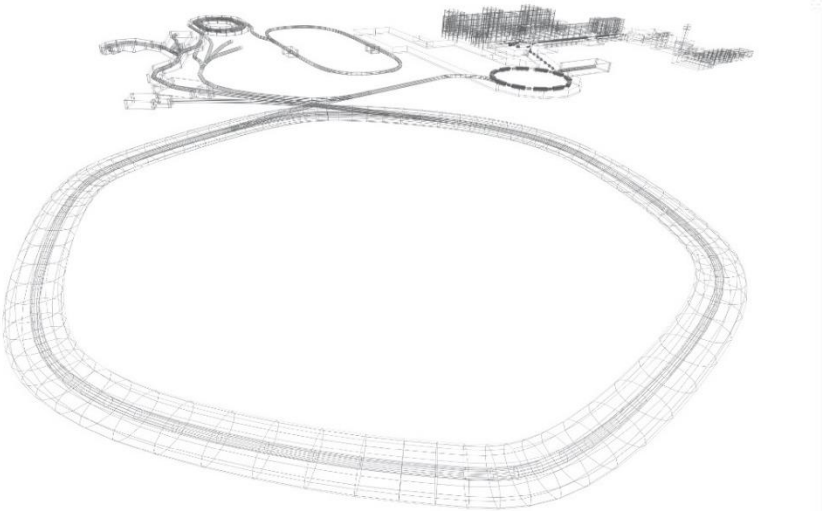
Elektrische
Energie
Technik

GSI Darmstadt



FSPs des
Water Interlock- and Contol Modul
(ab FW 7.0.x)

Version vom: Mittwoch, 4. Mai 2022, 12:09:00

Aktuelle Versionen immer unter:

https://wiki.gsi.de/EPS/ACUModuleDescriptions

Inhaltsverzeichnis

1. Änderungsliste 1
2. FSPs Wasser Interlock und Kontroll Modul II (FG660.151) 2

FSP001_ModuleStatus **3**
$0x01_H/1_D/0x3031_{ASCII}$

FSP002_ModuleWarnings **4**
$0x02_H/2_D/0x3032_{ASCII}$

FSP003_ModuleErrors **5**
$0x03_H/3_D/0x3033_{ASCII}$

FSP004_ModuleInterlocks **6**
$0x04_H/4_D/0x3034_{ASCII}$

FSP009_ModuleSerialNumber **9**
$0x09_H/9_D/0x3039_{ASCII}$

FSP010_ModuleCommands **10**
$0x0A_H/10_D/0x3041_{ASCII}$

FSP011_ModuleInterlocksMask_n **11**
$0x0B_H/11_D/0x3042_{ASCII}$

FSP012_USIConfig **12**
$0x0C_H/12_D/0x3043_{ASCII}$

FSP013_PeripheralConfig **13**
$0x0D_H/13_D/0x3044_{ASCII}$

FSP030_SetValue_A **14**
$0x1E_H/30_D/0x3145_{ASCII}$

FSP031_SetValue_B **15**
$0x1F_H/31_D/0x3146_{ASCII}$

FSP040_RemoteUpdateStatus **16**
$0x28_H/40_D/0x3238_{ASCII}$

FSP041_RemoteUpdateCommands **17**
$0x29_H/41_D/0x3239_{ASCII}$

FSP042_RemoteUpdateData **18**
$0x2A_H/42_D/0x3241_{ASCII}$

FSP045_AlteraRemoteUpdateCmd **20**
$0x2D_H/45_D/0x3244_{ASCII}$

FSP046_AlteraRemoteUpdateStatus **21**
$0x2E_H/46_D/0x3245_{ASCII}$

FSP053_ModuleTemperatures **22**
$0x35_H/53_D/0x3335_{ASCII}$

FSP054_ModuleTemperaturesComparationThresholds **23**
$0x36_H/54_D/0x3336_{ASCII}$

FSP058_ParameterChecksumValue **24**
$0x3A_H/58_D/0x3341_{ASCII}$

FSP059_ParameterChecksumValueCalculated **25**
$0x3B_H/59_D/0x3342_{ASCII}$

FSP063_WaterFlowThresholds **26**
$0x3F_H/63_D/0x3346_{ASCII}$

FSP064_InvertedWaterFlow_TimePeriodeBetweenTwoPulses_in_us **27**
$0x40_H/64_D/0x3440_{ASCII}$

FSP066_ADT7410_Temperatures **28**
$0x42_H/66_D/0x3442_{ASCII}$

FSP070_Controller_1_InputSourceSelectionMultiplexer **29**
$0x46_H/70_D/0x3436_{ASCII}$

