogen.

[3..0] ADC Mittelwertbildung

[3..0]	Mittelwertbildung
0x0	Mittelwertbildung aus
0x1	2fach Mittelung
0x2	4fach Mittelung
0x3	8fach Mittelung
0x4	16fach Mittelung

Name	FSP061_ ADCCalibrationGAIN
Adresse	0x3D_H/61_D/0x3344_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x4010E4 _H

Wird als Hilfs-FSP zur Kalibrierung des ADC DAC IO Moduls verwendet. Enthält die Verstärkungskorrektur (Skalierungsfaktor) des 18 Bit ADC, der über PowerConfigAdvanced beim Kalibriervorgang ermittelt und dann nicht flüchtig im ADC DAC IO 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] Vorzeichenbehafteter 18 Bit Skalierungswert zur Korrektur der ADC Verstärkung

Name	FSP062_ADCCalibrationOFFSET
Adresse	0x3E_H/62_D/0x3345_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x000EC9 _H

Wird als Hilfs-FSP zur Kalibrierung des ADC DAC Moduls verwendet. Enthält die Offsetkorrektur des 18 Bit ADC, der über PowerConfigAdvanced beim Kalibriervorgang ermittelt und dann nicht flüchtig im ADC DAC IO 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] Vorzeichenbehaftete 18 Bit Offsetkorrektur des ADC

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 18 Bit ADC zurück (FAV: Floating Average Value). Der ADC läuft frei und erfasst permanent Werte. Diese werden bis zu 16fach gemittelt (siehe: „FSP060_ADC_Configuration“)

[23..0] Vorzeichenbehafteter ADC Wert (20 Bit), bis zu 16fach gemittelt

Name	FSP064_InterlockSelectMUX
Adresse	0x40_H/64_D/0x3440_{ASCII}
Tiefe	3 Byte /24 Bit
I/O	lesen / schreiben
Reset	0x00_0C_00 _H

Diese FSP legt fest welches korrespondierende Bit der anstehenden Interlocks eine Sonderfunktion auslöst.

Der dabei zu selektierende Bit-Bus ist 32 Bit breit (gezählt werden die Bits von 0..31) und ist wie folgt belegt:

[InterlockBits(31..0)]

InterlockBits entspricht dabei der Interlockverteilung aus „FSP004_ModuleInterlocks“.

[23..22] n.u.

[20..16] Bitposition, an der das Hauptschutz-Interlock zu finden ist.
 Das Hauptschutz wird über einen elektrischen Schaltausgang bedient.
 Das Hauptschutz-Interlock ist im abgeschalteten Zustand des Gerätes immer aktiv, da das offene Hauptschutz automatisch einen Fehler signalisiert. Soll das Gerät nun eingeschaltet werden, muss das Hauptschutz-Interlock zunächst über die
 „FSP011_ModuleInterlocksMask_n“ als zugelassenes Interlocks nach der ReglerFreigabe definiert werden. Dadurch ist der zugehörige Interlockeingang bis zur Regler-Freigabe deaktiviert. Außerdem muss die Bitposition in diesem FSP festgelegt werden, damit die Rückmeldung des Hauptschützes über den zugehörigen Interlockeingang im ADC DAC IO Modul korrekt verarbeitet werden kann. Andernfalls wird das ADC DAC IO Modul keine Regler-Freigabe erteilen.

Nach dem Reset ist Interlockbit[12] ausgewählt.

[15..0] n.u.

Name	FSP066_ValCounter
Adresse	0x42_H/66_D/0x3432_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen / schreiben
Reset	0x00_00_00_00 _H

Gibt Zeiten vor, die das ADC DAC IO für die Ein-/Ausschaltzyklen benötigt

- [31..24] ,Val_CutOffTime_s' = Wertigkeit * 1 Sekunde, gibt die Zeit an die vergeht, bis die Wiedereinschaltssperre des Gerätes freigegeben wird nachdem das Gerät ausgeschaltet wurde
- [23..16] ,Val_Timer3_RunTime_in_s' = Wertigkeit * 1 Sekunde, gibt die Zeit an die vergeht, bis die Regler-Freigabe erfolgt nachdem der elektrische Schaltkontakt 3 eingeschaltet wurde
- [15..8] ,Val_Timer2_RunTime_in_s' = Wertigkeit * 1 Sekunden, gibt die Zeit an die vergeht, bis der elektrische Schaltkontakt 3 eingeschaltet wird nachdem der elektrische Schaltkontakt 2 eingeschaltet wurde
- [7..0] ,Val_Timer1_RunTime_in_s' = Wertigkeit * 1 Sekunde, gibt die Zeit an die vergeht, bis der elektrische Schaltkontakt 2 eingeschaltet wird nachdem der elektrische Schaltkontakt 1 eingeschaltet wurde

Name	FSP070_Controller_1_InputSourceSelectionMultiplexer
Adresse	0x46_H/70_D/0x3436_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x00_0_0 _H

Repräsentiert die Einstellungen der PI Regler 1 Eingangsmultiplexer für dessen Soll- und Istwert-Quellen.

[15..10] n.u.

[9] wenn '1' => Regler 1 Quellenwahl für Istwert-Quellen-Multiplexer-Ausgangssignal invertiert

[8] wenn '1' => Regler 1 Quellenwahl für Sollwert-Quellen-Multiplexer-Ausgangssignal invertiert

[7..4] Regler 1 Quellenwahl für das Istwert-Quellen-Multiplexer-Ausgangssignal (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A, bestimmt durch „FSP030_SetValue_A“
0x2	Sollwert B, bestimmt durch „FSP031_SetValue_B“
0x3	Sollwert C, bestimmt durch „FSP033_SetValue_C“
0x4	Sollwert D, bestimmt durch „FSP034_SetValue_D“
0x5	intFuncGenOutput
0x6	ADC_FAV_Result
0x7	HighSpeedPort_ReceiveData[31..12]
0x8	Ausgang des PI_Controller 2, Regler 2, P-Anteil
0x9	Ausgang des PI_Controller 2, Regler 2, I-Anteil
0xA	Ausgang des PI_Controller 2, Regler 2, PI-Anteil
0xB	Ausgang des PI_Controller 1, Regler 1, P-Anteil
0xC	Ausgang des PI_Controller 1, Regler 1, I-Anteil
0xD	Ausgang des PI_Controller 1, Regler 1, PI-Anteil
0xE	n.u.
0xF	n.u.

