

FSPs der
Multi Function Unit
mit LWL Erweiterung
(MFU LWL)
(ab FW 7.0.0)

Version vom: Donnerstag, 22. August 2024, 07:52:00

Inhaltsverzeichnis

1. Änderungsliste	1
2. FSPs MFU LWL	2
FSP001_ModuleStatus	3
0x01 _H /1 _D /0x3031 _{ASCII}	
FSP009_ModuleSerialNumber	5
0x09 _H /9 _D /0x3039 _{ASCII}	
FSP010_ModuleCommands	6
0x0A _H /10 _D /0x3041 _{ASCII}	
FSP013_PeripheralConfig	7
0x0D _H /13 _D /0x3044 _{ASCII}	
FSP014_CurrentScale	8
0x0E _H /14 _D /0x3045 _{ASCII}	
FSP015_VoltageScale	9
0x0F _H /15 _D /0x3046 _{ASCII}	
FSP016_BFieldScale	10
0x10 _H /16 _D /0x3130 _{ASCII}	
FSP020_ActualValue_A	11
0x14 _H /20 _D /0x3134 _{ASCII}	
FSP021_ActualValue_B	12
0x15 _H /21 _D /0x3231 _{ASCII}	
FSP029_ActualValuePhysicalQuantities	13
0x1D _H /29 _D /0x3144 _{ASCII}	
FSP030_SetValue_A	14
0x1E _H /30 _D /0x3145 _{ASCII}	
FSP031_SetValue_B	15
0x1F _H /31 _D /0x3146 _{ASCII}	
FSP039_SetValuePhysicalQuantities	16
0x27 _H /39 _D /0x3237 _{ASCII}	
FSP045_AlteraRemoteUpdateCmd	17
0x2D _H /45 _D /0x3244 _{ASCII}	
FSP046_AlteraRemoteUpdateStatus	18
0x2E _H /46 _D /0x3245 _{ASCII}	
FSP050_ModuleSupplyValues	19
0x32 _H /50 _D /0x3332 _{ASCII}	
FSP053_ModuleTemperatures	20
0x35 _H /53 _D /0x3335 _{ASCII}	
FSP054_ModuleTemperaturesComparisonThresholds	21
0x36 _H /54 _D /0x3336 _{ASCII}	
FSP058_ParameterChecksumValue	22
0x3A _H /58 _D /0x3341 _{ASCII}	
FSP059_ParameterChecksumValueCalculated	23
0x3B _H /59 _D /0x3342 _{ASCII}	
FSP060_SlopeLimiter_C1	24
0x3C _H /60 _D /0x3343 _{ASCII}	
FSP061_DifferenceCalculatorMultiplier	25
0x3D _H /61 _D /0x3631 _{ASCII}	
FSP062_LocalSetValue	26
0x3E _H /62 _D /0x3632 _{ASCII}	
FSP063_MPS	27
0x3F _H /63 _D /0x3346 _{ASCII}	
FSP064_USixHS_Multiplexer	28
0x40 _H /64 _D /0x3634 _{ASCII}	
FSP065_FrontLemoMultiplexer	35
0x41 _H /65 _D /0x3431 _{ASCII}	
FSP067_Defined_US1	37
0x43 _H /67 _D /0x3433 _{ASCII}	
FSP068_ButtonAndLEMOInStatus	38
0x44 _H /68 _D /0x3434 _{ASCII}	
FSP069_ExternalTriplinesStatus	39
0x45 _H /69 _D /0x3435 _{ASCII}	
FSP070_Controller_1_2_InputSourceSelectionMultiplexer	40
0x46 _H /70 _D /0x3436 _{ASCII}	
FSP071_Controller_1_SetValue	42
0x47 _H /71 _D /0x3731 _{ASCII}	
FSP072_Controller_1_ActualValue	43

0x48 _H /72 _D /0x3438 _{ASCII}	
FSP073_Controller_1_Limits	44
0x49 _H /73 _D /0x3439 _{ASCII}	
FSP074_Controller_1_PI_Settings	45
0x4A _H /60 _D /0x3441 _{ASCII}	
FSP075_Controller_1_I_Part_ComparatorLimits	46
0x4B _H /75 _D /0x3442 _{ASCII}	
FSP076_Controller_1_SetValueDeviation	47
0x4C _H /76 _D /0x3443 _{ASCII}	
FSP077_Controller_1_PI_Output	48
0x4D _H /77 _D /0x3444 _{ASCII}	
FSP078_Controller_1_P2_Part_ComparatorLimits	49
0x4E _H /78 _D /0x3445 _{ASCII}	
FSP079_Controller_1_SlopeLimiterOutput.....	50
0x4F _H /79 _D /0x3446 _{ASCII}	
FSP080_SlopeLimiter_C2	51
0x3C _H /60 _D /0x3343 _{ASCII}	
FSP081_Controller_2_SetValue	52
0x51 _H /81 _D /0x3531 _{ASCII}	
FSP082_Controller_2_ActualValue	53
0x52 _H /82 _D /0x3532 _{ASCII}	
FSP083_Controller_2_Limits	54
0x53 _H /83 _D /0x3533 _{ASCII}	
FSP084_Controller_2_PI_Settings	55
0x54 _H /84 _D /0x3534 _{ASCII}	
FSP085_Controller2_I_Part_ComparatorLimits	56
0x55 _H /85 _D /0x3535 _{ASCII}	
FSP086_Controller_2_SetValueDeviation	57
0x56 _H /86 _D /0x3536 _{ASCII}	
FSP087_Controller_2_PI_Output	58
0x57 _H /87 _D /0x3837 _{ASCII}	
FSP088_Controller_2_P2_Part_ComparatorLimits	59
0x58 _H /88 _D /0x3838 _{ASCII}	
FSP089_Controller_2_SlopeLimiterOutput.....	60
0x59 _H /89 _D /0x3839 _{ASCII}	
FSP090_Adder_SourceSelectionMultiplexer	61
0x5A _H /90 _D /0x3541 _{ASCII}	
FSP091_Adder_Limits.....	62
0x5B _H /91 _D /0x3542 _{ASCII}	
FSP092_Adder_SumOut	63
0x5C _H /92 _D /0x3543 _{ASCII}	
FSP093_CorrFactorPI_Limits	64
0x5D _H /93 _D /0x3544 _{ASCII}	
FSP094_CorrFactorPI_kP	65
0x5E _H /94 _D /0x3545 _{ASCII}	
FSP095_ComparatorControl	66
0x5F _H /95 _D /0x3546 _{ASCII}	
FSP096_SerialComDt2SendSel	68
0x60 _H /096 _D /0x360 _{ASCII}	
FSP097_SelVal2CompP2Comp	70
0x61 _H /97 _D /0x3631 _{ASCII}	
FSP098_Selectable_kIkP1	71
0x62 _H /98 _D /0x3632 _{ASCII}	
FSP099_Selectable_kIkP1Thresholds	72
0x63 _H /99 _D /0x3633 _{ASCII}	
FSP100_V5_ComparatorLimits	73
0x64 _H /100 _D /0x3634 _{ASCII}	
FSP101_Degauss_ComparatorLimit	74
0x65 _H /101 _D /0x3635 _{ASCII}	
FSP102_PWM_FDrive1_ComparatorLimits	75
0x66 _H /102 _D /0x3636 _{ASCII}	
FSP103_PWM_FDrive2_ComparatorLimits	76
0x67 _H /103 _D /0x3637 _{ASCII}	
FSP105_IGBT_AlternateSetValue.....	77
0x69 _H /105 _D /0x3639 _{ASCII}	
FSP106_EnergyRecoverLimitation_CurrentDriveValue	78
0x6A _H /106 _D /0x3641 _{ASCII}	

FSP107_CtrlEnable_Delay.....	79
0x6B _H /107 _D /0x3642 _{ASCII}	
FSP108_SerialCom_1_ErrorGenCfg.....	80
0x6C _H /108 _D /0x3643 _{ASCII}	
FSP109_SerialCom_1_Status.....	81
0x6D _H /109 _D /0x3644 _{ASCII}	
FSP110_DACx_and_ScopeChannelx_SourceSelectionMultiplexer.....	82
0x6E _H /110 _D /0x3645 _{ASCII}	
FSP111_SerialCom_2_ErrorGenCfg.....	85
0x6F _H /111 _D /0x3646 _{ASCII}	
FSP112_SerialCom_2_Status.....	86
0x70 _H /112 _D /0x3730 _{ASCII}	
FSP113_SerialCom_3_ErrorGenCfg.....	87
0x71 _H /113 _D /0x3731 _{ASCII}	
FSP114_intScopeTFTSettings.....	88
0x72 _H /114 _D /0x3732 _{ASCII}	
FSP116_intScopeSettings.....	89
0x74 _H /116 _D /0x3734 _{ASCII}	
FSP117_intScopeTriggerReadOut.....	91
0x75 _H /117 _D /0x3735 _{ASCII}	
FSP118_intScopeDataReadOutAddress.....	92
0x76 _H /118 _D /0x3736 _{ASCII}	
FSP119_intScopeDataReadOut.....	93
0x77 _H /119 _D /0x3737 _{ASCII}	
FSP120_intFunctionGenerator.....	94
0x78 _H /120 _D /0x3738 _{ASCII}	
FSP121_ControllerStatusBits LE/SE.....	96
0x79 _H /121 _D /0x3739 _{ASCII}	
FSP122_SerialCom_3_Status.....	97
0x7A _H /122 _D /0x3741 _{ASCII}	
FSP123_SerialCom_4_ErrorGenCfg.....	98
0x7B _H /123 _D /0x3742 _{ASCII}	
FSP124_SerialCom_4_Status.....	99
0x7C _H /124 _D /0x3743 _{ASCII}	

1. Änderungsliste

Datum	Name	Kommentar
17.03.2023	D. Schupp	Dokument erstellt aus ACU-FSPs_MFU_7_5
01.03.2024	D. Schupp	[FSP252_UpdateMFU_CFI_SoftwareViaRemote] Info über „Bulkerase“ des CFI hinzugefügt. FSP242_SW_CPU_Status[29] ergänzt.
16.08.2024	D. Schupp	MFU Soft FSP, die Bestandteil der NIOSII Software sind, sind nun ausgelagert in einem separaten Dokument [ACU-SoftFSPs_Nios_7_...]
16.08.2024	D. Schupp	MFU LWL FW 7.1 Ergänzungen
22.08.2024	D. Schupp	FSP108_SerialCom_1_ErrorGenCfg FSP111_SerialCom_2_ErrorGenCfg FSP113_SerialCom_3_ErrorGenCfg FSP123_SerialCom_4_ErrorGenCfg Inhalte ergänzt

2. FSPs MFU LWL

Dieses Dokument behandelt modulspezifische FSPs der MultiFunctionUnit (MFU) mit LWL Erweiterung. FSPs die mittels NIOSII Software innerhalb der MFU erzeugt werden beginnen mit „_SW_“, sind nicht in der MFU Hardware implementiert, sondern Bestandteil der Nios-Software. Diese FSPs sind ab sofort im Dokument [ACU-SoftFSPs_NIOS_7_...] zu finden.

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

[23..22] n.u., immer ‚0‘

[21] Zustand Remote/Local Schalter: ‚0‘ ACU System im Local-Betrieb ‚1‘, ACU System im Remote-Betrieb

[20] Controller Enabled
wenn ‚1‘ ist Regler freigegeben

[19..16] Gerätestatus (4 Bit)

[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>cMDDisableController</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] Kommando (lokales oder „remote“ Kommando)
→ siehe FSP010_ModuleCommands

[11..9] n.u. immer 0

[8] USIIsHighSpeed
wenn ‚1‘ ist mindestens eine der USI im Highspeed Mode

[7..6] 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, nachdem der Nios2 die Bootsequenz abgearbeitet hat.
- [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.
Dies ist dann der Fall, wenn entweder der Nios2 die Parameter aus dem internen Flash geladen hat oder die Parameter über einen PC mittels PCA in die ACu geladen wurden.

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

Der FSP enthält die MFU Serien Nummer

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] MFU Seriennummer (48 Bit)

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

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

Ist die MFU Betriebsart „Remote“ erfolgt die Kommandoübermittlung über das Kontrollsystem-Kommandoregister und die Daten aus diesem FSP werden ignoriert.

Zu beachten ist: die MFU reagiert nur auf Änderungen der Kommandos. Soll also das gleiche Kommando mehrfach hintereinander ausgeführt werden, bedarf es jedes Mal dem Kommando *cCMDNoAction* dazwischen.

[7..4] n.u.

[3..0] Command_USB, diese Kommandos werden z.B. von PowerConfigAdvanced oder ein Terminalprogramm über die MFU USB Verbindung gesetzt und steuern die Gerätefunktion, sofern sich die MFU in der Betriebsart „Local“ befindet.