FSP071_Controller_1_DifferenceCalculatorMultiplier **31**
$0x47_H/71_D/0x3731_{ASCII}$

FSP072_Controller_1_PI_Settings **32**
$0x48_H/72_D/0x3448_{ASCII}$

FSP073_Controller_1_Limits **33**
$0x49_H/73_D/0x3439_{ASCII}$

FSP074_Controller_1_ComparatorLimits **34**
$0x4A_H/74_D/0x3441_{ASCII}$

FSP077_Controller_1_AdderSourceSelectionMultiplexer **35**
$0x4D_H/77_D/0x3444_{ASCII}$

FSP078_Controller_1_AdderLimits **36**
$0x4E_H/78_D/0x3445_{ASCII}$

FSP079_Controller_1_Values **37**
$0x4F_H/79_D/0x3446_{ASCII}$

FSP080_Controller_2_InputSourceSelectionMultiplexer **38**

0x50$_H$/80$_D$/0x3530$_{ASCII}$
FSP081_Controller_2_DifferenceCalculatorMultiplier **40**
0x51$_H$/81$_D$/0x3531$_{ASCII}$
FSP082_Controller_2_PI_Settings **41**
0x52$_H$/82$_D$/0x3542$_{ASCII}$
FSP083_Controller_2_Limits **42**
0x53$_H$/83$_D$/0x3533$_{ASCII}$
FSP084_Controller_2_ComparatorLimits **43**
0x54$_H$/84$_D$/0x3544$_{ASCII}$
FSP089_Controller_2_Values **44**
0x59$_H$/89$_D$/0x3539$_{ASCII}$
FSP102_WaterValve_Config **45**
0x66$_H$/102$_D$/0x3636$_{ASCII}$
FSP103_WaterValve_MuxOut_AbsValue **46**
0x67$_H$/103$_D$/0x3637$_{ASCII}$
FSP113_HighSpeed_ReturnChannel_SourceSelectionMux **47**
0x71$_H$/113$_D$/0x3731$_{ASCII}$

1. Änderungsliste

Datum	Name	Kommentar
31.03.2022	D. Schupp	Dokument erstellt aus ACU-FSP mUSIc TFT

2. FSPs Wasser Interlock und Kontroll Modul II (FG660.151)

Dieses Kapitel behandelt modulspezifische FSPs des Wasser-Interlock und Kontroll Moduls Version 2 (WICM II).

Für ältere Module bitte ältere Versionen dieses Dokuments verwenden.

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

[23..16] Reserviert
Reserviert für zukünftige Anwendungen (immer ‚0')

[15..12] Modul Kommando

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

[11..9] Reserviert
Reserviert für zukünftige Anwendungen (immer ‚0')

[8] USIIsHighSpeed
wenn ‚1' ist USI im Highspeed Mode

[7..6] Reserviert
Reserviert für zukünftige Anwendungen (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.

[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	$0x(\text{siehe Beschreibung})_H$

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] des 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	$0x(\text{siehe Beschreibung})_H$

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] des FSP001_ModuleStatus = ‚0'.

[23..0] n.u., immer ‚1'

Name	**FSP004_ModuleInterlocks**
Adresse	**$0x04_H/4_D/0x3034_{ASCII}$**
Tiefe	6 Byte / 48 Bit
I/O	lesen
Reset	$0x(\text{siehe Beschreibung})_H$

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] des FSP001_ModuleStatus = ‚0'.

Nicht genutzte Interlockbits müssen ‚1' sein!

WICM:

Das Modul kann sowohl mit klassischen Schwebekegelwächtern, als auch Turbinenwächtern betrieben werden. Werden keine zu überwachenden Durchflussmengen für die Wasserwächter parametriert, sind die Anschlüsse der Wasserwächter automatisch für Schwebekegelwächter definiert. In diesem Fall sind immer 2 Interlockbits für einen Wasserdurchflusswächter zeitgleich zuständig. Bit 0 und 1 für Wächter 1, Bit 2 und 3 für Wächter 2 usw.

Werden Duchflussmengen für die Wasserwächter parametriert, geben die geraden Bit einen zu geringen, die ungeraden Bit einen zu hohen Durchlfuss an. Bit 0 = Durchlfluss Wächter 1 zu gering, Bit 1 = Durchfluss Wächter 1 zu hoch, Bit 2 = Durchfluss Wächter 2 zu gering, Bit 3 = Durchfluss Wächter 2 zu hoch usw.