[3..0] Regler 1 Quellenwahl für Sollwert-Quellen-Multiplexer-Ausgangssignal (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A, bestimmt durch „FSP030_SetValue_A“
0x2	Sollwert B, bestimmt durch „FSP031_SetValue_B“
0x3	Sollwert C, bestimmt durch „FSP033_SetValue_C“
0x4	Sollwert D, bestimmt durch „FSP034_SetValue_D“
0x5	intFuncGenOutput
0x6	ADC_FAV_Result
0x7	HighSpeedPort_ReceiveData[31..12]
0x8	Ausgang des PI_Controller 2, Regler 2, P-Anteil
0x9	Ausgang des PI_Controller 2, Regler 2, I-Anteil
0xA	Ausgang des PI_Controller 2, Regler 2, PI-Anteil
0xB	Ausgang des PI_Controller 1, Regler 1, P-Anteil
0xC	Ausgang des PI_Controller 1, Regler 1, I-Anteil
0xD	Ausgang des PI_Controller 1, Regler 1, PI-Anteil

0xE	n.u.
0xF	n.u.

Name	FSP071_Controller_1_DifferenceCalculatorMultiplier
Adresse	0x47_H/71_D/0x3731_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x0001 _H

Repräsentiert den Multiplikator für den Multiplikand Delta I der Regelabweichung des Regler 1.

Die Regelabweichung zwischen Soll- und Istwert sollte i.d.R. sehr klein sein. Das ist hinderlich sofern der Wert auf einem Anzeigeinstrument (z.B. ext. Oszilloskop) dargestellt werden soll. Aus diesem Grund befinden sich in jedem Regler Multiplikator-Stufen, die den Wert der Regelabweichung mit einem Multiplikator so hoch verstärken können, dass eine vernünftige Darstellung möglich ist.

Der Multiplikator versteht sich als vorzeichenbehafteter Integer zwischen $\pm 2^{10}$.

[15..11] n.u.

[10..0] Multiplikator (11 Bit) für den Differenzbildner Regler 1

Name	FSP072_Controller_1_PI_Settings
Adresse	0x48_H/72_D/0x3448_{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 Regler 1 gesetzt werden.

[103..96] Controller1_PI_Control

[103..97] n.u.

[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 Regler 1.

[63..32] Controller1_P2_Part (32 Bit), repräsentiert den 2. P Anteil des PI Regler 1.

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

Name	FSP073_Controller1_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 Regler 1 festgelegt werden.

[47..24] Controller1_MaxVal (20 Bit), repräsentiert den oberen (maximalen) Grenzwert des PI Regler 1

[23..0] Controller1_MinVal (20 Bit), repräsentiert den unteren (minimalen) Grenzwert des PI Regler 1

Name	FSP074_Controller_1_ComparatorLimits
Adresse	0x4A_H/74_D/0x3441_{ASCII}
Tiefe	12 Byte / 96 Bit
I/O	lesen / schreiben
Reset	0x000000_000000_000000_000000 _H

Repräsentiert die Bereichsgrenzen in denen der I Anteil, bzw. P2 bei der Regelung berücksichtigt werden soll.

- [96..72] Controller1_I_Part_ComparatorOFFThreshold (20 Bit)
- [71..48] Controller1_I_Part_ComparatorONThreshold (20 Bit)
- [47..24] Controller1_P2_Part_ComparatorOFFThreshold (20 Bit)
- [23..0] Controller1_P2_Part_ComparatorONThreshold (20 Bit)

Name	FSP077_Controller_1_AdderSourceSelectionMultiplexer
Adresse	0x4D_H/77_D/0x3444_{ASCII}
Tiefe	2 Byte / 12 Bit
I/O	lesen / schreiben
Reset	0x00_0_0 _H

Repräsentiert die Einstellungen des Addierer Eingangs-Multiplexer für dessen Summanden 1 und 2.

[15..10] n.u.

[9] '1' Ausgangswert des Multiplexer [AdderSummand_2] wird invertiert

[8] '1' Ausgangswert des Multiplexer [AdderSummand_1] wird invertiert

[7..4] Quellenwahl für Multiplexes Ausgangssignal [AdderSummand_2] (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A, bestimmt durch „FSP030_SetValue_A“
0x2	Sollwert B, bestimmt durch „FSP031_SetValue_B“
0x3	HighSpeedPort_ReceivedData[31..12]
0x4	0
0x5	Ausgang des PI_Controller, Regler 1, P-Anteil
0x6	Ausgang des PI_Controller, Regler 1, I-Anteil
0x7	Ausgang des PI_Controller, Regler 1, PI-Anteil
0x8	Ausgang des PI_Controller, Regler 2, P-Anteil
0x9	Ausgang des PI_Controller, Regler 2, I-Anteil
0xA	Ausgang des PI_Controller, Regler 2, PI-Anteil
0xB	n.u.
0xC	n.u.
0xD	n.u.
0xE	n.u.
0xF	n.u.

[3..0] Quellenwahl für Multiplexer Ausgangssignal [AdderSummand_1] (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A, bestimmt durch „FSP030_SetValue_A“
0x2	Sollwert B, bestimmt durch „FSP031_SetValue_B“
0x3	HighSpeedPort_ReceivedData[31..12]
0x4	0
0x5	Ausgang des PI_Controller, Regler 1, P-Anteil
0x6	Ausgang des PI_Controller, Regler 1, I-Anteil
0x7	Ausgang des PI_Controller, Regler 1, PI-Anteil
0x8	Ausgang des PI_Controller, Regler 2, P-Anteil
0x9	Ausgang des PI_Controller, Regler 2, I-Anteil
0xA	Ausgang des PI_Controller, Regler 2, PI-Anteil
0xB	n.u.
0xC	n.u.
0xD	n.u.
0xE	n.u.
0xF	n.u.

Name	FSP078_Controller_1_AdderLimits
Adresse	0x4E_H/78_D/0x3445_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Über diesen FSP werden die Bereichsgrenzen des Addierers festgelegt.

[47..24] Adder_MaxVal (20 Bit), repräsentiert den oberen (maximalen) Grenzwert des Addierer.

[23..0] Adder_MinVal (20 Bit), repräsentiert den unteren (minimalen) Grenzwert des Addierer.