[3..0]	Kommando
0x0	<i>cCMDNoAction</i> keine Aktion
0x1	<i>cCMDSwitchUnitOn</i> Gerät einschalten (wenn möglich)
0x2	<i>cCMDSwitchUnitOff</i> Gerät abschalten
0x3	<i>cCMDResetUnit</i> Reset durchführen (z.B. Interlocks)
0x4	<i>cCMDDisableController</i>
0x5	<i>cCMDTriggerSomething</i> hiermit lassen sich Sonderfunktionen in Modulen auslösen

Name	FSP013_PeripheralConfig
Adresse	0x0D_H/13_D/0x3044_{ASCII}
Tiefe	1 Byte / 8 Bit
I/O	lesen / schreiben
Reset	0x82 _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 in die MFU geladen (per PC oder durch die MFU selbst) 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 Freigabe des Reglers erfolgt aber nur, wenn die Vergleichs-Prüfsumme auch zu der aus den restlichen Parametern gebildeten Prüfsumme passt.
Das Löschen dieses Bit löst ggf. eine gezielte Reglersperre aus und die „alte“ in der MFU errechnete Prüfsumme wird gelöscht.
Die MFU führt im normalen Betrieb zyklische Zugriffe auf die FSP durch, dies auch während neue Parameter geladen werden. Lesezugriffe auf die FSP sind kein Problem, da die Prüfsumme nur bei Schreibzugriffen aufaddiert wird. Damit diese Zugriffe nicht für eine Verfälschung der Parameter Prüfsumme führen, werden Schreib-Zugriffe seitens der MFU unterbunden solange dieses Bit ‚0‘ ist.
Ggf. ist es sinnvoll, dieses Bit nicht mit einem direkten Zugriff auf FSP013 zu löschen, da hierdurch evtl. auch weitere Bits des FSP013 verändert werden. Soll dieses Bit nur gezielt gelöscht werden, kann auch FSP241 benutzt werden.
- [6..4] n.u.
- [3] Wenn ‚0‘ kann Bit [2] vom Anwender über das Menü NICHT geändert werden, d.h. es ist lokal am Gerät NICHT möglich zwischen Strom- und/oder Feldregelung umzuschalten. Wenn ‚1‘ kann Bit [2] hingegen am Gerät lokal über das Menü geändert werden. Nach dem Reset ist diese Bit immer ‚0‘, d.h. der Nutzer kann lokal den Regelmodus NICHT ändern.
- [2] Wenn ‚0‘ wird Stromgeregelt, wenn ‚1‘ Feldgeregelt. Nach dem ‚Reset‘ ist immer die Stromregelung gewählt.
- [1] Wenn ‚0‘ wird der Regler gesperrt, d.h. für eine Reglerfreigabe muss dieses Bit gesetzt werden. Dieses Bit ist UND-Verknüpft mit Bit[7].
- [0] wenn ‚1‘ können über den USB Port (Front MFU) Sollwerte und Kommandos (Ein, Aus, Reset) gesetzt werden, wenn ‚0‘ über Backplane-Interface (Interface-Karte oder SCU)

Name	FSP014_CurrentScale
Adresse	0x0E_H/14_D/0x3045_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen / schreiben
Reset	0x0000_000A _H

Repräsentiert die Stromskalierung des ACU Systems und des darin zur Anwendung kommenden DCCT (10V = CurrentScale Ampere).

[31] wenn ‚1‘ → bipolar, dabei reicht es, wenn FSP014[31], FSP015[31] oder FSP016[31] gesetzt ist, das Gerät wird dann in jedem Fall als bipolar betrachtet

[30..24] n.u.

[23..0] Stromskalierung (24 Bit)

Name	FSP015_VoltageScale
Adresse	0x0F_H/15_D/0x3046_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen / schreiben
Reset	Reset:0x0000_000A _H

Repräsentiert die Spannungskalierung des ACU Systems

[31] wenn ‚1‘ → bipolar, dabei reicht es, wenn FSP014[31], FSP015[31] oder FSP016[31] gesetzt ist, das Gerät wird dann in jedem Fall als bipolar betrachtet

[30..24] n.u.

[23..0] Spannungskalierung (24 Bit)

Name	FSP016_BFieldScale
Adresse	0x10_H/16_D/0x3130_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen / schreiben
Reset	Reset:0x0000_000A _H

Repräsentiert die Feldskalierung des ACU Systems und der darin zur Anwendung kommenden Feldsonde.

[31] wenn ‚1‘ → bipolar, dabei reicht es, wenn FSP014[31], FSP015[31] oder FSP016[31] gesetzt ist, das Gerät wird dann in jedem Fall als bipolar betrachtet

[30..24] n.u.

[23..0] Feldkalibrierung (24 Bit)

Name	FSP020_ActualValue_A
Adresse	0x14_H/20_D/0x3134_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann der Istwert_A gelesen werden. I.d.R. ist dies der Stromistwert des DCCT.

[23..0] Vorzeichenbehafteter 20 Bit Wert, Bit[3..0] sind immer ,0'.

Name	FSP021_ActualValue_B
Adresse	0x15_H/21_D/0x3231_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann der Istwert_ **B** gelesen werden. I.d.R. ist dies der Feldistwert der Feldsonde.

[23..0] Vorzeichenbehafteter 20 Bit Wert, Bit[3..0] sind immer ,0'.

Name	FSP029_ActualValuePhysicalQuantities
Adresse	0x1D_H/29_D/0x3144_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x0030 _H

Gibt die physikalische Größe der Istwerte an.

[15..12] (physikalische Einheit des Istwertes D) **obsolet**

[11..8] (physikalische Einheit des Istwertes C) **obsolet**

[7..4] physikalische Einheit des Istwertes B

[3..0] physikalische Einheit des Istwertes A

[3..0]	Einheit
0x0	Ampere (A)
0x1	Volt (V)
0x2	Temperatur (C)
0x3	Magnetische Flussdichte (T)
0x4	Magnetische Flussdichte (G)
0x5 ... 0xF	n.u.

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

Über diesen FSP wird der Sollwert **A** im Remotebetrieb an die MFU übermittelt.

- [23..0] Vorzeichenbehafteter 20 Bit Wert (0xFFFF bis 0x7FFFF), wird über verschiedene Multiplexer diversen Zielen vorgegeben, Bit[3..0] sind immer ‚0‘:
- Als **Sollwert** für den PI Regler **1** mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
 - Als **Sollwert** für den PI Regler **2** mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
 - Als **Istwert** für den PI Regler **1** mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
 - Als **Summand 1** des Addierers **1**, welcher über „FSP090_Adder_SourceSelectionMultiplexer“ konfiguriert wird.
 - Als Quelle für den „ACU_SovereigntySelector_USBorIFK_SCU“ Block, dessen Ausgang als „SetValue_A_MultiplexedOutput“ Verwendung findet. Abhängig von „FSPFSP013_PeripheralConfig[0]“ wird entweder dieser Sollwert(1) oder der Sollwert des Kontrollsystems(0) via Backplane für „SetValue_A_MultiplexedOutput“ verwendet.

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

Über diesen FSP wird der Sollwert **B** im Remotebetrieb an die MFU übermittelt.

[23..0] Vorzeichenbehafteter 20 Bit Wert (0xFFFF bis 0x7FFFF)
wird über verschiedene Multiplexer diversen Zielen vorgegeben, Bit[3..0] sind immer ,0':

- Als **Istwert** für den PI Regler **2** mittels des Sollwert-Multiplexer, welcher über „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“ konfiguriert wird.
- Als Summand **2** des Addierers **1**, welcher über „FSP090_Adder_SourceSelectionMultiplexer“ konfiguriert wird.
- Als Quelle für den „ACU_SovereigntySelector_USBorIFK_SCU“ Block, dessen Ausgang als „SetValue_B_MultiplexedOutput“ Verwendung findet. Abhängig von „FSPFSP013_PeripheralConfig[0]“ wird entweder dieser Sollwert(1) oder der Sollwert des Kontrollsystems(0) via Backplane für „SetValue_B_MultiplexedOutput“ verwendet.

Name	FSP039_SetValuePhysicalQuantities
Adresse	0x27_H/39_D/0x3237_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x0030 _H

Gibt die physikalische Größe der Sollwerte an.

[15..12] (physikalische Einheit des Sollwertes D) obsolet

[11..8] (physikalische Einheit des Sollwertes C) obsolet

[7..4] physikalische Einheit des Sollwertes B

[3..0] physikalische Einheit des Sollwertes A

[3..0]	Einheit
0x0	Ampere (A)
0x1	Volt (V)
0x2	Temperatur (C)
0x3	Magnetische Flussdichte (T)
0x4	Magnetische Flussdichte (G)
0x5 ... 0xF	n.u.

Name	FSP045_AlteraRemoteUpdateCmd
Adresse	0x2D_H/45_D/0x3244_{ASCII}
Tiefe	7 Byte / 56 Bit
I/O	lesen / schreiben
Reset	Reset:0x00800000_00_0_0_0_0 _H

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

Imagetyp lesen

Bit[4] = ,0' (Read)

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

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

Imagetyp wechseln

Bit[4] = ,1' (Write)

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

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

[23..17] n.u.

[16] Reset WD Disable (only for debug)

[15..13] n.u.

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

[11..9] n.u.

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

[7..5] n.u.

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

[3..2] n.u.

[1..0] Read Source

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

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

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

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

Liefert die vorzeichenbehafteten Betriebsspannungen des Moduls. Immer 2 Byte stehen für eine Spannung, d.h. bei 8 Spannungen ist dieses FSP 16 Byte tief. Die Spannungen sind vorzeichenbehaftet und wie folgt sortiert.

- [127..112] vorzeichenbehaftete **-12 Volt** (13 Bit)
- [111..96] vorzeichenbehaftete **Pad1** (13 Bit) (floatet, da Pad offen)
- [95..80] vorzeichenbehaftete **12 Volt** (13 Bit)
- [79..64] vorzeichenbehaftete **Pad2** (13 Bit) (floatet, da Pad offen)
- [63..48] vorzeichenbehaftete **5 Volt** (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)

Die Bitwertigkeiten sind wie folgt:

bei 1V2, 2V5, 3V3 und 5V0 werden die gemessenen Werte direkt mit dem LSB Wert (SUP_LSB_Size = 0.002441) multipliziert und ergeben so die Spannung. Bei -12V und 12V erfolgt ebenfalls zuerst die LSB Wert Multiplikation und anschließend die Multiplikation mit dem Faktor 11, der dem vorgeschalteten Spannungsteiler entspricht. Die tatsächlich gemessenen Spannungen liegen nämlich bei 1/11 der Originalspannung.

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

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

[31..24] Alarm_Info

[7..0]	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] Device **3**: Temperatur Modul Mitte (8 Bit)

[15..8] Device **2**: Temperatur Längsregler (8 Bit)

[7..0] Device **1**: Temperatur FPGA (8 Bit)

Die Schwellen der Temperaturgrenzen werden im „FSP054_ModuleTemperaturesComparationThresholds“ festgelegt.

Name	FSP054_ModuleTemperaturesComparisonThresholds
Adresse	0x36_H/54_D/0x3336_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x46_4646 _H

Stellt die 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] Vergleichswert Device **3**: Temperatur Modul Mitte (8 Bit)

[15..8] Vergleichswert Device **2**: Temperatur Längsregler (8 Bit)

[7..0] Vergleichswert Device **1**: Temperatur FPGA (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 „FSP058_ParameterChecksumValue“ verglichen.

[23..0] Prüfsumme der Datenübertragung vom PC zur MFU.

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 zur MFU.

Name	FSP060_SlopeLimiter_C1
Adresse	0x3C_H/60_D/0x3343_{ASCII}
Tiefe	30 Byte / 240 Bit
I/O	lesen / schreiben
Reset	0x00_000000_000000_000000_000000_000000_0000_0000_0000_0000_ 745D17_8BA2E8 _H

Repräsentiert Parameter des Steilheitsbegrenzers.

Das Äquivalent für den Steilheitsbegrenzer für den Regler **2** findet sich im „FSP080_SlopeLimiter_C2“.

[239..234] n.u.

[233] Aktiviert den Steilheitbegrenzer,
wenn ‚1‘ UND ‚ACU_ControllerEnable‘

[232] Wenn ‚0‘ wird der Startwert des Steilheitbegrenzers mit 0 geladen, andernfalls der Wert
des Eingangs ‚Dynamic_Threshold_RiseTime_Selection‘.
Dies ist i.d.R. ‚Controller1_ActValueMuxOut‘

[231..216] RampThresholdValue_D (16 Bit)

[215..200] RampThresholdValue_C (16 Bit)

[199..184] RampThresholdValue_B (16 Bit)

[183..168] RampThresholdValue_A (16 Bit)

[167..144] RampRiseTime_E (24 Bit)

[143..120] RampRiseTime_D (24 Bit)

[119..96] RampRiseTime_C (24 Bit)

[95..72] RampRiseTime_B (24 Bit)

[71..48] RampRiseTime_A (24 Bit)

[47..24] Obere Grenze (24 Bit)

[23..0] Untere Grenze (24 Bit)

Name	FSP061_DifferenceCalculatorMultiplier
Adresse	0x3D_H/61_D/0x3631_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x03E8_03E8_03E8 _H

Repräsentiert den Multiplikator für den Multiplikand Delta I der Regelabweichung der Regler **1** und **2**, bzw. den Multiplikand Delta I der Differenzen zwischen Istwert Regler **1** und Istwert Regler **2**.