Gespeicherte Interlocks

[47..44] n.u., immer ‚1'

[43] wenn '0', Wasserwächter 10 Durchfluss zu hoch (Turbinenwächter)

[42] wenn '0', Wasserwächter 10 Durchfluss zu gering (Turbinenwächter)

[43..42] wenn ‚0' Wasserwächter 10 Durchfluss-Fehler (Kegelwächter)

[41] wenn '0', Wasserwächter 9 Durchfluss zu hoch (Turbinenwächter)

[40] wenn '0', Wasserwächter 9 Durchfluss zu gering (Turbinenwächter)

[41..40] wenn ‚0' Wasserwächter 9 Durchfluss-Fehler (Kegelwächter)

[39] wenn '0', Wasserwächter 8 Durchfluss zu hoch (Turbinenwächter)

[38] wenn '0', Wasserwächter 8 Durchfluss zu gering (Turbinenwächter)

[39..38] wenn ‚0' Wasserwächter 9 Durchfluss-Fehler (Kegelwächter)

[37] wenn '0', Wasserwächter 7 Durchfluss zu hoch (Turbinenwächter)

[36] wenn '0', Wasserwächter 7 Durchfluss zu gering (Turbinenwächter)

[37..36] wenn ‚0' Wasserwächter 7 Durchfluss-Fehler (Kegelwächter)

[35] wenn '0', Wasserwächter 6 Durchfluss zu hoch (Turbinenwächter)

[34] wenn '0', Wasserwächter 6 Durchfluss zu gering (Turbinenwächter)

[35..34] wenn ‚0' Wasserwächter 6 Durchfluss-Fehler (Kegelwächter)

[33] wenn '0', Wasserwächter 5 Durchfluss zu hoch (Turbinenwächter)

[32] wenn '0', Wasserwächter 5 Durchfluss zu gering (Turbinenwächter)

[33..32] wenn ‚0‘ Wasserwächter 5 Durchfluss-Fehler (Kegelwächter)

[31] wenn '0', Wasserwächter 4 Durchfluss zu hoch (Turbinenwächter)

[30] wenn '0', Wasserwächter 4 Durchfluss zu gering (Turbinenwächter)

[31..30] wenn ‚0‘ Wasserwächter 4 Durchfluss-Fehler (Kegelwächter)

[29] wenn '0', Wasserwächter 3 Durchfluss zu hoch (Turbinenwächter)

[28] wenn '0', Wasserwächter 3 Durchfluss zu gering (Turbinenwächter)

[29..28] wenn ‚0‘ Wasserwächter 3 Durchfluss-Fehler (Kegelwächter)

[27] wenn '0', Wasserwächter 2 Durchfluss zu hoch (Turbinenwächter)

[26] wenn '0', Wasserwächter 2 Durchfluss zu gering (Turbinenwächter)

[27..26] wenn ‚0‘ Wasserwächter 2 Durchfluss-Fehler (Kegelwächter)

[25] wenn '0', Wasserwächter 1 Durchfluss zu hoch (Turbinenwächter)

[24] wenn '0', Wasserwächter 1 Durchfluss zu gering (Turbinenwächter)

[25..24] wenn ‚0‘ Wasserwächter 1 Durchfluss-Fehler (Kegelwächter)

Aktuell anstehende Interlocks

[23..20] n.u., immer ‚1’

[19] wenn '0', Wasserwächter 10 Durchfluss zu hoch (Turbinenwächter)

[18] wenn '0', Wasserwächter 10 Durchfluss zu gering (Turbinenwächter)

[19..18] wenn ‚0‘ Wasserwächter 10 Durchfluss-Fehler (Kegelwächter)

[17] wenn '0', Wasserwächter 9 Durchfluss zu hoch (Turbinenwächter)

[16] wenn '0', Wasserwächter 9 Durchfluss zu gering (Turbinenwächter)

[17..16] wenn ‚0‘ Wasserwächter 9 Durchfluss-Fehler (Kegelwächter)

[15] wenn '0', Wasserwächter 8 Durchfluss zu hoch (Turbinenwächter)

[14] wenn '0', Wasserwächter 8 Durchfluss zu gering (Turbinenwächter)

[15..14] wenn ‚0‘ Wasserwächter 8 Durchfluss-Fehler (Kegelwächter)

[13] wenn '0', Wasserwächter 7 Durchfluss zu hoch (Turbinenwächter)

[12] wenn '0', Wasserwächter 7 Durchfluss zu gering (Turbinenwächter)

[13..12] wenn ‚0‘ Wasserwächter 7 Durchfluss-Fehler (Kegelwächter)

[11] wenn '0', Wasserwächter 6 Durchfluss zu hoch (Turbinenwächter)