Name	FSP079_Controller_1_Values
Adresse	0x4F_H/79_D/0x3446_{ASCII}
Tiefe	27 Byte / 216 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Liefert Ausgangswerte des Regler 1 zurück

[215..192]	Controller_1_Adder_1_SumOut (20 Bit)
[191..168]	Controller_1_Summand_2_MultiplexerOut (20 Bit)
[167..144]	Controller_1_Summand_1_MultiplexerOut (20 Bit)
[143..120]	Controller_1_Pi_Part_Output (20 Bit)
[119..96]	Controller_1_I_Part_Output (20 Bit)
[95..72]	Controller_1_P_Part_Output (20 Bit)
[71..48]	Controller_1_Deviation (20 Bit)
[47..24]	Controller_1_ActValueMuxOut (20 Bit)
[23..0]	Controller_1_SetValueMuxOut (20 Bit)

Name	FSP080_Controller_2_InputSourceSelectionMultiplexer
Adresse	0x50_H/80_D/0x3530_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x00_0_0 _H

Repräsentiert die Einstellungen der PI Regler 2 Eingangsmultiplexer für dessen Soll- und Istwert-Quellen.

[15..10] n.u.

[9] wenn '1' Regler 2 Quellenwahl für Istwert-Quellen-Multiplexer-Ausgangssignal invertiert

[8] wenn ,1' Regler 2 Quellenwahl für Sollwert-Quellen-Multiplexer-Ausgangssignal invertiert

[7..4] Regler 2 Quellenwahl für das Istwert-Quellen-Multiplexer-Ausgangssignal (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A, bestimmt durch „FSP030_SetValue_A“
0x2	Sollwert B, bestimmt durch „FSP031_SetValue_B“
0x3	Sollwert C, bestimmt durch „FSP033_SetValue_C“
0x4	Sollwert D, bestimmt durch „FSP034_SetValue_D“
0x5	intFuncGenOutput
0x6	ADC_FAV_Result
0x7	HighSpeedPort_ReceiveData[31..12]
0x8	Ausgang des PI_Controller 2, Regler 2, P-Anteil
0x9	Ausgang des PI_Controller 2, Regler 2, I-Anteil
0xA	Ausgang des PI_Controller 2, Regler 2, PI-Anteil
0xB	Ausgang des PI_Controller 1, Regler 1, P-Anteil
0xC	Ausgang des PI_Controller 1, Regler 1, I-Anteil
0xD	Ausgang des PI_Controller 1, Regler 1, PI-Anteil
0xE	n.u.
0xF	n.u.

[3..0] Regler 2 Quellenwahl für Sollwertquellen-Multiplexer-Ausgangssignal (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A, bestimmt durch „FSP030_SetValue_A“
0x2	Sollwert B, bestimmt durch „FSP031_SetValue_B“
0x3	Sollwert C, bestimmt durch „FSP033_SetValue_C“
0x4	Sollwert D, bestimmt durch „FSP034_SetValue_D“
0x5	intFuncGenOutput
0x6	ADC_FAV_Result
0x7	HighSpeedPort_ReceiveData[31..12]
0x8	Ausgang des PI_Controller 2, Regler 2, P-Anteil
0x9	Ausgang des PI_Controller 2, Regler 2, I-Anteil
0xA	Ausgang des PI_Controller 2, Regler 2, PI-Anteil
0xB	Ausgang des PI_Controller 1, Regler 1, P-Anteil
0xC	Ausgang des PI_Controller 1, Regler 1, I-Anteil
0xD	Ausgang des PI_Controller 1, Regler 1, PI-Anteil
0xE	n.u.

0xF	n.u.
-----	------

Name	FSP081_Controller_2_DifferenceCalculatorMultiplier
Adresse	0x51_H/81_D/0x3531_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x0001 _H

Repräsentiert den Multiplikator für den Multiplikand Delta I der Regelabweichung des Reglers 2.

Die Regelabweichung zwischen Soll- und Istwert sollte i.d.R. sehr klein sein. Das ist hinderlich sofern der Wert auf einem Anzeigeinstrument (z.B. ext. Oszilloskop) dargestellt werden soll. Aus diesem Grund befinden sich in jedem Regler Multiplikator-Stufen, die den Wert der Regelabweichung mit einem Multiplikator so hoch verstärken können, dass eine vernünftige Darstellung möglich ist.

Der Multiplikator versteht sich als signed Integer zwischen $\pm 2^{10}$.

[15..11] n.u.

[10..0] Multiplikator (11 Bit) für den Differenzbildner Regler 2

Name	FSP082_Controller_2_Pi_Settings
Adresse	0x52_H/82_D/0x3542_{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 Reglers 2 gesetzt werden.

[103..96] Controller2_Pi_Control

[103..97] n.u.

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

[95..64] Controller2_I_Part (32 Bit), repräsentiert den I Anteil des PI Regler 2.

[63..32] Controller2_P2_Part (32 Bit), repräsentiert den 2. P Anteil des PI Regler 2.

[31..0] Controller2_P1_Part (32 Bit), repräsentiert den 1. P Anteil des PI Regler 2.

Name	FSP083_Controller_2_Limits
Adresse	0x53_H/83_D/0x3533_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Über diesen FSP können die Bereichsgrenzen des PI Regler 2 festgelegt werden.

[47..24] Controller2_MaxVal (20 Bit), repräsentiert den oberen (maximalen) Grenzwert des PI Regler 2

[23..0] Controller2_MinVal (20 Bit), repräsentiert den unteren (minimalen) Grenzwert des PI Regler 2

Name	FSP084_Controller_2_ComparatorLimits
Adresse	0x54_H/84_D/0x3544_{ASCII}
Tiefe	12 Byte / 96 Bit
I/O	lesen / schreiben
Reset	0x000000_000000_000000_000000 _H

Repräsentiert die Bereichsgrenzen in denen der I Anteil, bzw. P2 bei der Regelung berücksichtigt werden soll.

- [96..72] Controller2_I_Part_ComparatorOFFThreshold (20 Bit)
- [71..48] Controller2_I_Part_ComparatorONThreshold (20 Bit)
- [47..24] Controller2_P2_Part_ComparatorOFFThreshold (20 Bit)
- [23..0] Controller2_P2_Part_ComparatorONThreshold (20 Bit)

Name	FSP089_Controller_2_Values
Adresse	0x59_H/89_D/0x3539_{ASCII}
Tiefe	18 Byte / 143 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Liefert Ausgangswerte des Reglers 2 zurück

[143..120]	Controller_2_Pi_Part_Output (20 Bit)
[119..96]	Controller_2_I_Part_Output (20 Bit)
[95..72]	Controller_2_P_Part_Output (20 Bit)
[71..48]	Controller_2_Deviation (20 Bit)
[47..24]	Controller_2_ActValueMuxOut (20 Bit)
[23..0]	Controller_2_SetValueMuxOut (20 Bit)

Name	FSP090_ICJX_Port_Configuration
Adresse	0x5A_H/90_D/0x3541_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x000000 _H