[47..27] n.u.

[26..16] Multiplikator (11 Bit) für den Differenzbilder Regler **2**

[15..11] n.u.

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

Name	FSP062_LocalSetValue
Adresse	0x3E_H/62_D/0x3632_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x000000 _H

Repräsentiert den lokalen (Hand) StromSOLLwert.

[23..0] Vorzeichenbehafteter 20 Bit Wert (0xFFFF bis 0x7FFFF), wird automatisch dem Sollwert_**A** zugewiesen, wenn der Schalter Local/Remote auf Local geschaltet ist. Die Bits[3..0] sind immer ,0'.

Name	FSP063_MPS
Adresse	0x3F_H/63_D/0x3346_{ASCII}
Tiefe	7 Byte / 56 Bit
I/O	lesen/schreiben
Reset	0x00_0000_0000_0000 _H

With this FSP it is possible to configure the MPS (Machine Protection System) comparator used to generate the MPS_OUT signal sent via LEMO.

[55..53] n.u.

[52] CtrlSel_1L_2H: it selects one of the two available PI controller outputs as value to compare in input to the hysteresis comparator. =1 => controller2 output selected; =0=> controller1 output selected.

[51..49] n.u.

[48] MPS_outSel_DIL_DITH: it selects between the hysteresis comparator output (=0) and the OR combination of the same signal with the tripline (=1) as value of MPS_OUT signal.

[47..24] Histeresys comparator **OFF** threshold.(20 Bit)

[23..0] Histeresys comparator **ON** threshold (20 Bit)

Name	FSP064_USI_{HS}_Multiplexer
Adresse	0x40_H/64_D/0x3634_{ASCII}
Tiefe	13 Byte / 103 Bit
I/O	lesen / schreiben
Reset	0x00_0000_0000_0000_0000_0000_0000 _H

Zur Nutzung der USI HighSpeed Transferkanäle darf an eine USI nur maximal ein Modul angeschlossen sein und dieses muss das USI HighSpeed Protokoll unterstützen.

Jede USI unterstützt genau einen USI HighSpeed-Transferkanal in eine Datenrichtung. D.h. ein HighSpeed-Transferkanal von der MFU zum angeschlossene Module mit HighSpeed Unterstützung und ein HighSpeed-Transferkanal vom angeschlossenen Modul mit HighSpeed-Unterstützung zur MFU.

Die Übertragung eines vollständigen Strings benötigt bei maximaler USI Bitrate von 20MBit 6us.

Mit diesem FSP ist eine Vielzahl von Multiplexer parametrierbar, die ein umfangreiches „routen“ von USI HighSpeed Transferkanal Daten von verschiedenen Quellen an verschiedene Ziele ermöglicht.

Die Betrachtung ist dabei wie folgt:

GEHEND[TO] die HighSpeed Daten gehen von der MFU an ein Modul

MFU interne Signale lassen sich hiermit auf USI HighSpeed-Transferkanäle legen. Es können Signale ausgewählt werden, welche via HighSpeed Transferkanal von der MFU über die gewählte USI an das daran angeschlossene Modul mit HighSpeed-Unterstützung übertragen werden.

KOMMEND[FROM] die HighSpeed Daten kommen von einem Modul zur MFU

Hiermit wird definiert von welchem Modul mit HighSpeed-Unterstützung welches Signal in die MFU zurück gelangt. Die Signale dieser USI HighSpeed-Transferkanäle müssen zuvor innerhalb des sendenden Moduls auf den USI HighSpeed-Transferkanal gelegt werden. Jedes Modul hat ggf. standardisierte Daten die im HighSpeed-Transferkanal übertragen werden. Der ADC wird z.B. immer ADC Werte im USI HighSpeed-Transferkanal senden, das ICM immer Netzgeräte-Status-Informationen. Sofern nicht alle Bits des USI HighSpeed-Transferkanals belegt sind ist es möglich wahlweise auf freie Bits zusätzliche Datensignale eines Moduls mittels eines Multiplexers zu legen. Diese zurück gesendeten Daten werden MFU intern wiederum auf bestimmte Signalwege aufgeschaltet, die für eine einwandfreie Funktion des ACU Gesamtsystems notwendig sind.

Kommende [FROM] Signale werden dabei teilweise noch über einen weiteren nachgeschalteten Multiplexer geführt um deren Wertigkeiten und Inhalte weiter zu verfeinern.

[103..100] ControllerX_PreSelectionMux_2 KOMMEND[FROM]

[99..96] ControllerX_PreSelectionMux_1 KOMMEND[FROM]

[95..92] n.u.

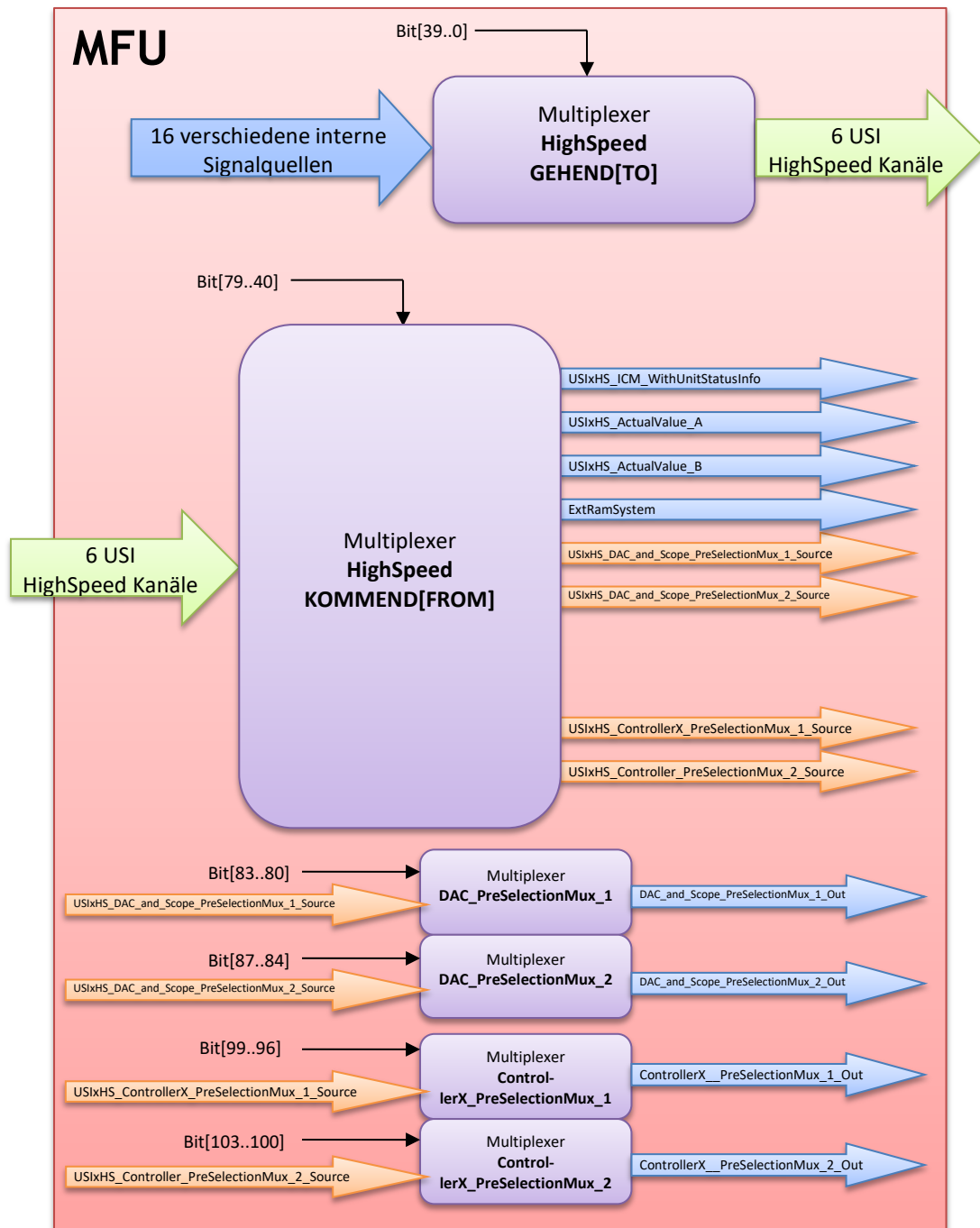
[91..88] n.u.

[87..84] DAC_and_Scope_PreSelectionMux_2 KOMMEND[FROM]

[83..80] DAC_and_Scope_PreSelectionMux_1 KOMMEND[FROM]

[79..40] Multiplexer HighSpeed Daten KOMMEND[FROM]

[39..0] Multiplexer HighSpeed GEHEND[TO]



PreSelectionMux ermöglichen beliebige USI HighSpeed Kanäle von den jeweiligen Quellen zum Ziel zu leiten und dabei eine Vorauswahl zu treffen, welche Bit des empfangenen Wertes dabei am Ziel wirklich ankommen sollen.

[103..100] ControllerX_PreSelectionMux_2

Über 4 Bits werden die Daten am Eingang dieses Multiplexers einem Ausgang (ControllerX_PreSelectionMux_2_Out[19..0]) zugewiesen.

[3..0]	Daten am Eingang
0x0	n.u
0x1	USIxHS_ControllerX_PreSelectionMux_2_Source[31..12]
0x2	USIxHS_ControllerX_PreSelectionMux_2_Source[31..18],GND_BUS[5..0]
0x3	USIxHS_ControllerX_PreSelectionMux_2_Source[17..4],GND_BUS[5..0]
0x4	n.u., immer GND_BUS[19..0]
0x5	GND_BUS[19..0]

...	
0xF	

[99..96] ControllerX_PreSelectionMux_1

Über 4 Bits werden die Daten am Eingang dieses Multiplexers einem Ausgang (ControllerX_PreSelectionMux_1_Out[19..0]) zugewiesen.

[3..0]	Daten am Eingang
0x0	n.u.
0x1	USI _x HS_ControllerX_PreSelectionMux_1_Source[31..12]
0x2	USI _x HS_ControllerX_PreSelectionMux_1_Source[31..18],GND_BUS[5..0]
0x3	USI _x HS_ControllerX_PreSelectionMux_1_Source[17..4],GND_BUS[5..0]
0x4	n.u., immer GND_BUS[19..0]
...	
0xF	

[95..92] n.u.

[91..88] n.u.

[87..84] DAC_and_Scope_PreSelectionMux_2

Über 4 Bits werden die Daten am Eingang dieses Multiplexers einem Ausgang (DAC_and_Scope_PreSelectionMux_2_Out[19..0]) zugewiesen.

[3..0]	Daten am Eingang
0x0	n.u.
0x1	USI _x HS_DAC_and_Scope_PreSelectionMux_2_Source[31..12]
0x2	USI _x HS_DAC_and_Scope_PreSelectionMux_2_Source[31..18],GND_BUS[5..0]
0x3	USI _x HS_DAC_and_Scope_PreSelectionMux_2_Source[17..4],GND_BUS[5..0]
0x4	n.u., immer GND_BUS[19..0]
...	
0xF	

[83..80] DAC_PreSelectionMux_1

Über 4 Bits werden die Daten am Eingang dieses Multiplexers einem Ausgang (DAC_PreSelectionMux_1_Out[19..0]) zugewiesen.

[3..0]	Daten am Eingang
0x0	n.u.
0x1	USI _x HS_DAC_PreSelectionMux_1_Source[31..12]
0x2	USI _x HS_DAC_PreSelectionMux_1_Source[31..18],GND_BUS[5..0]
0x3	USI _x HS_DAC_PreSelectionMux_1_Source[17..4],GND_BUS[5..0]
0x4	n.u., immer GND_BUS[19..0]
...	
0xF	

[79..40] Mux_16to10_DataFROMHighSpeedPorts

Dieser Multiplexer hat 16 Eingänge, die in Abhängigkeit der Bitnibbel 10 Ausgängen zugeordnet werden. Die Daten an diesen Ausgängen entsprechen den empfangenen HighSpeedDaten der zugehörigen USI. Z.B. Eingang 2 ist der HighSpeed Empfangskanal von USI 2.