[10] wenn '0', Wasserwächter 6 Durchfluss zu gering (Turbinenwächter)

[11..10] wenn ‚0‘ Wasserwächter 6 Durchfluss-Fehler (Kegelwächter)

[9] wenn '0', Wasserwächter 5 Durchfluss zu hoch (Turbinenwächter)

[8] wenn '0', Wasserwächter 5 Durchfluss zu gering (Turbinenwächter)

[9..8] wenn ‚0‘ Wasserwächter 5 Durchfluss-Fehler (Kegelwächter)

[7] wenn '0', Wasserwächter 4 Durchfluss zu hoch (Turbinenwächter)

[6] wenn '0', Wasserwächter 4 Durchfluss zu gering (Turbinenwächter)

[7..6] wenn ‚0‘ Wasserwächter 4 Durchfluss-Fehler (Kegelwächter)

[5] wenn '0', Wasserwächter 3 Durchfluss zu hoch (Turbinenwächter)

[4] wenn '0', Wasserwächter 3 Durchfluss zu gering (Turbinenwächter)

[5..4] wenn ‚0‘ Wasserwächter 3 Durchfluss-Fehler (Kegelwächter)

[3] wenn '0', Wasserwächter 2 Durchfluss zu hoch (Turbinenwächter)

[2]	wenn '0', Wasserwächter 2 Durchfluss zu gering (Turbinenwächter)
[3..2]	wenn ‚0‘ Wasserwächter 2 Durchfluss-Fehler (Kegelwächter)
[1]	wenn '0', Wasserwächter 1 Durchfluss zu hoch (Turbinenwächter)
[0]	wenn '0', Wasserwächter 1 Durchfluss zu gering (Turbinenwächter)
[1..0]	wenn ‚0‘ Wasserwächter 1 Durchfluss-Fehler (Kegelwächter)

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

Der FSP enthält die Modul Serien Nummern

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

[47..0] WICM 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..0] n.u.

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

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

Für das WICM 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	6 Byte / 48 Bit
I/O	Lesen / schreiben
Reset	$0x000000_000000_H$

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

Gespeicherte Interlocks

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

[47..44] n.u., immer ‚1'

[43..24] wenn '0', werden zugehörige Wasser-Durchflussmesser-Interlocks gespeichert

Aktuell anstehende Interlocks

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

[23..20] n.u., immer ‚1'

[19..0] wenn '0', werden zugehörige Wasser-Durchflussmesser-Interlocks 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	**FSP030_SetValue_A**
Adresse	**$0x1E_H/30_D/0x3145_{ASCII}$**
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	$0x0000_H$

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

[15..0] 16 Bit Sollwert_A, wird über verschiedene Multiplexer wahlweise diversen Zielen als endgültiger Sollwert_A vorgegeben:

- Als Sollwert für PI Regler 1 mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für PI Regler 1 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für PI Regler 2 mittels des Sollwert-Multiplexer, welcher über „FSP080_Controller_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für interne PI Regler 2 mittels des Istwert-Multiplexer, welcher über „FSP080_Controller_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Summand 1 und/oder Summand 2 für den internen Regler 1 Addierer mittels des Regler_1_Summand_1_Multiplexers, welcher über „FSP077_Controller_1_AdderSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für den WaterValveAbsValue welcher über „FSP102_WaterValve_Config“ konfiguriert“ wird.

Name	**FSP031_SetValue_B**
Adresse	**$0x1F_H/31_D/0x3146_{ASCII}$**
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	$0x0000_H$

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

[15..0] 16 Bit Sollwert_B, wird über verschiedene Multiplexer wahlweise diversen Zielen als endgültiger Sollwert_B vorgegeben:

- Als Sollwert für PI Regler 1 mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für PI Regler 1 mittels des Istwert-Multiplexer, welcher über „FSP070_Controller_1_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für PI Regler 2 mittels des Sollwert-Multiplexer, welcher über „FSP080_Controller_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Istwert für interne PI Regler 2 mittels des Istwert-Multiplexer, welcher über „FSP080_Controller_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Summand 1 und/oder Summand 2 für den internen Regler 1 Addierer mittels des Regler_1_Summand_1_Multiplexers, welcher über „FSP077_Controller_1_AdderSourceSelectionMultiplexer“ konfiguriert wird.
- Als Sollwert für den WaterValveAbsValue welcher über „FSP102_WaterValve_Config“ konfiguriert“ wird.