- [23] RD/nWr, wenn ,1' Daten lesen, wenn ,0' Daten schreiben
- [22] Anzahl der zu lesenden oder schreibenden Bytes,
wenn ,0' = 1 Byte,
wenn ,1' = 2 Byte
- [21] Mode,
wenn ,1' - der ICJX wird nur über die FSPs konfiguriert und betrieben
wenn ,0' - der Chip läuft selbstständig als Interlockkontroller und Schaltsystem.
- [20..16] 5 Bit Register Startadresse
- [15..0] 16 Bit Register Daten beim schreiben; beim Lesen sind diese Bit ,don't care'

Name	FSP091_ICJX_RD_Data
Adresse	0x5B_H/90_D/0x3542_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) _H

Die aus dem ICJX gelesenen Daten

Ist in „FSP090_ICJX_Port_Configuration[21] = Mode 0 gewählt läuft der ICJX selbstständig als Interlock-kontroller und Schaltsystem

Name	FSP100_DACxSourceSelectionMultiplexer
Adresse	0x64_H/100_D/0x3634_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x00_0_1_1_1 _H

Repräsentiert die Einstellungen der DAC Eingangsmultiplexer.

[23..20] n.u.

[19] '1' Ausgangswert des Multiplexers [DAC_4_Input] → X8 wird invertiert

[18] '1' Ausgangswert des Multiplexers [DAC_3_Input] → X7 wird invertiert

[17] '1' Ausgangswert des Multiplexers [DAC_2_Input] → X6 wird invertiert

[16] '1' Ausgangswert des Multiplexers [DAC_1_Input] → X5 wird invertiert

[15..12] Quellenwahl für Multiplexer Ausgangssignal [DAC_4_Input] → X8

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	FSP030_SetValue_A[23..4] (20 Bit)
0x2	FSP031_SetValue_B[23..4] (20 Bit)
0x3	Sollwert des Regler 1, bestimmt durch „FSP070_Controller_1_InputSourceSelectionMultiplexer“ (20 Bit)
0x4	Sollwert des Regler 2, bestimmt durch „FSP080_Controller_2_InputSourceSelectionMultiplexer“ (20 Bit)
0x5	ADC_FAV_Result, Ausgabewert des 18 Bit HW ADC nach dessen Kalibrierung und evtl. Mittellung (20 Bit)
0x6	ADC_FAV_Result_CH2, aktivierbar über „FSP114_ADCAverageValSel“ (20 Bit)
0x7	ADC_FAV_Result_CH4, aktivierbar über „FSP114_ADCAverageValSel“ (20 Bit)
0x8	EXT_ADC_VAL_C, Istwert von Kanals A des 13 Bit ADC (13 Bit)
0x9	EXT_ADC_VAL_D, Istwert von Kanals B des 13 Bit ADC (13 Bit)
0xA	intFuncGenOutput (20 Bit)
0xB	Controller_2_P_Part_Output
0xC	Controller_2_I_Part_Ouput
0xD	Controller_2_PI_Part_Ouput
0xE	HighSpeedport_Slave_1_Received[31..12] (20 Bit)
0xF	HighSpeedport_Slave_2_Received[31..12] (20 Bit)

[11..8] Quellenwahl für Multiplexer Ausgangssignal [DAC_3_Input] → X7

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	FSP030_SetValue_A[23..4] (20 Bit)
0x2	FSP031_SetValue_B[23..4] (20 Bit)
0x3	Sollwert des Regler 1, bestimmt durch „FSP070_Controller_1_InputSourceSelectionMultiplexer“ (20 Bit)
0x4	Sollwert des Regler 2, bestimmt durch „FSP080_Controller_2_InputSourceSelectionMultiplexer“ (20 Bit)
0x5	ADC_FAV_Result, Ausgabewert des 18 Bit HW ADC nach dessen Kalibrierung und evtl. Mittellung (20 Bit)
0x6	ADC_FAV_Result_CH2, aktivierbar über „FSP114_ADCAverageValSel“ (20 Bit)
0x7	ADC_FAV_Result_CH4, aktivierbar über „FSP114_ADCAverageValSel“ (20 Bit)

0x8	EXT_ADC_VAL_C, Istwert von Kanals A des 13 Bit ADC (13 Bit)
0x9	EXT_ADC_VAL_D, Istwert von Kanals B des 13 Bit ADC (13 Bit)
0xA	intFuncGenOutput (20 Bit)
0xB	Controller_2_P_Part_Output
0xC	Controller_2_I_Part_Ouput
0xD	Controller_2_PI_Part_Ouput
0xE	HighSpeedport_Slave_1_Received[31..12] (20 Bit)
0xF	HighSpeedport_Slave_2_Received[31..12] (20 Bit)

[7..4] Quellenwahl für Multiplexer Ausgangssignal [DAC_2_Input] → X6

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	FSP030_SetValue_A[23..4] (20 Bit)
0x2	FSP031_SetValue_B[23..4] (20 Bit)
0x3	Sollwert des Regler 1, bestimmt durch „FSP070_Controller_1_InputSourceSelectionMultiplexer“ (20 Bit)
0x4	Sollwert des Regler 2, bestimmt durch „FSP080_Controller_2_InputSourceSelectionMultiplexer“ (20 Bit)
0x5	ADC_FAV_Result, Ausgabewert des 18 Bit HW ADC nach dessen Kalibrierung und evtl. Mittellung (20 Bit)
0x6	ADC_FAV_Result_CH1, aktivierbar über „FSP114_ADCAverageValSel“ (20 Bit)
0x7	ADC_FAV_Result_CH3, aktivierbar über „FSP114_ADCAverageValSel“ (20 Bit)
0x8	EXT_ADC_VAL_A, Istwert von Kanals A des 13 Bit ADC (13 Bit)
0x9	EXT_ADC_VAL_B, Istwert von Kanals B des 13 Bit ADC (13 Bit)
0xA	intFuncGenOutput (20 Bit)
0xB	Controller_1_P_Part_Output
0xC	Controller_1_I_Part_Ouput
0xD	Controller_1_PI_Part_Ouput
0xE	HighSpeedport_Slave_1_Received[31..12] (20 Bit)
0xF	HighSpeedport_Slave_2_Received[31..12] (20 Bit)