[79..76] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang 10 gelegt wird

[75..72] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang 9 gelegt wird

[71..68] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang 8 gelegt wird

- [67..64] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang **7** gelegt wird
- [63..60] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang **6** gelegt wird
- [59..56] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang **5** gelegt wird
- [55..52] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang **4** gelegt wird
- [51..48] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang **3** gelegt wird
- [47..44] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang **2** gelegt wird
- [43..40] legt fest, welcher HighSpeedPort_x_RcvdData an den Ausgang **1** gelegt wird

Nachfolgend sind die Belegungen der Multiplexereingänge aufgeführt

DataIn	Daten am Eingang
1	HighSpeedPort_1_RcvdData [31..0]
2	HighSpeedPort_2_RcvdData [31..0]
3	HighSpeedPort_3_RcvdData [31..0]
4	HighSpeedPort_4_RcvdData [31..0]
5	HighSpeedPort_5_RcvdData [31..0]
6	HighSpeedPort_6_RcvdData [31..0]
7	PayloadRx_1_data[23..0],GND_BUS[7..0] *)
8	PayloadRx_2_data[23..0],GND_BUS[7..0] *)
9	PayloadRx_3_data[23..0],GND_BUS[7..0] *)
10	PayloadRx_4_data[23..0],GND_BUS[7..0] *)

*) Diese Signale stehen erst ab MFU LWL FW 7.1. zur Verfügung. Vorher sind diese Eingänge immer [0].

Die Ausgänge erwarten folgende Informationen

DataOut	Daten am Ausgang
1	USIxHS_ICM_WithUnitStatusInfo[31..0]
2	USIxHS_ActualValue_A[31..0]
3	USIxHS_ActualValue_B[31..0]
4	ExtRamSystem[31..0]
5	USIxHS_DAC_and_Scope_PreSelectionMux_1_Source[31..0]
6	USIxHS_DAC_and_Scope_PreSelectionMux_2_Source[31..0]
7	n.u.
8	n.u.
9	USIxHS_ControllerX_PreSelectionMux_1_Source[31..0]
10	USIxHS_ControllerX_PreSelectionMux_2_Source[31..0]

An den Bits[79..40] werden die Eingänge (DataIn) für die jeweiligen Ausgänge (DataOut) des Multiplexers gewählt.

Nachfolgend ist aufgeführt welche Bitkombination den jeweiligen Eingang auf welchen Ausgang legt.

Bitkombination	Zuweisung (Daten am Ausgang)
0x0	Eingang 1 [31..0]
0x1	Eingang 2 [31..0]
0x2	Eingang 3 [31..0]
0x3	Eingang 4 [31..0]
0x4	Eingang 5 [31..0]
0x5	Eingang 6 [31..0]
0x6	Eingang 7 [31..0]

0x7	Eingang 8 [31..0]
0x8	Eingang 9 [31..0]
0x9	Eingang 10 [31..0]
0xA..0xF	Ausgang ist 0

Beispiel:

Die Bitkombination (Ausgang n) → [(n*4)-1..(n-4)] legt fest welcher Eingang (HighSpeed-Port_x_RcvdData [31..0]) auf den Ausgang gelegt wird.

Sollen die Daten von Eingang **2** an Ausgang **2** müssen die Bit 47.44 auf 1 stehen

$$[(2 * 4)-1 .. (2 * 4) -4] = [47..44]$$

Sollen die Daten von Eingang **9** an Ausgang **6** müssen die Bit 63..60 auf 1 stehen

$$[(6 * 4)-1 .. (6 * 4) -4] = [63..60]$$

[39..0] Mux_16to10_DataTOHighSpeedPorts

Dieser Multiplexer hat 16 Eingänge, die in Abhängigkeit der Bitnibbel 10 Ausgängen zugeordnet werden. Die Daten an diesen Ausgängen entsprechen den zu sendenden HighSpeedDaten der zugehörigen USI. Z.B. Ausgang **2** ist der HighSpeed Sendekanal von USI **2**.

[39..36] ~~legt fest, welcher Eingang an HighSpeedPort_10_TransferData gelegt wird *)~~

[35..32] ~~legt fest, welcher Eingang an HighSpeedPort_9_TransferData gelegt wird *)~~

[31..28] ~~legt fest, welcher Eingang an HighSpeedPort_8_TransferData gelegt wird *)~~

[27..24] ~~legt fest, welcher Eingang an HighSpeedPort_7_TransferData gelegt wird *)~~

[23..20] legt fest, welcher Eingang an HighSpeedPort_6_TransferData gelegt wird

[19..16] legt fest, welcher Eingang an HighSpeedPort_5_TransferData gelegt wird

[15..12] legt fest, welcher Eingang an HighSpeedPort_4_TransferData gelegt wird

[11..8] legt fest, welcher Eingang an HighSpeedPort_3_TransferData gelegt wird

[7..4] legt fest, welcher Eingang an HighSpeedPort_2_TransferData gelegt wird

[3..0] legt fest, welcher Eingang an HighSpeedPort_1_TransferData gelegt wird

*) Diese Kanäle sind zwar physisch vorhanden, können jedoch nicht verwendet werden, da die MFU LWL nur 6 externe USIs unterstützt.

Nachfolgend sind die Belegungen der Multiplexereingänge aufgeführt

DataIn	Daten am Eingang
1	[31..5] GND_BUS[26..0] [4] IGBT_V5_Active [3..0] Command[3..0]
2	[31..12] ERLOutValue[19..0] [11..5] GND_BUS[6..0] [4] IGBT_V5_Active [3..0] Command[3..0]
3	[31..12] Controller1_SetValueMuxOut[19..0] [11..5] GND_BUS[6..0] [4] IGBT_V5_Active, [3..0] Command[3..0]
4	[31..12] Controller2_SetValueMuxOut[19..0] [11..5] GND_BUS[6..0] [4] IGBT_V5_Active [3..0] Command[3..0]
5	[31..13] GND_BUS[31..13] [12..9] HighSpeedPort_3_Out[3..0]

	[8..4] <i>MUX_DesignatedLoad</i> [4..0] [3..0] <i>Command</i> [3..0]
6	[31..12] <i>SetValue_A_MuxOut</i> [19..0] [11..5] <i>GND_BUS</i> [6..0] [4] <i>IGBT_V5_Active</i> [3..0] <i>Command</i> [3..0]
7	[31..12] <i>SetValue_B_MuxOut</i> [19..0] [11..5] <i>GND_BUS</i> [6..0] [4] <i>IGBT_V5_Active</i> [3..1] <i>Command</i> [3..0]
8	[31..0] <i>USIxHS_ICM_WithUnitStatusInfo</i> [31..0]
9	[31..18] <i>Controller1_SetValueDeviation</i> [19..6] [17..4] <i>Controller1_ActValMuxOut</i> [19..6] [3..0] <i>Command</i> [3..0]
10	[31..18] <i>SlopeLimiterOut_C1</i> [19..6] [17..4] <i>Controller1_ActValMuxOut</i> [19..6] [3..0] <i>Command</i> [3..0]
11	[31..18] <i>ERLOutValue</i> [19..6] [17..5] <i>Controller1_SetValueDeviation</i> [19..7] [4] <i>IGBT_V5_Active</i> , [3..0] <i>Command</i> [3..0]
12	[31..18] <i>SlopeLimiterOut_C2</i> [19..6] [17..4] <i>Controller2_ActValMuxOut</i> [19..6] [3..0] <i>Command</i> [3..0]
13	[31..18] <i>Controller1_ActValMuxOut</i> [19..6] [17..4] <i>Controller2_ActValMuxOut</i> [19..6] [3..0] <i>Command</i> [3..0]
14	[31..8] <i>FSP20_ActualValue_A</i> [23..0] [7..5] <i>GND_BUS</i> [2..0] [4] <i>IGBT_V5_Active</i> [3..0] <i>Command</i> [3..0]
15	n.u.
16	n.u.

Nachfolgend ist aufgeführt welche Bitkombination den jeweiligen Eingang auf welchen Ausgang legt.

Bitkombination	Zuweisung (Daten am Ausgang)
0x0	Eingang 1 [31..0]
0x1	Eingang 2 [31..0]
0x2	Eingang 3 [31..0]
0x3	Eingang 4 [31..0]
0x4	Eingang 5 [31..0]
0x5	Eingang 6 [31..0]
0x6	Eingang 7 [31..0]
0x7	Eingang 8 [31..0]
0x8	Eingang 9 [31..0]
0x9	Eingang 10 [31..0]
0xA	Eingang 11 [31..0]
0xB	Eingang 12 [31..0]
0xC	Eingang 13 [31..0]
0xD	Eingang 14 [31..0]

0xE	Eingang 15 [31..0]
0xF	Eingang 16 [31..0]

Beispiel:

Die Bitkombination (USI n) $\rightarrow [(n*4)-1..(n-4)]$ legen fest welcher Eingang auf den Ausgang (HighSpeedPort_x_TransferData) gelegt wird.

Sollen die Daten von Eingang **2** an HighSpeedPort_**2**_TransferData (USI2) müssen die Bit 7..4 auf 0x1 stehen

$$[(2 * 4)-1 .. (2 * 4) -4] = [7..4]$$

Sollen die Daten von Eingang **9** an HighSpeedPort_**6**_TransferData (USI6) müssen die Bit 23..20 auf 0x8 stehen

$$[(6 * 4)-1 .. (6 * 4) -4] = [23..20]$$

Name	FSP065_FrontLemoMultiplexer
Adresse	0x41_H/65_D/0x3431_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x17_17 _H

Repräsentiert die Einstellungen der Ausgangsmultiplexer für die Front-Lemo-Ausgangsbuchsen.

[15..13] n.u.

[12..8] Quellenwahl für Front LEMO Ausgang **2** (X2) (5 Bit)

[4..0]	Ausgang des Multiplexers
0x00	intScopeTriggerEventDetected
0x01	intScopeTimeBasePulse
0x02	ControllerStatusBit[0] (GND)
0x03	ControllerStatusBit[1] (GND)
0x04	ControllerStatusBit[2] (GND)
0x05	ControllerStatusBit[3] (GND)
0x06	ControllerStatusBit[4] (GND)
0x07	ControllerStatusBit[5] (GND)
0x08	ControllerStatusBit[6] (GND)
0x09	ControllerStatusBit[7] (ACU_Controller_Enable)
0x0A	ControllerStatusBit[8] (CurrentEqual)
0x0B	ControllerStatusBit[9] (IGBT_V5_Active)
0x0C	ControllerStatusBit[10] (Controller2_IGBT_V5_Active)
0x0D	ControllerStatusBit[11] (Controller1_IGBT_V5_Active)
0x0E	ControllerStatusBit[12] (PWM_FDRIVE_Comparator_MuxOut)
0x0F	ControllerStatusBit[13] (Controller2_PWM_FDRIVE_Comparator)
0x10	ControllerStatusBit[14] (Controller1_PWM_FDRIVE_Comparator)
0x11	ControllerStatusBit[15] (Degaussing_Active)
0x12	ControllerStatusBit[16] (Controller2_P2_Part_Active)
0x13	ControllerStatusBit[17] (Controller1_P2_Part_Active)
0x14	ControllerStatusBit[18] (Controller2_I_Part_Disable)
0x15	ControllerStatusBit[19] (Controller1_I_Part_Disable)
0x16	1 (VCC)
0x17	0 (GND)
0x18	MPS
0x19	
...	n.u.
0x1F	

[7..5] n.u.

[4..0] Quellenwahl für Front LEMO Ausgang **1** (X1) (5 Bit)

[4..0]	Ausgang des Multiplexers
0x00	intScopeTriggerEventDetected
0x01	intScopeTimeBasePulse
0x02	ControllerStatusBit[0] (GND)
0x03	ControllerStatusBit[1] (GND)
0x04	ControllerStatusBit[2] (GND)

0x05	ControllerStatusBit[3] (GND)
0x06	ControllerStatusBit[4] (GND)
0x07	ControllerStatusBit[5] (GND)
0x08	ControllerStatusBit[6] (GND)
0x09	ControllerStatusBit[7] (ACU_Controller_Enable)
0x0A	ControllerStatusBit[8] (CurrentEqual)
0x0B	ControllerStatusBit[9] (IGBT_V5_Active)
0x0C	ControllerStatusBit[10] (Controller2_IGBT_V5_Active)
0x0D	ControllerStatusBit[11] (Controller1_IGBT_V5_Active)
0x0E	ControllerStatusBit[12] (PWM_FDRIVE_Comparator_MuxOut)
0x0F	ControllerStatusBit[13] (Controller2_PWM_FDRIVE_Comparator)
0x10	ControllerStatusBit[14] (Controller1_PWM_FDRIVE_Comparator)
0x11	ControllerStatusBit[15] (Degaussing_Active)
0x12	ControllerStatusBit[16] (Controller2_P2_Part_Active)
0x13	ControllerStatusBit[17] (Controller1_P2_Part_Active)
0x14	ControllerStatusBit[18] (Controller2_I_Part_Disable)
0x15	ControllerStatusBit[19] (Controller1_I_Part_Disable)
0x16	1 (VCC)
0x17	0 (GND)
0x18	MPS
0x19	n.u.
...	
0x1F	

Name	FSP067_Defined_USI
Adresse	0x43_H/67_D/0x3433_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x0000H

This FSP is used to enable the High Speed data received/sent from/to MFU. It is written during the power converter setup and it contains the information of which USI is defined in the configuration file.