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

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	$0x(\text{siehe Beschreibung})_H$

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 Schreibkommandos 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	**FSP045_AlteraRemoteUpdateCmd**
Adresse	**$0x2D_H/45_D/0x3244_{ASCII}$**
Tiefe	6/7 Byte / 48/56 Bit
I/O	lesen / schreiben
Reset	Reset:0x00100000_00_0_0_0_0_H

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

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)

Für CycloneV ändert sich dieses FSP inhaltlich leicht.
Die Startadresse wandert um 4 Bits nach links (beginnt nicht mehr bei Bit 20, sondern erst bei Bit 24) und wird um 4 weitere Bits ergänzt (hat also nun die Breite 32 Bits). (18.12.19 – DS)

Altes Format

[47..44] n.u.

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

[19..17] n.u.

Neues Format

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

[23..17] n.u.

Gemeinsam unverändert

[16] Reset WD Disable (only for debug)

[15..13] n.u.

[12] Start Write (steigende Flanke an diesem Bit startet die FSM zum 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	**FSP053_ModuleTemperatures**
Adresse	**$0x35_H/53_D/0x3335_{ASCII}$**
Tiefe	4 Byte / 32 Bit
I/O	lesen
Reset	$0x(\text{siehe Beschreibung})_H$

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

[31..24] Alarm_Info

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

[23..16] Sensor 3: Temperatur Modul Mitte

[15..8] Sensor 2: Temperatur FPGA

[7..0] Sensor 1: Temperatur DC-DC Wandler

Die Schwellen der Temperaturgrenzen werden im „FSP054_ModuleTemperaturesComparationThresholds“festgelegt.

Name	**FSP054_ModuleTemperaturesComparationThresholds**
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 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 Umgebung (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	$0x00_0000_H$

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 WICM.

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	$0x(\text{siehe Beschreibung})_H$

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	**FSP063_WaterFlowThresholds**
Adresse	**$0x3F_H/63_D/0x3346_{ASCII}$**
Tiefe	60 Byte / 480 Bit
I/O	lesen / schreiben
Reset	0x000000_000000_000000_000000_000000_000000_000000_000000_ 000000_000000_000000_000000_000000_000000_000000_000000_ 000000_000000_000000_000000_H

Gibt die größte und kleinste zulässige Zeitspanne zwischen zwei Wasserdurchflussmesser-Pulsen in µs vor, bei deren Über, bzw. Unterschreitung ein Interlock ausgelöst wird.

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

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

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

[479..456] Wasserdurchflusspulsmesser 10 max. Zeit zwischen zwei Pulsen

[455..432] Wasserdurchflusspulsmesser 10 min.Zeit zwischen zwei Pulsen

[431..408] Wasserdurchflusspulsmesser 9 max.Zeit zwischen zwei Pulsen

[407..384] Wasserdurchflusspulsmesser 9 min.Zeit zwischen zwei Pulsen

[383..360] Wasserdurchflusspulsmesser 8 max.Zeit zwischen zwei Pulsen

[359..336] Wasserdurchflusspulsmesser 8 min.Zeit zwischen zwei Pulsen

[335..312] Wasserdurchflusspulsmesser 7 max.Zeit zwischen zwei Pulsen

[311..288] Wasserdurchflusspulsmesser 7 min.Zeit zwischen zwei Pulsen

[287..264] Wasserdurchflusspulsmesser 6 max.Zeit zwischen zwei Pulsen

[263..240] Wasserdurchflusspulsmesser 6 min.Zeit zwischen zwei Pulsen

[239..216] Wasserdurchflusspulsmesser 5 max.Zeit zwischen zwei Pulsen

[215..192] Wasserdurchflusspulsmesser 5 min.Zeit zwischen zwei Pulsen

[191..168] Wasserdurchflusspulsmesser 4 max.Zeit zwischen zwei Pulsen

[167..144] Wasserdurchflusspulsmesser 4 min.Zeit zwischen zwei Pulsen

[143..120] Wasserdurchflusspulsmesser 3 max.Zeit zwischen zwei Pulsen

[119..96] Wasserdurchflusspulsmesser 3 min. Zeit zwischen zwei Pulsen

[95..72] Wasserdurchflusspulsmesser 2 max.Zeit zwischen zwei Pulsen

[71..48] Wasserdurchflusspulsmesser 2 min. Zeit zwischen zwei Pulsen

[47..24] Wasserdurchflusspulsmesser 1 max. Zeit zwischen zwei Pulsen

[23..0] Wasserdurchflusspulsmesser 1 min. Zeit zwischen zwei Pulsen

Name	**FSP064_InvertedWaterFlow_TimePeriodeBetweenTwoPulses_in_us**
Adresse	**$0x40_H/64_D/0x3440_{ASCII}$**
Tiefe	30 Byte / 240 Bit
I/O	lesen
Reset	$0x(\text{siehe Beschreibung})_H$

Liefert die Zeitspanne zwischen zwei Pulsen des Wasserdurchflusspulsmessers in uSekunden.