[3..0] Quellenwahl für Multiplexer Ausgangssignal [DAC_1_Input] → X5

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	FSP030_SetValue_A[23..4] (20 Bit)
0x2	FSP031_SetValue_B[23..4] (20 Bit)
0x3	Sollwert des Regler 1, bestimmt durch „FSP070_Controller_1_InputSourceSelectionMultiplexer“ (20 Bit)
0x4	Sollwert des Regler 2, bestimmt durch „FSP080_Controller_2_InputSourceSelectionMultiplexer“ (20 Bit)
0x5	ADC_FAV_Result, Ausgabewert des 18 Bit HW ADC nach dessen Kalibrierung und evtl. Mittellung (20 Bit)
0x6	ADC_FAV_Result_CH1, aktivierbar über „FSP114_ADCAverageValSel“ (20 Bit)
0x7	ADC_FAV_Result_CH3, aktivierbar über „FSP114_ADCAverageValSel“ (20 Bit)
0x8	EXT_ADC_VAL_A, Istwert von Kanals A des 13 Bit ADC (13 Bit)
0x9	EXT_ADC_VAL_B, Istwert von Kanals B des 13 Bit ADC (13 Bit)
0xA	intFuncGenOutput (20 Bit)
0xB	Controller_1_P_Part_Output
0xC	Controller_1_I_Part_Ouput

0xD	Controller_1_PI_Part_Ouput
0xE	HighSpeedport_Slave_1_Received[31..12] (20 Bit)
0xF	HighSpeedport_Slave_2_Received[31..12] (20 Bit)

Name	FSP101_DACGain_Offset
Adresse	0x65_H/101_D/0x3635_{ASCII}
Tiefe	24 Byte / 192 Bit
I/O	lesen / schreiben
Reset	0x4000 0000 4000 0000 4000 0000 4000 0000 _H

Repräsentiert den Verstärkungsfaktor und Offsetwert des 4-fachen DAC für die vorne am ADC_DAC_IO-Modul vorhandenen DAC Diagnosebuchsen.

- [191..168] DAC 4 (Lemo X9) Gain, vorzeichenbehafteter 19 Bit Wert
- [167..144] DAC 4 (Lemo X9) Offset, vorzeichenbehafteter 18 Bit Wert
- [143..120] DAC 3 (Lemo X7) Gain, vorzeichenbehafteter 19 Bit Wert
- [119..96] DAC 3 (Lemo X7) Offset, vorzeichenbehafteter 18 Bit Wert
- [95..72] DAC 2 (Lemo X6) Gain, vorzeichenbehafteter 19 Bit Wert
- [71..48] DAC 2 (Lemo X6) Offset, vorzeichenbehafteter 18 Bit Wert
- [47..24] DAC 1 (Lemo X5) Gain, vorzeichenbehafteter 19 Bit Wert
- [23..0] DAC 1 (Lemo X5) Offset, vorzeichenbehafteter 18 Bit Wert

Name	FSP109_PWM_PLLPhaseShift_ReConfig
Adresse	0x6D_H/109_D/0x3644_{ASCII}
Tiefe	8 Byte / 64 Bit
I/O	schreiben/lesen
Reset	0x0000000000000000 _H

Enthält Daten für die re-konfigurable PWM.

[63..60] n.u.

[58][39..32][57][31..24]

PWM_PhSh_PLLReconf.MDataReconf (18 Bit)

[55..48][59][47..40]

PWM_PhSh_PLLReconf.NDataReconf (18 Bit)

[23..16][56][15..8]

PWM_PhSh_PLLReconf.CxDataReconf (17 Bit)

[7..5] n.u.

[4] wenn '1' => PWM_PhSh_PLLReconf.EnReconf

[3..1] n.u.

[0] wenn '1' => PWM_PhSh_PLLReconf.PhaseShiftEn

Name	FSP110_PWM_Config
Adresse	0x6E_H/110_D/0x3645_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	schreiben/lesen
Reset	0x00_0_0_0_0_0_0_0 _H

Dieses FSP konfiguriert die re-konfigurierbare PWM im Hinblick auf deren Sollwerte und Verhalten

[31..29] n.u.

[28] Wenn ,1': Ausgang des PWM Sollwert Multiplexer negieren

[27..24] PWM Sollwert Multiplexer Quellenwahl

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A, bestimmt durch „FSP030_SetValue_A“ (20 Bit)
0x2	Sollwert B, bestimmt durch „FSP031_SetValue_B“ (20 Bit)
0x3	Sollwert C, bestimmt durch „FSP033_SetValue_C“ (20 Bit)
0x4	Sollwert D, bestimmt durch „FSP034_SetValue_D“ (20 Bit)
0x5	intFuncGenOutput (20 Bit)
0x6	ADC_FAV_Result (20 Bit)
0x7	HighSpeedPort_ReceivedData[31..12] (20 Bit)
0x8	Ausgang des PI_Controller 2, Regler 1, P-Anteil
0x9	Ausgang des PI_Controller 2, Regler 1,I-Anteil
0xA	Ausgang des PI_Controller 2 , Regler 1,PI-Anteil
0xB	Ausgang des PI_Controller 1, Regler 2, P-Anteil
0xC	Ausgang des PI_Controller 1, Regler 2,I-Anteil
0xD	Ausgang des PI_Controller 1, Regler 2,PI-Anteil
0xE	Ausgang des Addierers von Controller 1
0xF	0

[23] PulseInhibitor

[22] n.u.

[21..20] SawToothPeriod (2 Bit)

[19..16] PWM Mode (4 Bit)

[15..12] V5OnInhnCntValue (4 Bit)

[11..8] Totzeit für V6 (4 Bit)

[7..0] Totzeit für V5 (4 Bit)

[3..0] Totzeit für V1..V4 (4 Bit)

Name	FSP111_PWM_Limits
Adresse	0x6F_H/111_D/0x3646_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	schreiben/lesen
Reset	0x0000_0000 _H

Beinhaltet die Bereichsgrenzen der PWM

[31..16] PWM_MaxValue (14 Bit)

[15..0] PWM_MinValue (14 Bit)

Name	FSP112_PWM_InhibitValue
Adresse	0x70_H/112_D/0x3730_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	schreiben/lesen
Reset	0x000000 _H

Beinhaltet das PWM_InhibitValue (Impulsperre der PWM)

[23..0] PWM_InhibitValue (14 Bit)

Name	FSP113_HighSpeed_ReturnChannel_SourceSelectionMux
Adresse	0x71_H/113_D/0x3731_{ASCII}
Tiefe	5 Byte / 40 Bit
I/O	schreiben/lesen
Reset	0x0_0_0_0_0_0_0_0_0_0 _H

ADC_DAC_IO_Ver.II verfügt über insgesamt 3 USI Anschlüsse.