[15..10] Not used.

[9] Enable High Speed data for USI**10** (active high)

[8] Enable High Speed data for USI**9** (active high)

[7] Enable High Speed data for USI**8** (active high)

[6] Enable High Speed data for USI**7** (active high)

[5] Enable High Speed data for USI**6** (active high)

[4] Enable High Speed data for USI**5** (active high)

[3] Enable High Speed data for USI**4** (active high)

[2] Enable High Speed data for USI**3** (active high)

[1] Enable High Speed data for USI**2** (active high)

[0] Enable High Speed data for USI**1** (active high)

Name	FSP068_ButtonAndLEMOInStatus
Adresse	0x44_H/68_D/0x3434_{ASCII}
Tiefe	1 Byte / 8 Bit
I/O	Lesen
Reset	0x(siehe Beschreibung) _H

Stellt die Status der Tasten, Schalter und LEMO Eingänge zur Verfügung.

- [7] Status LEMO_IN[2] (X4)
- [6] Status LEMO_IN[1] (X3)
- [5] n.u., immer ,0'
- [4] n.u., immer ,0'
- [3] Status Schalter REMOTE/LOCAL, ,1' = REMOTE
- [2] Status RESET Taste, ,1' = gedrückt
- [1] Status OFF Taste, ,1' = gedrückt
- [0] Status ON Taste, ,1' = gedrückt

Name	FSP069_ExternalTriplinesStatus
Adresse	0x45_H/69_D/0x3435_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	Lesen
Reset	0x(siehe Beschreibung) _H

Stellt die Status der externen Reißleinen zur Verfügung.

[31..26] n.u., immer ,0'

[25..16] ExtTripLinesPulled_n, tatsächlicher Zustand der externen Reißleinen. Eine Reißleine gilt als geschlossen, sofern das zugehörige Bit gesetzt (,1') ist.

[15.10] n.u., immer ,0'

[9..0] ExtTripLinesPulled_nMem, gespeicherter Zustand der externen Reißleinen. Eine Reißleine gilt als geschlossen, sofern das zugehörige Bit gesetzt (,1') ist. Wird eine Reißleine geöffnet wird sofort der Zustand aller Reißleinen gespeichert und hier ausgegeben. Die Reißleine die zuerst ausgelöst wurde ist ,0', alle übrigen Bits blieben unverändert. Erst wenn alle Reißleinen wieder geschlossen sind, wird die Ausgabe von , ExtTripLinesPulled_nMem' sich wieder auf ,0x11_1111_1111' ändern.

Name	FSP070_Controller_1_2_InputSourceSelectionMultiplexer
Adresse	0x46_H/70_D/0x3436_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x00_0_0_0_0 _H

Repräsentiert die Einstellungen der PI Regler **1** und **2** Eingangsmultiplexer für dessen Soll- und Istwertquelle.

[23..20] n.u.

[19] '1' Ausgangswert des Regler **2** Istwertquellen-Multiplexers wird invertiert

[18] '1' Ausgangswert des Regler **2** Sollwertquellen-Multiplexers wird invertiert

[17] '1' Ausgangswert des Regler **1** Istwertquellen-Multiplexers wird invertiert

[16] '1' Ausgangswert des Regler **1** Sollwertquellen-Multiplexers wird invertiert

[15..12] Regler **2** Quellenwahl für das Istwertquellen-Multiplexer-Ausgangssignal (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Istwert A , FSP020_ActualValue_A
0x2	Istwert B , FSP021_ActualValue_B
0x3	Ausgang interner Funktionsgenerator
0x4	Summand 1 für den Addierer, bestimmt durch "FSP090_Adder_SourceSelectionMultiplexer[8][3..0]"
0x5	Controller 1 _SetValueDeviation, analog FSP076_Controller_1_SetValueDeviation
0x6	Controller X _PreSelectionMux_1_Out, bestimmt durch FSP064_USIxHS_Multiplexer[99..96]
0x7	Controller X _PreSelectionMux_2_Out, bestimmt durch FSP064_USIxHS_Multiplexer[103..100]
0x8	Sollwert A , FSP030_SetValue_A
0x9	...
0xF	n.u., immer GND_BUS

[11..8] Regler **2** Quellenwahl für Sollwertquellen-Multiplexer-Ausgangssignal (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A , bestimmt durch „FSP013_PeripheralConfig[0]“
0x2	Sollwert B , bestimmt durch „FSP013_PeripheralConfig[0]“
0x3	Sollwert A , FSP030_SetValue_A
0x4	Ausgang interner Funktionsgenerator
0x5	Summand 1 für den Addierer, bestimmt durch "FSP090_Adder_SourceSelectionMultiplexer[16][3..0]"
0x6	Controller 1 _SetValueDeviation, analog FSP076_Controller_1_SetValueDeviation
0x7	Controller X _PreSelectionMux_1_Out, bestimmt durch FSP064_USIxHS_Multiplexer[99..96]
0x8	Controller X _PreSelectionMux_2_Out, bestimmt durch FSP064_USIxHS_Multiplexer[103..100]
0x9	Controller 1 _SetValueMuxOut[19..0], bestimmt durch FSP070_Controller_1_2_InputSourceSelectionMultiplexer[3..0]

0xA	Controller1_PI_Part_Output[19..0]
0xB	n.u., immer GND_BUS
...	
0xF	

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

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Istwert A , bestimmt durch FSP020_ActualValue_A
0x2	Istwert B , bestimmt durch FSP021_ActualValue_B
0x3	Ausgang interner Funktionsgenerator
0x4	ControllerX_PreSelectionMux_1_Out, bestimmt durch FSP064_USIxHS_Multiplexer[99..96]
0x5	ControllerX_PreSelectionMux_2_Out, bestimmt durch FSP064_USIxHS_Multiplexer[103..100]
0x6	Sollwert A , FSP030_SetValue_A
0x7	n.u., immer GND_BUS
...	
0xF	

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

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A , bestimmt durch „FSP013_PeripheralConfig[0]“
0x2	Sollwert B , bestimmt durch „FSP013_PeripheralConfig[0]“
0x3	Ausgang interner Funktionsgenerator
0x4	ControllerX_PreSelectionMux_1_Out, bestimmt durch FSP064_USIxHS_Multiplexer[99..96]
0x5	ControllerX_PreSelectionMux_2_Out, bestimmt durch FSP064_USIxHS_Multiplexer[103..100]
0x6	Sollwert A , FSP030_SetValue_A
0x7	n.u., immer GND_BUS
...	
0xF	

Name	FSP071_Controller_1_SetValue
Adresse	0x47_H/71_D/0x3731_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann der Sollwert des PI Regler **1** ausgelesen werden. Der Sollwert wird dabei über den Multiplexer „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“ ausgewählt.

Das Äquivalent für Regler **2** ist der „FSP081_Controller_2_SetValue“.

[23..0] Vorzeichenbehafteter 20 Bit Sollwert des PI Regler **1**, die Bits[3..0] sind immer ,0‘.

Name	FSP072_Controller_1_ActualValue
Adresse	0x48_H/72_D/0x3438_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann der Istwert des PI Regler **1** ausgelesen werden. Der Istwert wird dabei über den Multiplexer „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“ ausgewählt.

Das Äquivalent für Regler **2** ist der „FSP082_Controller_2_ActualValue“.

[23..0] vorzeichenbehafteter 20 Bit Istwert des PI Regler **1**, die Bits[3..0] sind immer ,0‘.

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.

Das Äquivalent für Regler **2** ist der „FSP083_Controller_2_Limits“.

[47..24] Controller_1_MaxVal (20 Bit), repräsentiert den oberen (maximalen) Grenzwert des PI Regler **1**, die Bits[3..0] sind immer ,0‘.

[23..0] Controller_1_MinVal (20 Bit), repräsentiert den unteren (minimalen) Grenzwert des PI Regler **1**, die Bits[3..0] sind immer ,0‘.

Name	FSP074_Controller_1_PI_Settings
Adresse	0x4A_H/60_D/0x3441_{ASCII}
Tiefe	13 Byte / 104 Bit
I/O	lesen / schreiben
Reset	0x00_00000000_00000000_00000000 _H

Über diesen FSP können die I und P1, P2 Anteile des Regler **1** gesetzt werden.

Das Äquivalent für Regler **2** ist der „FSP084_Controller_2_PI_Settings“.

[103..97] n.u.

[96] Controller_1_PI_Control - Wenn ‚1‘ wird der I-Anteil des Reglers um den Faktor 1000 verlangsamt.

[95..64] Controller_1_I_Part (32 Bit), repräsentiert den I Anteil des PI Regler **1**.

[63..32] Controller_1_P2_Part (32 Bit), repräsentiert den 2. P Anteil des PI Regler **1**.

[63..18] Vorkommastelle (Integer)

[17..0] Nachkommastelle

[31..0] Controller_1_P1_Part (32 Bit), repräsentiert den 1. P Anteil des PI Regler **1**.

[31..20] Vorkommastelle (Integer)

[19..0] Nachkommastelle

Name	FSP075_Controller_1_I_Part_ComparatorLimits
Adresse	0x4B_H/75_D/0x3442_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Repräsentiert die Bereichsgrenzen für Regler **1**, in denen der I Anteil bei der Regelung berücksichtigt werden soll.

Das Äquivalent für Regler **2** ist der „FSP085_Controller2_I_Part_ComparatorLimits“.

[47..24] Controller_1_I_Part_ComparatorOFFThreshold (20 Bit), die Bits[3..0] sind immer ‚0‘.

[23..0] Controller_1_I_Part_ComparatorONThreshold (20 Bit), die Bits[3..0] sind immer ‚0‘.

Name	FSP076_Controller_1_SetValueDeviation
Adresse	0x4C_H/76_D/0x3443_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann die Regeldifferenz (Delta I) des PI Regler **1** gelesen werden. Die Regeldifferenz wird dabei im Differenzbildner des Regler **1** aus dem Minuend SetValue (Ausgang: „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“, bzw. „FSP071_Controller_1_SetValue“) und dem Subtrahend ActualValue (Ausgang: „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“, bzw. „FSP072_Controller_1_ActualValue“).

Das Äquivalent für Regler **2** ist der „FSP086_Controller_2_SetValueDeviation“.

[23..0] SetValueDeviation (20 Bit), die Bits[3..0] sind immer ,0‘.

Name	FSP077_Controller_1_PI_Output
Adresse	0x4D_H/77_D/0x3444_{ASCII}
Tiefe	9 Byte / 72 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP können der PI-Anteil, I-Anteil und P-Anteil des PI Regler **1** gelesen werden

Das Äquivalent für Regler **2** ist der „FSP087_Controller_2_PI_Output“.

[71..48] Controller_1_P_Part_Output(20 Bit), die Bits[3..0] sind immer ,0‘.

[47..24] Controller_1_I_Part_Output (20 Bit), die Bits[3..0] sind immer ,0‘.

[23..0] Controller_1_PI_Output (20 Bit), die Bits[3..0] sind immer ,0‘.

Name	FSP078_Controller_1_P2_Part_ComparatorLimits
Adresse	0x4E_H/78_D/0x3445_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Repräsentiert die Bereichsgrenzen in denen der P2 Anteil bei der Regelung des Regler **1** berücksichtigt werden soll.

Das Äquivalent für Regler **2** ist der „FSP088_Controller_2_P2_Part_ComparatorLimits“.

[47..24] Controller_1_P2_Part_ComparatorOFFThreshold (20 Bit), die Bits[3..0] sind immer '0':

[23..0] Controller_1_P2_Part_ComparatorONThreshold (20 Bit), die Bits[3..0] sind immer '0':

Name	FSP079_Controller_1_SlopeLimiterOutput
Adresse	0x4F_H/79_D/0x34465_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Liefert den Ausgangswert des Steilheitsbegrenzers für Regler **1**

Das Äquivalent für Regler **2** ist der „FSP089_Controller_2_SlopeLimiterOutput“.

[23..0] SlopeLimiterOutput Regler **1** (20 Bit), die Bits[3..0] sind immer '0':

Name	FSP080_SlopeLimiter_C2
Adresse	0x3C_H/60_D/0x3343_{ASCII}
Tiefe	30 Byte / 240 Bit
I/O	lesen / schreiben
Reset	0x00_000000_000000_000000_000000_000000_0000_0000_0000_0000_ 745D17_8BA2E8 _H

Repräsentiert Parameter des Steilheitsbegrenzers.