Dabei wird der Wert wie folgt errechnet:

Beispiel: 100us => $0x64_H$ wird im 2er Komplement als 0x7FFF9C erfasst und ausgegben.

0x7FFFFF (max. Wert) – tatsächliche Zeitspanne in µs. Dadurch kann dieser Wert u.a. zur Regelung benutzt werden.

[239..216] Zeitspanne beim Wasserdurchflusspulsmesser 10

[215..192] Zeitspanne beim Wasserdurchflusspulsmesser 9

[191..168] Zeitspanne beim Wasserdurchflusspulsmesser 8

[167..144] Zeitspanne beim Wasserdurchflusspulsmesser 7

[143..120] Zeitspanne beim Wasserdurchflusspulsmesser 6

[119..96] Zeitspanne beim Wasserdurchflusspulsmesser 5

[95..72] Zeitspanne beim Wasserdurchflusspulsmesser 4

[71..48] Zeitspanne beim Wasserdurchflusspulsmesser 3

[47..24] Zeitspanne beim Wasserdurchflusspulsmesser 2

[23..0] Zeitspanne beim Wasserdurchflusspulsmesser 1

Name	**FSP066_ADT7410_Temperatures**
Adresse	**$0x42_H/66_D/0x3442_{ASCII}$**
Tiefe	9 Byte / 72 Bit
I/O	lesen
Reset	$0x(\text{siehe Beschreibung})_H$

Es können bis zu 4 externe ADT7410 Temperatursensoren an das Modul angeschlossen werden.

[71] Wenn der ADT7410 an Kanal A gelesen wird ist dieses Bit ‚1‘

[70..68] n.u., immer ‚0‘

[67..64] Status

[3..0]	Beschreibung
0x00	n.u., Resetzustand
0x01	Device 1 an Kanal A gefunden
0x02	Device 1 an Kanal B gefunden
0x04	Device 1 an Kanal C gefunden
0x08	Device 1 an Kanal D gefunden

[63..48] Temperatur des ADT7410 an Kanal A (15 Bit)

[47..32] Temperatur des ADT7410 an Kanal B (15 Bit)

[31..16] Temperatur des ADT7410 an Kanal C (15 Bit)

[15..0] Temperatur des ADT7410 an Kanal D (15 Bit)

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	HighSpeedPort_ReceiveData[31..12]
0x4	0
0x5	ADT7410 Temp.Sensor an Kanal A
0x6	ADT7410 Temp.Sensor an Kanal B
0x7	ADT7410 Temp.Sensor an Kanal C
0x8	ADT7410 Temp.Sensor an Kanal D
0x9	Ausgang des PI_Controller, Regler 2, PI-Anteil
0xA	WaterFlow_TimeBetweenTwoPulses [23..4], Pulse 1
0xB	WaterFlow_TimeBetweenTwoPulses [47..28], Pulse 2
0xC	WaterFlow_TimeBetweenTwoPulses [71..58], Pulse 7
0xD	WaterFlow_TimeBetweenTwoPulses [95..76], Pulse 8
0xE	WaterFlow_TimeBetweenTwoPulses [215..196], Pulse 9
0xF	WaterFlow_TimeBetweenTwoPulses [239..220], Pulse 10

[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	HighSpeedPort_ReceiveData[31..12]
0x4	Ausgang des PI_Controller, Regler 2, P-Anteil
0x5	Ausgang des PI_Controller, Regler 2,I-Anteil
0x6	Ausgang des PI_Controller, Regler 2,PI-Anteil
0x7	n.u.
0x8	n.u.
0x9	n.u.
0xA	n.u.
0xB	n.u.
0xC	n.u.
0xD	n.u.
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 Multiplikatorstufen, 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 +/- 2^{10}.

[15..11] n.u.

[10..0] Multiplikator (11 Bit) für den Differenzbilder 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..97] n.u.

[96] Controller1_PI_Control, 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 P2 Anteil des PI Regler 1.

[31..0] Controller1_P1_Part (32 Bit), repräsentiert den P1 Anteil des PI Regler 1.

Name	**FSP073_Controller_1_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 und der P2 Anteil bei der Regelung berücksichtig werden soll.

[95..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 Eingangsmultiplexers für dessen Summanden 1 und 2.

[15..10] n.u.