Ermöglicht das Setzen der HighSpeed Rückkanal-Multiplexer

Der HighSpeed Rückkanal verfügt über insgesamt maximal 34 Bits, die wie folgt belegt sind.

frei, nicht benutzt, immer ,0' [33..32]

HighSpeed_ReturnChannel_Higher_14Bits_Mux [31..18] definierbar

HighSpeed_ReturnChannel_Lower_14Bits_Mux [17..4] definierbar

SwitchingOperationsStatus [3..0] fix

[39..36] HighSpeed_ReturnChannel_USI_C-3_Lower_18Bits

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_A[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x2	ADC_FAV_Result_B[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x3	ADC_FAV_Result_C[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x4	ADC_FAV_Result_D[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x5	Controller_1_SetValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x6	Controller_1_ActValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x7	Controller_1_SetValueDeviation[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x8	Controller_1_PI_Part_Output[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x9	Controller_1_Adder_1_SumOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xA	Controller_2_SetValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xB	Controller_2_ActValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xC	Controller_2_PI_Part_Output[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xD	HighSpeedPort_Slave_1_ReceivedData[17..0], GND_BUS[1..0]
0xE	HighSpeedPort_Slave_2_ReceivedData[17..0], GND_BUS[1..0]
0xF	HighSpeedPort_Slave_3_ReceivedData[17..0], GND_BUS[1..0]

[35..32] HighSpeed_ReturnChannel_USI_C-3_Higher_14Bits

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_A[19..0]
0x2	ADC_FAV_Result_B[19..0]
0x3	ADC_FAV_Result_C[19..0]
0x4	ADC_FAV_Result_D[19..0]
0x5	Controller_1_SetValueMuxOut[19..0]

0x6	Controller_1_ActValueMuxOut[19..0]
0x7	Controller_1_SetValueDeviation[19..0]
0x8	Controller_1_PI_Part_Output[19..0]
0x9	Controller_1_Adder_1_SumOut[19..0]
0xA	Controller_2_SetValueMuxOut[19..0]
0xB	Controller_2_ActValueMuxOut[19..0]
0xC	Controller_2_PI_Part_Output[19..0]
0xD	HighSpeedPort_Slave_1_ReceivedData[31..18], GND_BUS[5..0]
0xE	HighSpeedPort_Slave_2_ReceivedData[31..18], GND_BUS[5..0]
0xF	HighSpeedPort_Slave_3_ReceivedData[31..18], GND_BUS[5..0]

[31..28] HighSpeed_ReturnChannel_USI_B-2_Lower_18Bits

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_A[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x2	ADC_FAV_Result_B[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x3	ADC_FAV_Result_C[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x4	ADC_FAV_Result_D[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x5	Controller_1_SetValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x6	Controller_1_ActValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x7	Controller_1_SetValueDeviation[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x8	Controller_1_PI_Part_Output[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x9	Controller_1_Adder_1_SumOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xA	Controller_2_SetValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xB	Controller_2_ActValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xC	Controller_2_PI_Part_Output[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xD	HighSpeedPort_Slave_1_ReceivedData[17..0], GND_BUS[1..0]
0xE	HighSpeedPort_Slave_2_ReceivedData[17..0], GND_BUS[1..0]
0xF	HighSpeedPort_Slave_3_ReceivedData[17..0], GND_BUS[1..0]

[27..24] HighSpeed_ReturnChannel_USI_B-2_Higher_14Bits

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_A[19..0]
0x2	ADC_FAV_Result_B[19..0]
0x3	ADC_FAV_Result_C[19..0]
0x4	ADC_FAV_Result_D[19..0]
0x5	Controller_1_SetValueMuxOut[19..0]
0x6	Controller_1_ActValueMuxOut[19..0]
0x7	Controller_1_SetValueDeviation[19..0]
0x8	Controller_1_PI_Part_Output[19..0]
0x9	Controller_1_Adder_1_SumOut[19..0]

	0xA	Controller_2_SetValueMuxOut[19..0]
	0xB	Controller_2_ActValueMuxOut[19..0]
	0xC	Controller_2_PI_Part_Output[19..0]
	0xD	HighSpeedPort_Slave_1_ReceivedData[31..18], GND_BUS[5..0]
	0xE	HighSpeedPort_Slave_2_ReceivedData[31..18], GND_BUS[5..0]
	0xF	HighSpeedPort_Slave_3_ReceivedData[31..18], GND_BUS[5..0]

[23..20] HighSpeed_ReturnChannel_USI_A-1_Lower_18Bits

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_A[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x2	ADC_FAV_Result_B[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x3	ADC_FAV_Result_C[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x4	ADC_FAV_Result_D[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x5	Controller_1_SetValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x6	Controller_1_ActValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x7	Controller_1_SetValueDeviation[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x8	Controller_1_PI_Part_Output[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0x9	Controller_1_Adder_1_SumOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xA	Controller_2_SetValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xB	Controller_2_ActValueMuxOut[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xC	Controller_2_PI_Part_Output[19..6], SwitchingOperationsStatus[3..0], GND_BUS[1..0]
0xD	HighSpeedPort_Slave_1_ReceivedData[17..0], GND_BUS[1..0]
0xE	HighSpeedPort_Slave_2_ReceivedData[17..0], GND_BUS[1..0]
0xF	HighSpeedPort_Slave_3_ReceivedData[17..0], GND_BUS[1..0]

[19..16] HighSpeed_ReturnChannel_USI_A-1_Higher_14Bits

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_A[19..0]
0x2	ADC_FAV_Result_B[19..0]
0x3	ADC_FAV_Result_C[19..0]
0x4	ADC_FAV_Result_D[19..0]
0x5	Controller_1_SetValueMuxOut[19..0]
0x6	Controller_1_ActValueMuxOut[19..0]
0x7	Controller_1_SetValueDeviation[19..0]
0x8	Controller_1_PI_Part_Output[19..0]
0x9	Controller_1_Adder_1_SumOut[19..0]
0xA	Controller_2_SetValueMuxOut[19..0]
0xB	Controller_2_ActValueMuxOut[19..0]
0xC	Controller_2_PI_Part_Output[19..0]
0xD	HighSpeedPort_Slave_1_ReceivedData[31..18], GND_BUS[5..0]

0xE	HighSpeedPort_Slave_2_ReceivedData[31..18], GND_BUS[5..0]
0xF	HighSpeedPort_Slave_3_ReceivedData[31..18], GND_BUS[5..0]