Das Äquivalent für den Steilheitsbegrenzer **1** findet sich im „FSP060_SlopeLimiter_C1“.

[239..234] n.u.

[233] Aktiviert den Steilheitbegrenzer,
wenn ‚1‘ UND ‚ACU_ControllerEnable‘

[232] Wenn ‚0‘ wird der Startwert des Steilheitbegrenzers mit 0 geladen, andernfalls der Wert
des Eingangs ‚Dynamic_Threshold_RiseTime_Selection‘.
Dies ist i.d.R. ‚Controller1_ActValueMuxOut‘

[231..216] RampThresholdValue_D (16 Bit)

[215..200] RampThresholdValue_C (16 Bit)

[199..184] RampThresholdValue_B (16 Bit)

[183..168] RampThresholdValue_A (16 Bit)

[167..144] RampRiseTime_E (24 Bit)

[143..120] RampRiseTime_D (24 Bit)

[119..96] RampRiseTime_C (24 Bit)

[95..72] RampRiseTime_B (24 Bit)

[71..48] RampRiseTime_A (24 Bit)

[47..24] Obere Grenze (24 Bit)

[23..0] Untere Grenze (24 Bit)

Name	FSP081_Controller_2_SetValue
Adresse	0x51_H/81_D/0x3531_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann der Sollwert des PI Regler **2** ausgelesen werden. Der Sollwert wird dabei über den Multiplexer „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“ ausgewählt.

Das Äquivalent für Regler **1** ist der „FSP071_Controller_1_SetValue“.

[23..0] vorzeichenbehafteter 20 Bit Sollwert des PI Regler **2**, die Bits[3..0] sind immer ,0‘.

Name	FSP082_Controller_2_ActualValue
Adresse	0x52_H/82_D/0x3532_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann der Istwert des PI Regler **2** ausgelesen werden. Der Istwert wird dabei über den Multiplexer „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“ ausgewählt.

Das Äquivalent für Regler **1** ist der „FSP072_Controller_1_ActualValue“.

[23..0] vorzeichenbehafteter 20 Bit Istwert des PI Regler **2**, die Bits[3..0] sind immer ,0'.

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.

Das Äquivalent für Regler **1** ist der „FSP073_Controller_1_Limits“.

[47..24] Controller2_MaxVal (20 Bit), repräsentiert den oberen (maximalen) Grenzwert des PI Regler **2**, die Bits[3..0] sind immer ,0‘.

[23..0] Controller2_MinVal (20 Bit), repräsentiert den unteren (minimalen) Grenzwert des PI Regler **2**, die Bits[3..0] sind immer ,0‘.

Name	FSP084_Controller_2_PI_Settings
Adresse	0x54_H/84_D/0x3534_{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.

Das Äquivalent für Regler **1** ist der „FSP074_Controller_1_PI_Settings“.

[103..96] Controller_2_PI_Control

[103..97] n.u.

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

[95..64] Controller_2_I_Part (32 Bit), repräsentiert den I Anteil des PI Regler **2**.

[63..32] Controller_2_P2_Part (32 Bit), repräsentiert den 2. P Anteil des PI Regler **2**.

[31..0] Controller_2_P1_Part (32 Bit), repräsentiert den 1. P Anteil des PI Regler **2**.

Name	FSP085_Controller2_I_Part_ComparatorLimits
Adresse	0x55_H/85_D/0x3535_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Repräsentiert die Bereichsgrenzen für Regler **2**, in denen der I Anteil bei der Regelung berücksichtigt werden soll.

Das Äquivalent für Regler **1** ist der „FSP075_Controller_1_I_Part_ComparatorLimits“.

[47..24] Controller_2_I_Part_ComparatorOFFThreshold (20 Bit), die Bits[3..0] sind immer ‚0‘.

[23..0] Controller_2_I_Part_ComparatorONThreshold (20 Bit), die Bits[3..0] sind immer ‚0‘.

Name	FSP086_Controller_2_SetValueDeviation
Adresse	0x56_H/86_D/0x3536_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann die Regeldifferenz (Delta I) des PI Regler **2** gelesen werden. Die Regeldifferenz wird dabei im Differenzbildner des Regler **2** aus dem Minuend SetValue (Ausgang: „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“, bzw. „FSP081_Controller_2_SetValue“) und dem Subtrahend ActualValue (Ausgang: „FSP070_Controller_1_2_InputSourceSelectionMultiplexer“, bzw. „FSP082_Controller_2_ActualValue“).

Das Äquivalent für Regler **1** ist der „FSP076_Controller_1_SetValueDeviation“.

[23..0] SetValueDeviation (20 Bit), die Bits[3..0] sind immer ,0‘.

Name	FSP087_Controller_2_PI_Output
Adresse	0x57_H/87_D/0x3837_{ASCII}
Tiefe	9 Byte / 72 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP können der PI-Anteil, I-Anteil und P-Anteil des PI Regler **2** gelesen werden

Das Äquivalent für Regler **1** ist der „FSP077_Controller_1_PI_Output“.

[71..48] Controller_2_P_Part_Output(20 Bit), die Bits[3..0] sind immer ,0‘.

[47..24] Controller_2_I_Part_Output (20 Bit), die Bits[3..0] sind immer ,0‘.

[23..0] Controller_2_PI_Output (20 Bit), die Bits[3..0] sind immer ,0‘.

Name	FSP088_Controller_2_P2_Part_ComparatorLimits
Adresse	0x58_H/88_D/0x3838_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Repräsentiert die Bereichsgrenzen in denen der P2 Anteil bei der Regelung des Regler **2** berücksichtigt werden soll.

Das Äquivalent für Regler **1** ist der „FSP078_Controller_1_P2_Part_ComparatorLimits“.

[47..24] Controller_2_P2_Part_ComparatorOFFThreshold (20 Bit), die Bits[3..0] sind immer '0':

[23..0] Controller_2_P2_Part_ComparatorONThreshold (20 Bit), die Bits[3..0] sind immer '0':

Name	FSP089_Controller_2_SlopeLimiterOutput
Adresse	0x59_H/89_D/0x3839_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Liefert den Ausgangswert des Steilheitsbegrenzers für Regler **2**

Das Äquivalent für Regler **1** ist der „FSP079_Controller_1_SlopeLimiterOutput“.

[23..0] SlopeLimiterOutput Regler **2** (20 Bit), die Bits[3..0] sind immer '0':

Name	FSP090_Adder_SourceSelectionMultiplexer
Adresse	0x5A_H/90_D/0x3541_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x00_0_0 _H

Repräsentiert die Einstellungen der Addierer Eingangsmultiplexer für dessen Summanden **1** und **2**.

[15..10] n.u.

[9] '1' Ausgangswert des Multiplexers [Adder_Summand_2] wird invertiert

[8] '1' Ausgangswert des Multiplexers [Adder_Summand_1] wird invertiert

[7..4] Quellenwahl für Multiplexer Ausgangssignal [Adder_Summand_2] (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert B , bestimmt durch FSP031_SetValue_B
0x2	Controller2_SetValueDeviation, analog FSP086_Controller_2_SetValueDeviation
0x3	Controller2_PI_Part_Output
0x4	SlopeLimiterOutput_C2[19..0]
0x5	Controller2_SetValueMuxOut[19..0], bestimmt durch FSP070_Controller_1_2_InputSourceSelectionMultiplexer[18][11..8]
0x6	n.u., immer GND_BUS[19..0]
...	
0xF	

[3..0] Quellenwahl für Multiplexer Ausgangssignal [Adder_Summand_1] (4 Bit)

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Sollwert A , bestimmt durch FSP030_SetValue_A
0x2	SlopeLimiterOutput_C1[19..0]
0x3	Controller1_SetValueDeviation, analog FSP076_Controller_1_SetValueDeviation
0x4	Controller1_PI_Part_Output
0x5	Controller1_SetValueMuxOut[19..0], bestimmt durch FSP070_Controller_1_2_InputSourceSelectionMultiplexer[16][3..0]
0x6	n.u., immer GND_BUS[19..0]
...	
0xF	

Name	FSP091_Adder_Limits
Adresse	0x5B_H/91_D/0x3542_{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 Addierers. Die Bits[3..0] sind immer ‚0‘.
- [23..0] Adder_MinVal (20 Bit), repräsentiert den unteren (minimalen) Grenzwert des Addierers. Die Bits[3..0] sind immer ‚0‘.

Name	FSP092_Adder_SumOut
Adresse	0x5C_H/92_D/0x3543_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Über diesen FSP kann die Summe des Addierwerkes gelesen werden. Die Summanden werden dabei über „FSP090_Adder_SourceSelectionMultiplexer“ die Bereichsgrenzen über „FSP091_Adder_Limits“ festgelegt.

[23..0] Adder_SumOut (20 Bit), die Bits[3..0] sind immer ‚0‘.

Name	FSP093_CorrFactorPI_Limits
Adresse	0x5D_H/93_D/0x3544_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Über diesen FSP werden die Bereichsgrenzen des CorrectorFactor PI Regler festgelegt.

- [47..24] MaxVal (20 Bit), repräsentiert den oberen (maximalen) Grenzwert.
Die Bits[3..0] sind immer ‚0‘.
- [23..0] MinVal (20 Bit), repräsentiert den unteren (minimalen) Grenzwert.
Die Bits[3..0] sind immer ‚0‘.

Name	FSP094_CorrFactorPI_kP
Adresse	0x5E_H/94_D/0x3545_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen / schreiben
Reset	0x00000000 _H

Über diesen FSP können die P Anteile des CorrectionFactor Reglers gesetzt werden.

[31..0] CorrFactorPI_kP (32 Bit), repräsentiert den P Anteil des CorrFactorPI.

Name	FSP095_ComparatorControl
Adresse	0x5F_H/95_D/0x3546_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x0000 _H

Repräsentiert verschiedene Komparator-Kontrollzustände in der MFU.

- [15..13] n.u.
- [12] ,1' aktiviert den Komparator zum Aktivieren/Deaktivieren folgender Funktion, sofern die Komparatorgrenzen „FSP103_PWM_FDrive2_ComparatorLimits“ durch die Regelabweichung des Regler 2 über-, bzw. unterschritten werden und Degaussing (Entmagnetisierung) aktiviert ist:

Es wird bei Komparatorauslösung anstelle des Ergebnis der Addierstufe „FSP092_Adder_SumOut“ der Wert „FSP105_IGBT_AlternateSet“ als Steuerwert der Rückspeisungsbegrenzung verwendet.
- [11] ,1' aktiviert den Komparator zum Aktivieren/Deaktivieren folgender Funktion, sofern die Komparatorgrenzen „FSP102_PWM_FDrive1_ComparatorLimits“ durch die Regelabweichung des Regler 1 über-, bzw. unterschritten werden und Degaussing (Entmagnetisierung) nicht aktiviert ist:

Es wird bei Komparatorauslösung anstelle des Ergebnis der Addierstufe „FSP092_Adder_SumOut“ der Wert „FSP105_IGBT_AlternateSet“ als Steuerwert Rückspeisungsbegrenzung verwendet.
- [10] ,1' aktiviert die Rückspeisungsbegrenzung
- [9] n.u.
- [8] ,1' aktiviert den Komparator zum Aktivieren/Deaktivieren des P2-Anteils beim Regler 2, sofern dieser die Komparatorgrenzen „FSP088_Controller_2_P2_Part_ComparatorLimits“ durch die Regelabweichung des Regler 2 über-, bzw. unterschreitet.
- [7] ,1' aktiviert den Komparator zum Aktivieren/Deaktivieren des I-Anteils beim Regler 2, sofern dieser die Komparatorgrenzen „FSP085_Controller2_I_Part_ComparatorLimits“ durch die Regelabweichung des Regler 2 über-, bzw. unterschreitet.
- [6] n.u.
- [5] ,1' aktiviert die Degaussing- (Entmagnetisierungs)funktion und den Komparator für deren Durchführung, sofern dieser die Komparatorgrenzen „FSP101_Degauss_ComparatorLimit“ durch den Sollwert des Regler 1 über-, bzw. unterschreitet.
- [4] n.u.
- [3] ,1' aktiviert den Komparator zum Aktivieren/Deaktivieren folgender Funktion, sofern die Komparatorgrenzen „FSP100_V5_ComparatorLimits“ durch die Regelabweichung des Regler 1 über-, bzw. unterschritten werden und Degaussing (Entmagnetisierung) nicht aktiviert ist:

,1' aktiviert den Komparator zum Aktivieren/Deaktivieren folgender Funktion, sofern die Komparatorgrenzen „FSP100_V5_ComparatorLimits“ durch die Regelabweichung des Regler 2 über-, bzw. unterschritten werden und Degaussing (Entmagnetisierung) aktiviert ist:

Es wird bei Komparatorauslösung IGBT V5 aktiviert, abhängig vom aktiven Degaussing. Ist dieses aktiv, wird der Komparatorausgang von Regler 2 benutzt, sonst von Regler 1.
- [2] n.u.