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

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

[7..4] Quellenwahl für Multiplexer 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 2, P-Anteil
0x6	Ausgang des PI_Controller, Regler 2, I-Anteil
0x7	Ausgang des PI_Controller, Regler 2, PI-Anteil
0x8	Ausgang des PI_Controller, Regler 1, P-Anteil
0x9	Ausgang des PI_Controller, Regler 1, I-Anteil
0xA	Ausgang des PI_Controller, Regler 1, PI-Anteil
0xB	WaterFlow_TimeBetweenTwoPulses [23..4], Pulse 1
0xC	WaterFlow_TimeBetweenTwoPulses [47..28], Pulse 2
0xD	WaterFlow_TimeBetweenTwoPulses [71..52], Pulse 3
0xE	WaterFlow_TimeBetweenTwoPulses [215..196], Pulse 9
0xF	WaterFlow_TimeBetweenTwoPulses [239..220], Pulse 10

[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	WaterFlow_TimeBetweenTwoPulses [23..4], Pulse 1
0xC	WaterFlow_TimeBetweenTwoPulses [47..28], Pulse 2
0xD	WaterFlow_TimeBetweenTwoPulses [71..52], Pulse 3
0xE	WaterFlow_TimeBetweenTwoPulses [215..196], Pulse 9
0xF	WaterFlow_TimeBetweenTwoPulses [239..220], Pulse 10

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..22] AdderMaxVal (20 Bit), repräsentiert den oberen (maximalen) Grenzwert des Addierer.

[23..0] AdderMinVal (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	$0x(\text{siehe Beschreibung})_H$

Über diesen FSP lassen sich Regler 1 relevante Werte zurück lesen.

[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_MultipliedDeviation (20 Bit)

[47..24] Controller _1_ActValueMuxOut (20 Bit)

[23..0] Controller_1_SetValueMux_Out (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	HighSpeedPort_ReceiveData[31..12]
0x4	ADT7410 Temp.Sensor an Kanal A
0x5	ADT7410 Temp.Sensor an Kanal B
0x6	ADT7410 Temp.Sensor an Kanal C
0x7	ADT7410 Temp.Sensor an Kanal D
0x8	Ausgang des PI_Controller, Regler 1, P-Anteil
0x9	Ausgang des PI_Controller, Regler 1,I-Anteil
0xA	Ausgang des PI_Controller, Regler 1,PI-Anteil
0xB	WaterFlow_TimeBetweenTwoPulses [23..4], Pulse 1
0xC	WaterFlow_TimeBetweenTwoPulses [47..28], Pulse 2
0xD	WaterFlow_TimeBetweenTwoPulses [71..52], Pulse 3
0xE	WaterFlow_TimeBetweenTwoPulses [215..196], Pulse 9
0xF	WaterFlow_TimeBetweenTwoPulses [239..220], Pulse 10

[3..0] Regler 2 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	HighSpeedPort_ReceiveData[31..12]
0x4	Ausgang des PI_Controller, Regler 1, P-Anteil
0x5	Ausgang des PI_Controller, Regler 1,I-Anteil
0x6	Ausgang des PI_Controller, Regler 1,PI-Anteil
0x7	n.u.
0x8	n.u.
0x9	n.u.
0xA	n.u.
0xB	n.u.
0xC	n.u.
0xD	n.u.
0xE	n.u.

0xF	n.u.

Name	**FSP081_Controller_2_DifferenceCalculatorMultiplier**
Adresse	**$0x51_H/81_D/0x3531_{ASCII}$**
Tiefe	4 Byte / 32 Bit
I/O	lesen / schreiben
Reset	$0x0001_H$

Repräsentiert den Multiplikator für den Multiplikand Delta I der Regelabweichung des Regler 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 Multiplikatorstufen, 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 +/- 2^{10}.

[15..11] n.u.

[10..0] Multiplikator (11 Bit) für den Differenzbilder 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 Regler 1 gesetzt werden.

[103..97] n.u.

[96] Controller1_PI_Control
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	**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 und der P2 Anteil bei der Regelung berücksichtig werden soll.

[95..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	**FSP089_Controller_2_Values**
Adresse	**$0x59_H/89_D/0x3539_{ASCII}$**
Tiefe	18 Byte / 144 Bit
I/O	lesen
Reset	$0x(\text{siehe Beschreibung})_H$

Über diesen FSP lassen sich Regler 2 relevante Werte zurück lesen.