[15..12] ADC_FAV_Result_D
Trifft ein Vorauswahl bzgl. des „ADC_FAV_Result_D“ Wertes

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_CH1[19..0]
0x2	ADC_FAV_Result_CH2[19..0]
0x3	ADC_FAV_Result_CH3[19..0]
0x4	ADC_FAV_Result_CH4[19..0]
0x5	ADC_FAV_Result_CH1[5..0], GND_BUS[13..0]
0x6	ADC_FAV_Result_CH2[5..0], GND_BUS[13..0]
0x7	ADC_FAV_Result_CH3[5..0], GND_BUS[13..0]
0x8	ADC_FAV_Result_CH4[5..0], GND_BUS[13..0]
0x9	ADC_FAV_Result[19..0]
0xA	EXT_ADC_VAL_A[15..0], GND_BUS[3..0]
0xB	EXT_ADC_VAL_B[15..0], GND_BUS[3..0]
0xC	EXT_ADC_VAL_C[15..0], GND_BUS[3..0]
0xD	EXT_ADC_VAL_D[15..0], GND_BUS[3..0]
0xE	GND_BUS[19..0]
0xF	GND_BUS[19..0]

[11..8] ADC_FAV_Result_C
Trifft ein Vorauswahl bzgl. des „ADC_FAV_Result_C“ Wertes

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_CH1[19..0]
0x2	ADC_FAV_Result_CH2[19..0]
0x3	ADC_FAV_Result_CH3[19..0]
0x4	ADC_FAV_Result_CH4[19..0]
0x5	ADC_FAV_Result_CH1[5..0], GND_BUS[13..0]
0x6	ADC_FAV_Result_CH2[5..0], GND_BUS[13..0]
0x7	ADC_FAV_Result_CH3[5..0], GND_BUS[13..0]
0x8	ADC_FAV_Result_CH4[5..0], GND_BUS[13..0]
0x9	ADC_FAV_Result[19..0]
0xA	EXT_ADC_VAL_A[15..0], GND_BUS[3..0]
0xB	EXT_ADC_VAL_B[15..0], GND_BUS[3..0]
0xC	EXT_ADC_VAL_C[15..0], GND_BUS[3..0]
0xD	EXT_ADC_VAL_D[15..0], GND_BUS[3..0]
0xE	GND_BUS[19..0]
0xF	GND_BUS[19..0]

[7..4] ADC_FAV_Result_B
Trifft ein Vorauswahl bzgl. des „ADC_FAV_Result_B“ Wertes

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_CH1[19..0]
0x2	ADC_FAV_Result_CH2[19..0]

0x3	ADC_FAV_Result_CH3[19..0]
0x4	ADC_FAV_Result_CH4[19..0]
0x5	ADC_FAV_Result_CH1[5..0], GND_BUS[13..0]
0x6	ADC_FAV_Result_CH2[5..0], GND_BUS[13..0]
0x7	ADC_FAV_Result_CH3[5..0], GND_BUS[13..0]
0x8	ADC_FAV_Result_CH4[5..0], GND_BUS[13..0]
0x9	ADC_FAV_Result[19..0]
0xA	EXT_ADC_VAL_A[15..0], GND_BUS[3..0]
0xB	EXT_ADC_VAL_B[15..0], GND_BUS[3..0]
0xC	EXT_ADC_VAL_C[15..0], GND_BUS[3..0]
0xD	EXT_ADC_VAL_D[15..0], GND_BUS[3..0]
0xE	GND_BUS[19..0]
0xF	GND_BUS[19..0]

[3..0]

ADC_FAV_Result_A

Trifft ein Vorauswahl bzgl. des „ADC_FAV_Result_A“ Wertes

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	ADC_FAV_Result_CH1[19..0]
0x2	ADC_FAV_Result_CH2[19..0]
0x3	ADC_FAV_Result_CH3[19..0]
0x4	ADC_FAV_Result_CH4[19..0]
0x5	ADC_FAV_Result_CH1[5..0], GND_BUS[13..0]
0x6	ADC_FAV_Result_CH2[5..0], GND_BUS[13..0]
0x7	ADC_FAV_Result_CH3[5..0], GND_BUS[13..0]
0x8	ADC_FAV_Result_CH4[5..0], GND_BUS[13..0]
0x9	ADC_FAV_Result[19..0]
0xA	EXT_ADC_VAL_A[15..0], GND_BUS[3..0]
0xB	EXT_ADC_VAL_B[15..0], GND_BUS[3..0]
0xC	EXT_ADC_VAL_C[15..0], GND_BUS[3..0]
0xD	EXT_ADC_VAL_D[15..0], GND_BUS[3..0]
0xE	GND_BUS[19..0]
0xF	GND_BUS[19..0]

Name	FSP114_ADCAverageValSel
Adresse	0x72_H/114_D/0x3752_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	schreiben/lesen
Reset	0x0001 _H

Dieses FSP konfiguriert das ‚ACU_ADC_AverageValSel‘ Module. Dessen Aufgabe ist es u.a. ein auf dem ‚FG660_24x_ADC_DAC_IO_Extentionmodule‘ befindlichen ADC1404 Multitplexer zu bedienen. Die Steuerausgänge des ADC DAC IO Moduls sind A14/C14 über die ‚FG660_25x_ADC_DAC_IO_Extensionmodule_Backplane‘.

Auf diesem ADC_DAC_IO_Extensionmodul befinden sich 4 AD8250 Instrumentenverstärker zur Erfassung von bis zu 4 Istwerten. Über den ADC1404 kann jeweils einer dieser Werte über die Backplane (Pin B23) auf den 18 Bit HW ADC des ADC DAC IO Moduls gelegt werden.

‚ACU_ADC_AverageValSel‘ wählt den ersten der zu selektierenden Kanäle (gesetzt mittels ‚ADC_Mux_Aconf‘) an und wartet bis der Eingangswert vom 18 Bit HW ADC erfasst, kalibriert und ggf. gemittelt wurde. Das Ergebnis wird, sofern ‚ADC_nrValToEscape‘ = 1 ist, als ‚ADC_FAV_Result_CHx‘ zur Verfügung gestellt. Ist ‚ADC_nrValToEscape‘ > 1, ird zunächst die Anzahl gewählter Werte verworfen, bevor ein Ergebnis an ‚ADC_FAV_Result_CHx‘ ausgegeben wird. Sind mehr als ein Kanal selektiert, wechselt nun der Multiplexer automatisch zum nächsten gewählten Kanal und der Prozess beginnt von neuem.

[15..12] n.u.