- [1] ,1' aktiviert den Komparator zum Aktivieren/Deaktivieren des P2-Anteils beim Regler **1**, sofern die Reglerabweichung des Regler **1** „FSP078_Controller_1_P2_Part_ComparatorLimits“ über-, bzw. unterschreitet.
- [0] ,1' aktiviert den Komparator zum Aktivieren/Deaktivieren des I-Anteils beim Regler **1**, sofern dieser die Komparatorgrenzen „FSP075_Controller_1_I_Part_ComparatorLimits“ über-, bzw. unterschreitet.

Name	FSP096_SerialComDt2SendSel
Adresse	0x60_H/096_D/0x360_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen/schreiben
Reset	0x(siehe Beschreibung) _H

It selects the data to be sent for all 4 LWL channels.

Das ErrGenCfg FSP für SerialCom_1 ist „FSP108_SerialCom_1_ErrorGenCfg“.
 Das ErrGenCfg FSP für SerialCom_2 ist „FSP111_SerialCom_2_ErrorGenCfg“.
 Das ErrGenCfg FSP für SerialCom_3 ist „FSP113_SerialCom_3_ErrorGenCfg“.
 Das ErrGenCfg FSP für SerialCom_4 ist „FSP123_SerialCom_4_ErrorGenCfg“.

Der Status für SerialCom_1 ist „FSP109_SerialCom_1_Status“.
 Der Status für SerialCom_2 ist „FSP112_SerialCom_2_Status“.
 Der Status für SerialCom_3 ist „FSP122_SerialCom_3_Status“.
 Der Status für SerialCom_4 ist „FSP124_SerialCom_4_Status“.

[15..12] Channel4 data to send selector

[3..0]	Ausgang des Multiplexers
0x0	Zeros
0x1	Counter incremented at 2 MHz
0x2	Counter incremented at 1 MHz
0x3	Counter incremented at 500KHz
0x4	Counter incremented at 250KHz
0x5	Function generator 1 (SCU)
0x6	Channel4 received payload
0x7	Function generator 2 (SCU)
0x8	SetValueA (FSP030)
0x9	LocalSetValue
0xA	Internal function generator
0xB	ERLOutValue
0xC	n.u., immer GND_BUS
...	
0xF	

[11..8] Channel3 data to send selector

[3..0]	Ausgang des Multiplexers
0x0	Zeros
0x1	Counter incremented at 2 MHz
0x2	Counter incremented at 1 MHz
0x3	Counter incremented at 500KHz
0x4	Counter incremented at 250KHz
0x5	Function generator 1 (SCU)
0x6	Channel3 received payload
0x7	Function generator 2 (SCU)
0x8	SetValueA (FSP030)
0x9	LocalSetValue
0xA	Internal function generator
0xB	ERLOutValue
0xC	n.u., immer GND_BUS
...	

	0xF	
[7..4]	Channel2 data to send selector	
	[3..0]	Ausgang des Multiplexers
	0x0	Zeros
	0x1	Counter incremented at 2 MHz
	0x2	Counter incremented at 1 MHz
	0x3	Counter incremented at 500KHz
	0x4	Counter incremented at 250KHz
	0x5	Function generator 1 (SCU)
	0x6	Channel2 received payload
	0x7	Function generator 2 (SCU)
	0x8	SetValueA (FSP030)
	0x9	LocalSetValue
	0xA	Internal function generator
	0xB	ERLOutValue
	0xC	n.u., immer GND_BUS
	...	
	0xF	

[3..0]	Channel1 data to send selector	
	[3..0]	Ausgang des Multiplexers
	0x0	Zeros
	0x1	Counter incremented at 2 MHz
	0x2	Counter incremented at 1 MHz
	0x3	Counter incremented at 500KHz
	0x4	Counter incremented at 250KHz
	0x5	Function generator 1 (SCU)
	0x6	Channel1 received payload
	0x7	Function generator 2 (SCU)
	0x8	SetValueA (FSP030)
	0x9	LocalSetValue
	0xA	Internal function generator
	0xB	ERLOutValue
	0xC	n.u., immer GND_BUS
	...	
	0xF	

Name	FSP097_SelVal2CompP2Comp
Adresse	0x61_H/97_D/0x3631_{ASCII}
Tiefe	1 Byte / 8 Bit
I/O	lesen / schreiben
Reset	0x00 _H

Über dieses FSP kann für die Komparatoren zum aktivieren/deaktivieren des P2 Teils der PI-Regler **1** und **2** der zu vergleichende Wert gewählt werden.

Im normalen Modus ist dies immer die errechnete Reglabweichung zwischen Soll- und Istwert des Reglers zugehörigen Diffenzbilders.

Im ExpertMode können andere Vergleichswerte ausgewählt werden.

[7..4] Legt den Vergleichswert für den P2 Komparator des Regler **1** fest.

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Controller1_SetValueDeviation[19..0]
0x2	Controller2_SetValueDeviation[19..0]
0x3	SlopeLimiterOutputC1[19..0]
0x4	Controller2_SetValueMuxOut[19..0]
0x5	Controller1_ActValueMuxOut[19..0]
0x6	Controller2_ActValueMuxOut[19..0]
0x7	SlopeLimiterOutput_C2[19..0]
0x8 ... 0xF	n.u., immer GND_BUS

[3..0] Legt den Vergleichswert für den P2 Komparator des Regler **1** fest.

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Controller1_SetValueDeviation[19..0]
0x2	Controller2_SetValueDeviation[19..0]
0x3	SlopeLimiterOutputC1[19..0]
0x4	Controller2_SetValueMuxOut[19..0]
0x5	Controller1_ActValueMuxOut[19..0]
0x6	Controller2_ActValueMuxOut[19..0]
0x7	SlopeLimiterOutput_C2[19..0]
0x8 ... 0xF	n.u., immer GND_BUS

Name	FSP098_Selectable_klKp1
Adresse	0x62_H/98_D/0x3632_{ASCII}
Tiefe	32 Byte / 256 Bit
I/O	lesen / schreiben
Reset	0x0000000000000000_0000000000000000_0000000000000000_0000000000000000 _H

This FSP contains the 4 of 5 selectable coefficients (kl, kP1) for the controller number one. The first coefficients set is stored in the FSP

- [255..192] Fifth Coefficients set 64=32+32 bits.
 - [63..32] kl_5 coefficient
 - [31..0] kP1_5 coefficient
- [191..128] Fourth Coefficients set 64=32+32 bits.
 - [63..32] kl_4 coefficient
 - [31..0] kP1_4 coefficient
- [127..64] Third Coefficients set 64=32+32 bits.
 - [63..32] kl_3 coefficient
 - [31..0] kP1_3 coefficient
- [63..0] Second Coefficients set 64=32+32 bits.
 - [63..32] kl_2 coefficient
 - [31..0] kP1_2 coefficient

Name	FSP099_Selectable_klkP1Thresholds
Adresse	0x63_H/99_D/0x3633_{ASCII}
Tiefe	25 Byte / 200 Bit
I/O	lesen / schreiben
Reset	00_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000 _H

This FSP contains the thresholds (on and off) used to generate the coefficients multiplexer selector. The SelectorSignal is compared with these thresholds and a decoded signal is generated for the purpose above described. The FSP contains also the Enable bit used to turn on/off the coefficientSelector module. In off mode, the first coefficients set is always selected (module back compatible)

[199..192] FSP99_Selectable_klkP_Enable[7..0]

[191..0] FSP99_Selectable_klkP_Thresholds

[199..193] n.u.

[192] Enabled: ACU_CoefficientSelector

0=> The first coefficients set are selected to the output (transparent mode).

1=> The output coefficients take one of the 5 available inputs based on the SelectorSignal value.

[191..168] Fourth **Off** Threshold value (20b), die unteren 4 Bit sind immer '0'.

[167..144] Third **Off** Threshold value (20b), die unteren 4 Bit sind immer '0'.

[143..120] Second **Off** Threshold value (20b), die unteren 4 Bit sind immer '0'.

[119..96] First **Off** Threshold value (20b), die unteren 4 Bit sind immer '0'.

[95..72] Fourth **On** Threshold value (20b), die unteren 4 Bit sind immer '0'.

[71..48] Third **On** Threshold value (20b), die unteren 4 Bit sind immer '0'.

[47..24] Second **On** Threshold value (20b), die unteren 4 Bit sind immer '0'.

[23..4] First **On** Threshold value (20b), die unteren 4 Bit sind immer '0'.

Name	FSP100_V5_ComparatorLimits
Adresse	0x64_H/100_D/0x3634_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Legt die Grenzwerte des Komparators zur Aktivierung/Deaktivierung von IGBT V5 fest.

[47..24] V5_Comparator**OFF**Threshold (20 Bit), die Bit[3..0] sind immer '0'.

[23..0] V5_Comparator**ON**Threshold (20 Bit), die Bit[3..0] sind immer '0'.

Name	FSP101_Degauss_ComparatorLimit
Adresse	0x65_H/101_D/0x3635_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0xFFFF6A4_00095B _H

Repräsentiert die Schwelle des Komparators für das Entmagnetisieren (Degaussen)

[47..24] Degauss_Comparator**OFF**Threshold (20 Bit), die Bit[3..0] sind immer '0'.

[23..0] Degauss_Comparator**ON**Threshold (20 Bit), die Bit[3..0] sind immer '0'.

Name	FSP102_PWM_FDrive1_ComparatorLimits
Adresse	0x66_H/102_D/0x3636_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Legt die Grenzwerte des Komparators zur Aktivierung/Deaktivierung des alternativen Steuerwerts der Rückspeisungsbegrenzung fest, sofern Degaussen (Entmagnetisieren) nicht aktiv ist.

[47..24] PWM_FDRIVE1_Comparator**OFF**Threshold (20 Bit), die Bit[3..0] sind immer '0'.

[23..0] PWM_FDRIVE1_Comparator**ON**Threshold (20 Bit), die Bit[3..0] sind immer '0'.

Name	FSP103_PWM_FDrive2_ComparatorLimits
Adresse	0x67_H/103_D/0x3637_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Legt die Grenzwerte des Komparators zur Aktivierung/Deaktivierung des alternativen Steuerwerts der Rückspeisungsbegrenzung fest, sofern Degaussen (Entmagnetisieren) aktiv ist.

[47..24] PWM_FDRIVE2_Comparator**OFF**Threshold (20 Bit), die Bit[3..0] sind immer '0'.

[23..0] PWM_FDRIVE2_Comparator**ON**Threshold (20 Bit), die Bit[3..0] sind immer '0'.

Name	FSP105_IGBT_AlternateSetValue
Adresse	0x69_H/105_D/0x3639_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x000000 _H

Repräsentiert den alternativen Steuerwert für die Rückspeisungsbegrenzung, sofern einer der Komparatoren die Grenzen „FSP102_PWM_FDrive1_ComparatorLimits“, bei nicht aktivem Degaussing (Entmagnetisieren), bzw. „FSP103_PWM_FDrive2_ComparatorLimits“, bei aktivem Degaussing (Entmagnetisieren) unter-, bzw. überschritten hat.

[23..0] IGBT_AlternateSetValue (20 Bit), die Bit[3..0] sind immer '0'.

Name	FSP106_EnergyRecoverLimitation_CurrentDriveValue
Adresse	0x6A_H/106_D/0x3641_{ASCII}
Tiefe	6 Byte / 48 Bit
I/O	lesen / schreiben
Reset	0x000000_000000 _H

Funktion

[47..24] ERL_CurrentThresholdValue (20 Bit)

[23..0] ERL_CurrentDriveValue (20 Bit)

Name	FSP107_CtrlEnable_Delay
Adresse	0x6B_H/107_D/0x3642_{ASCII}
Tiefe	1 Byte / 8 Bit
I/O	lesen / schreiben
Reset	0x10 _H

[7..5] n.u.

[5] Enable

[3..0] DelayValue * 10us

Name	FSP108_SerialCom_1_ErrorGenCfg
Adresse	0x6C_H/108_D/0x3643_{ASCII}
Tiefe	5 Byte / 40 Bit
I/O	lesen/schreiben
Reset	0x000000000 _H

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

Das Äquivalent für SerialCom_2 ist „FSP111_SerialCom_2_ErrorGenCfg“.

Das Äquivalent für SerialCom_3 ist „FSP113_SerialCom_3_ErrorGenCfg“.

Das Äquivalent für SerialCom_4 ist „FSP123_SerialCom_4_ErrorGenCfg“.

Der Status für SerialCom_1 ist im „FSP109_SerialCom_1_Status“.