[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_MultipliedDeviation (20 Bit)

[47..24] Controller _2_ActValueMuxOut (20 Bit)

[23..0] Controller_2_SetValueMux_Out (20 Bit)

Name	**FSP102_WaterValve_Config**
Adresse	**$0x66_H/102_D/0x3636_{ASCII}$**
Tiefe	14 Byte / 112 Bit
I/O	lesen / schreiben
Reset	$0x00_00_000000_000000_000000_000000_H$

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

[111...104] WaterValveControl[7..0]

[104] wenn ‘1’ => WaterValveControl.PWMModeActive

[103..96] n.u.

[99..96] WaterValveMuxConfig

[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	0
0xC	0
0xD	0
0xE	0
0xF	0

[95..72] WaterValveMotorStartValue[23..0]

[71..48] WaterValveMaxValue[23..0]

[47..24] WaterValveMinValue[23..0]

[23..0] WaterValveHysteresis[23..0]

Name	**FSP103_WaterValve_MuxOut_AbsValue**
Adresse	**$0x67_H/103_D/0x3637_{ASCII}$**
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	$0x(\text{siehe Beschreibung})_H$

Liefert den Ausgangswert, den Absolutwert der Ventilsteuerung des unter „FSP102_WaterValve_Config“ eingestellten Multiplexers zurück.

[23..0] WaterValveMuxOutAbsValue [23..0] (20 Bit)

Name	**FSP113_HighSpeed_ReturnChannel_SourceSelectionMux**
Adresse	**$0x71_H/113_D/0x3731_{ASCII}$**
Tiefe	2 Byte / 16 Bit
I/O	schreiben/lesen
Reset	$0x00_0_0_H$

Ermöglicht das Setzen des HighSpeed Rückkanal-Multiplexers

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_Higher_14Bits_Mux	[17..4]	definierbar
SwitchingOperationsStatus	[3..0]	fix

[15..10] n.u.

[9] Ausgabe von HighSpeed_ReturnChannel_Lower_14Bits_Mux erfolgt invertiert

[8] Ausgabe von HighSpeed_ReturnChannel_Higher_14Bits_Mux erfolgt invertiert

[7..4] HighSpeed_ReturnChannel_Lower_14Bits_Mux

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	WaterFlow_TimeBetweenTwoPulses [23..4], Pulse 1
0x2	WaterFlow_TimeBetweenTwoPulses [47..28], Pulse 2
0x3	WaterFlow_TimeBetweenTwoPulses [71..52], Pulse 3
0x4	WaterFlow_TimeBetweenTwoPulses [95..76], Pulse 4
0x5	WaterFlow_TimeBetweenTwoPulses [119..100], Pulse 5
0x6	WaterFlow_TimeBetweenTwoPulses [143..124], Pulse 6
0x7	WaterFlow_TimeBetweenTwoPulses [167..148], Pulse 7
0x8	WaterFlow_TimeBetweenTwoPulses [191..172], Pulse 8
0x9	WaterFlow_TimeBetweenTwoPulses [215..196], Pulse 9
0xA	WaterFlow_TimeBetweenTwoPulses [239..220], Pulse 10
0xB	ADT7410 Temp.Sensor an Kanal C[15..0]
0xC	ADT7410 Temp.Sensor an Kanal D[15..0]
0xD	Controller_2_MultipliedDeviation[19..0]
0xE	Controller_1_Summand_2_MultiplexerOut[19..0]
0xF	HighSpeedPort_Received[31..12]

[3..0] HighSpeed_ReturnChannel_Higher_14Bits_Mux

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	WaterFlow_TimeBetweenTwoPulses [23..4], Pulse 1
0x2	WaterFlow_TimeBetweenTwoPulses [47..28], Pulse 2
0x3	WaterFlow_TimeBetweenTwoPulses [71..52], Pulse 3
0x4	WaterFlow_TimeBetweenTwoPulses [95..76], Pulse 4
0x5	WaterFlow_TimeBetweenTwoPulses [119..100], Pulse 5
0x6	WaterFlow_TimeBetweenTwoPulses [143..124], Pulse 6
0x7	WaterFlow_TimeBetweenTwoPulses [167..148], Pulse 7
0x8	WaterFlow_TimeBetweenTwoPulses [191..172], Pulse 8
0x9	WaterFlow_TimeBetweenTwoPulses [215..196], Pulse 9

0xA	WaterFlow_TimeBetweenTwoPulses [239..220], Pulse 10
0xB	Controller_1_MultipliedDeviation[19..0]
0xC	Controller_1_Summand_1_MultiplexerOut[19..0]
0xD	ADT7410 Temp.Sensor an Kanal A [15..0]
0xE	ADT7410 Temp.Sensor an Kanal B [15..0]
0xF	HighSpeedPort_Received[31..12]