[11..8] ADC_Mux_Aconf, Bitcodiert

Die Kanäle können auch kombiniert selektiert werden, „0110“ würde z.B. Kanal 2 und 3 selektieren

[11..8]	Ausgang des Multiplexers
0000	Kein Kanal selektiert
0001	Kanal 1 selektiert
0010	Kanal 2 selektiert
0100	Kanal 3 selektiert
1000	Kanal 4 selektiert

[7..0] ADC_nrValToEscape

Gibt die Anzahl der zu ignorierenden/zu verwerfenden Wert an, nachdem der Muliplxer geschaltet wurde. Dieser Wert muss mindesten ‚0x01‘ sein, da die Anzahl der zu ignorierenden/zu verwerfenden Werte ‚ADC_NrValToEscape[7..0]-1‘ ist. Wird der Wert auf 1 gesetzt wird folglich KEIN Wert ignoriert/verwerfen.

Name	FSP120_intFunctionGenerator
Adresse	0x78_H/120_D/0x3738_{ASCII}
Tiefe	16 Byte / 128 Bit
I/O	lesen / schreiben
Reset	0x0_0_000000_000000_000000_00000_0_00000_0 _H

Repräsentiert die Parameter für den internen Funktionsgenerator

[127..125] ExtSyncConfig

[3..0]	ExtSyncConfig
0x0	Funktionsgenerator läuft frei, unabhängig ‚ExtSyncSource‘
0x1	Rampe aufwärts startet nur mit High-Signal an ‚ExtSyncSource‘, Rampe abwärts findet NICHT statt
0x2	Rampe aufwärts startet nur mit High-Signal an ‚ExtSyncSource‘, Rampe abwärts startet nur mit High-Signal an ‚ExtSyncSource‘
0x3	-
0x4	-
0x5	-
0x6	-
0x7	-
0x8	-
0x9	-
0xA	-
0xB	-
0xC	-
0xD	-
0xE	-
0xF	-

[126..123] ExtSyncSource

[3..0]	ExtSyncConfig
0x0	GND
0x1	VCC
0x2	intScopeTriggerEventDetected
0x3	Front LEMO In 1
0x4	Front LEMO In 2
0x5	GND
0x6	GND
0x7	GND
0x8	GND
0x9	GND
0xA	GND
0xB	GND
0xC	GND
0xD	GND
0xE	GND
0xF	GND

- [119..96] RampRiseTime (24 Bit), Wenn X"00_0000" pulst der Funktionsgenerator, andernfalls wird gerammt. RampTime repräsentiert dabei die Pausenzeit zwischen zwei Rampenpunkten während der Anstiegs-, bzw. Abstiegszeit der Rampe. Gibt die Zeit der Pausenintervalle zwischen den einzelnen Inkrementen zwischen dem ‚PauseValue‘ und dem ‚PulseValue‘ an. Das Pausenintervall entspricht „(RampRise * 10ns) + 10 ns“.
 Beispiel: PauseValue = -32768_D
 PulseValue = 32767_D
 RampTime = 1_D
 Es liegen 65535 Stützpunkte zwischen dem Minimal und Maximalwert.
 Es wird alle „(RampRise * 10ns) + 10ns“ ein neuer Stützpunkt ausgegeben, d.h. alle 20ns.
 Eine Rampe dauert also 65535 * ((1*10ns)+10ns)
- [95..89] n.u. (7 Bit)
- [88..72] PauseDuration (17 Bit), repräsentiert die Zeit die der Pausewert des internen Funktionsgenerators anstehen soll. Ist Bit [16] gesetzt gilt der Wert von Bit [15..0] in uSekunde. Ist Bit [16] nicht gesetzt gilt der Wert von Bit [15..0] in mSekunden.
- [71..65] n.u. (7 Bit)
- [64..48] PulseDuration (17 Bit), repräsentiert die Zeit die der Pulswert des internen Funktionsgenerators anstehen soll. Ist Bit [16] gesetzt gilt der Wert von Bit [15..0] in uSekunde. Ist Bit [16] nicht gesetzt gilt der Wert von Bit [15..0] in mSekunden.
- [47..28] PauseValue (20 Bit), repräsentiert den unteren (minimalen) Pausewert des internen Funktionsgenerators. Vorzeichenbehafteter Wert im Bereich zwischen -11..+11 Volt.
- [27..24] n.u. (4 Bit)
- [23..4] PulseValue (20 Bit), repräsentiert den oberen (maximalen) Pulswert des internen Funktionsgenerators. Vorzeichenbehafteter Wert im Bereich zwischen -11..+11 Volt.
- [3..0] n.u. (4 Bit)

Name	FSP121_Ext_ADC_ComparatorThresholds
Adresse	0x79_H/121_D/0x3739_{ASCII}
Tiefe	16 Byte / 128 Bit
I/O	lesen / schreiben
Reset	0x000000_000000_000000_000000 _H

Legt die Schwellwerte für die FPGA internen Komparatoren der 4 generischen ADC Kanäle A-D fest.

- [127..112] generischer Komparator A (X4, Pin 1, 2 UND/ODER X1-A25), obere Schwelle
- [111..96] generischer Komparator A (X4, Pin 1, 2 UND/ODER X1-A25), untere Schwelle
- [95..80] generischer Komparator B (X4, Pin 3, 4 UND/ODER X1-A26), obere Schwelle
- [79..64] generischer Komparator B (X4, Pin 3, 4 UND/ODER X1-A26), untere Schwelle
- [63..48] generischer Komparator C (X4, Pin 5, 6 UND/ODER X1-A27), obere Schwelle
- [47..32] generischer Komparator C (X4, Pin 5, 6 UND/ODER X1-A27), untere Schwelle
- [31..16] generischer Komparator D (X4, Pin 7, 8 UND/ODER X1-A28), obere Schwelle
- [15..0] generischer Komparator D (X4, Pin 7, 8 UND/ODER X1-A28), untere Schwelle

Name	FSP122_Ext_ADC_Data
Adresse	0x7A_H/122_D/0x3741_{ASCII}
Tiefe	8 Byte / 64 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Enthält die Messwerte der 4 generischen ADC Kanäle.

- [63..48] Messwert generischer ADC A/1 (X4, Pin 1, 2 UND/ODER X1-A25)
- [47..32] Messwert generischer ADC B/2 (X4, Pin 3, 4 UND/ODER X1-A26)
- [31..16] Messwert generischer ADC C/3 (X4, Pin 5, 6 UND/ODER X1-A27)
- [15..0] Messwert generischer ADC D/4 (X4, Pin 7, 8 UND/ODER X1-A28)