Den Multiplexer für die zu sendenden Daten bedient „FSP096_SerialComDt2SendSel“.

[39..33] **Selected received payload** ^{*)}

[3..0]	Ausgang des Multiplexers
0x0	Disabled
0x1	PayloadRx_1_data[23..0]
0x2	PayloadRx_2_data[23..0]
0x3	PayloadRx_3_data[23..0]
0x4	PayloadRx_4_data[23..0]
0x5	n.u.
..	
0xF	

^{*)} Diese Signale stehen erst ab MFU LWL FW 7.1. zur Verfügung. Vorher n.u.

[35..33] n.u.

[32] Reset errors

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

[23..21] n.u.

[20] Error injection enable(active high)

[19..17] n.u.

[16] Change symbol flag

[15..13] n.u.

[12..8] Symbol mask

[7..0] Symbol number to corrupt

Name	FSP109_SerialCom_1_Status
Adresse	0x6D_H/109_D/0x3644_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Sammelt die Alarme für den LWL-Kanal **1**.

Das Äquivalent für SerialCom_2 ist „FSP112_SerialCom_2_Status“.

Das Äquivalent für SerialCom_3 ist „FSP122_SerialCom_3_Status“.

Das Äquivalent für SerialCom_4 ist „FSP124_SerialCom_4_Status“.

Das „Debugging“ der SerialCom_1 erfolgt im „FSP108_SerialCom_1_ErrorGenCfg“.

Den Multiplexer für die zu sendenden Daten bedient „FSP096_SerialComDt2SendSel“.

[23..17] n.u.

[16] PayloadAlarm_1

[15] LostConnection_1

[14] IllSymbComFailed_1

[13] PckgDis_1

[12] ParityComFailed_1

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

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

Name	FSP110_DACx_and_ScopeChannelx_SourceSelectionMultiplexer
Adresse	0x6E_H/110_D/0x3645_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x00_0_1_1_1 _H

Repräsentiert die Einstellungen der DAC Ausgangs- und int. Oszilloskop Eingangsmultiplexer.

Sowohl die Ausgabequellen der DACs, als auch die Eingangsquellen des internen Oszilloskops werden über einen einzigen Multiplexer geführt. Das interne Oszilloskop hat 4 Kanäle, die dabei jeweils den nachfolgenden Quellen entsprechen.

[23..20] n.u.

[19] '1' Ausgangswert des Multiplexers [DAC_4_and_ScopeChannel_4_Source]
→ X8 wird invertiert

[18] '1' Ausgangswert des Multiplexers [DAC_3_and_ScopeChannel_3_Source]
→ X7 wird invertiert

[17] '1' Ausgangswert des Multiplexers [DAC_2_and_ScopeChannel_2_Source]
→ X6 wird invertiert

[16] '1' Ausgangswert des Multiplexers [DAC_1_and_ScopeChannel_1_Source]
→ X5 wird invertiert

[15..12] Quellenwahl für Multiplexer Ausgangssignal [DAC_4_and_ScopeChannel_4_Source] → X8

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Controller1_SetValueMuxOut[19..0], Sollwert Regler 1, bestimmt durch „FSP070_Controller_1_2_InputSourceSelectionMultiplexer[16][3..0]“
0x2	Controller2_SetValueMuxOut[19..0], Sollwert Regler 2, bestimmt durch „FSP070_Controller_1_2_InputSourceSelectionMultiplexer[18][11..8],“
0x3	DAC_and_Scope_PreSelectionMux_1_Out[19..0], bestimmt durch “FSP064_USIxHS_Multiplexer[83..80]”
0x4	DAC_and_scope_PreSelectionMux_2_Out[19..0], bestimmt durch “FSP064_USIxHS_Multiplexer[87..84]”
0x5	Controller1_P_Part_Ouput[19..0]
0x6	Controller1_I_Part_Ouput[19..0]
0x7	Controller1_PI_Part_Ouput[19..0]
0x8	Controller2_P_Part_Ouput[19..0]
0x9	Controller2_I_Part_Ouput[19..0]
0xA	Controller2_PI_Part_Ouput[19..0]
0xB	Adder_SumOut[19..0]
0xC	ERLOutValue, Rückspeisungsbegrenzung[19..0]
0xD	PayloadRx_4_data[23..4]
0xE	intFuncGenOutput[19..0]
0xF	ControllerStatusBits[19..0] entspricht dem FSP121_ControllerStatusBits LE/SE [19] Controller_1_I_Part_Disable [18] Controller_2_I_Part_Disable [17] Controller_1_P2_Part_Active [16] Controller_2_P2_Part_Active [15] Degaussing_Active [14] Controller_1_PWM_FDrive_Comparator [13] Controller_2_PWM_FDrive_Comparator

[12]	PWM_FDriveComparator_MuxOut
[11]	Controller_1_IGBT_V5_Active
[10]	Controller_2_IGBT_V5_Active
[9]	IGBT_V5_Active
[8]	n.u., immer ,0'
[7]	ACU_Controller_Enable
[6]	SlopeLimiterEnable_C1
[5]	n.u., immer ,0'
[4]	SlopeLimiterEnable_C2
[3..0]	n.u., immer ,0'

[11..8] Quellenwahl für Multiplexer Ausgangssignal [DAC_3_and_ScopeChannel_3_Source] → X7

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Controller1_MultipliedDeviation[19..0], Ausgang des MFU_DifferenceCalculator, Regler 1, multiplizierte Reglerdifferenz
0x2	Controller2_MultipliedDeviation[19..0], Ausgang des MFU_DifferenceCalculator, Regler 2, multiplizierte Reglerdifferenz
0x3	DAC_and_scope_PreSelectionMux_1_Out[19..0], bestimmt durch "FSP064_USIxHS_Multiplexer[83..80]"
0x4	DAC_and_Scope_PreSelectionMux_2_Out[19..0], bestimmt durch "FSP064_USIxHS_Multiplexer[87..84]"
0x5	Controller1_P_Part_Ouput[19..0]
0x6	Controller1_I_Part_Ouput[19..0]
0x7	Controller1_PI_Part_Ouput[19..0]
0x8	Controller2_P_Part_Ouput[19..0]
0x9	Controller2_I_Part_Ouput[19..0]
0xA	Controller2_PI_Part_Ouput[19..0]
0xB	Adder_1_SumOut[19..0]
0xC	ERLOutValue[19..0], Rückspeisungsbegrenzung
0xD	PayloadRx_3_data[23..4]
0xE	intFuncGenOutput[19..0]
0xF	n.u., immer 0

[7..4] Quellenwahl für Multiplexer Ausgangssignal [DAC_2_and_ScopeChannel_2_Source] → X6

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Controller1_ActValueMuxOut[19..0], Istwert des Regler 1, bestimmt durch „FSP070_Controller_1_2_InputSourceSelectionMultiplexer[17][7..4]“
0x2	SetValue_B_MultiplexerOutput[19..0], bestimmt durch „FSP013_PeripheralConfig[0]“
0x3	ActualValue_A[19..0], analog „FSP020_ActualValue_A“
0x4	ActualValue_B[19..0], analog „FSP021_ActualValue_B“
0x5	SlopeLimiterOutput_C2[19..0]
0x6	Controller1_MultipliedDeviation[19..0], Ausgang des MFU_DifferenceCalculator, Regler 1, multiplizierte Reglerdifferenz
0x7	Controller2_MultipliedDeviation[19..0], Ausgang des MFU_DifferenceCalculator, Regler 2, multiplizierte Reglerdifferenz
0x8	DAC_and_Scope_PreSelectionMux_1_Out[19..0], bestimmt durch "FSP064_USIxHS_Multiplexer[83..80]"
0x9	DAC_and_Scope_PreSelectionMux_2_Out[19..0], bestimmt durch "FSP064_USIxHS_Multiplexer[87..84]"

0xA	Controller1_P_Part_Ouput[19..0]
0xB	Controller1_I_Part_Ouput[19..0]
0xC	Controller1_PI_Part_Ouput[19..0]
0xD	ERLOutValue[19..0], Rückspeisungsbegrenzung
0xE	PayloadRx_2_data[23..4]
0xF	intFuncGenOutput[19..0]

[3..0]

Quellenwahl für Multiplexer Ausgangssignal [DAC_1_and_ScopeChannel_1_Source] → X5

[3..0]	Ausgang des Multiplexers
0x0	0
0x1	Controller1_SetValueMuxOut[19..0], Sollwert Regler 1, bestimmt durch „FSP070_Controller_1_2_InputSourceSelectionMultiplexer[16][3..0]“
0x2	Controller2_SetValueMuxOut[19..0], Sollwert Regler 2, bestimmt durch „FSP070_Controller_1_2_InputSourceSelectionMultiplexer[18][11..8], „
0x3	SlopeLimiterOutput_C1[19..0]
0x4	Controller1_MultipliedDeviation[19..0], Ausgang des MFU_DifferenceCalculator, Regler 1, multiplizierte Regeldifferenz
0x5	Controller2_MultipliedDeviation[19..0], Ausgang des MFU_DifferenceCalculator, Regler 2, multiplizierte Regeldifferenz
0x6	intFuncGenOutput[19..0]
0x7	DAC_and_Scope_PreSelectionMux_1_Out[19..0], bestimmt durch “FSP064_USIxHS_Multiplexer[83..80]”
0x8	DAC_and_Scope_PreSelectionMux_2_Out[19..0], bestimmt durch “FSP064_USIxHS_Multiplexer[87..84]”
0x9	Controller1_I_Part_Ouput[19..0]
0xA	Controller1_PI_Part_Ouput[19..0]
0xB	ERLOutValue[19..0], Rückspeisungsbegrenzung
0xC	PayloadRx_1_data[23..4]
0xD	
...	n.u., immer 0
0xF	

Name	FSP111_SerialCom_2_ErrorGenCfg
Adresse	0x6F_H/111_D/0x3646_{ASCII}
Tiefe	5 Byte / 40 Bit
I/O	lesen/schreiben
Reset	0x000000000 _H

Konfiguriert die Fehlerinjektion, die Bitrate und setzt die Fehler für den LWL-Kanal **2** zurück.

Das Äquivalent für SerialCom_1 ist „FSP108_SerialCom_1_ErrorGenCfg“.

Das Äquivalent für SerialCom_3 ist „FSP113_SerialCom_3_ErrorGenCfg“.

Das Äquivalent für SerialCom_4 ist „FSP123_SerialCom_4_ErrorGenCfg“.

Der Status für SerialCom_2 ist im „FSP112_SerialCom_2_Status“.

Den Multiplexer für die zu sendenden Daten bedient „FSP096_SerialComDt2SendSel“.

[39..33] **Selected received payload *)**

[3..0]	Ausgang des Multiplexers
0x0	Disabled
0x1	PayloadRx_1_data[23..0]
0x2	PayloadRx_2_data[23..0]
0x3	PayloadRx_3_data[23..0]
0x4	PayloadRx_4_data[23..0]
0x5	n.u.
..	
0xF	

***) Diese Signale stehen erst ab MFU LWL FW 7.1. zur Verfügung. Vorher n.u.**

[35..33] n.u.

[32] Reset errors

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

[23..21] n.u.

[20] Error injection enable(active high)

[19..17] n.u.

[16] Change symbol flag

[15..13] n.u.

[12..8] Symbol mask

[7..0] Symbol number to corrupt

Name	FSP112_SerialCom_2_Status
Adresse	0x70_H/112_D/0x3730_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Sammelt die Alarme für den LWL-Kanal 2.

Das Äquivalent für SerialCom_1 ist „FSP109_SerialCom_1_Status“.

Das Äquivalent für SerialCom_3 ist „FSP122_SerialCom_3_Status“.

Das Äquivalent für SerialCom_4 ist „FSP124_SerialCom_4_Status“.

Das „Debugging“ der SerialCom_2 erfolgt im „FSP111_SerialCom_2_ErrorGenCfg“.

Den Multiplexer für die zu sendenden Daten bedient „FSP096_SerialComDt2SendSel“.

[23..17] n.u.

[16] PayloadAlarm_2

[15] LostConnection_2

[14] IllSymbComFailed_2

[13] PckgDis_2

[12] ParityComFailed_2

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

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

Name	FSP113_SerialCom_3_ErrorGenCfg
Adresse	0x71_H/113_D/0x3731_{ASCII}
Tiefe	5 Byte / 40 Bit
I/O	lesen/schreiben
Reset	0x000000000 _H

Konfiguriert die Fehlerinjektion, die Bitrate und setzt die Fehler für den LWL-Kanal **3** zurück.

Das Äquivalent für SerialCom_1 ist „FSP108_SerialCom_1_ErrorGenCfg“.

Das Äquivalent für SerialCom_2 ist „FSP111_SerialCom_2_ErrorGenCfg“.

Das Äquivalent für SerialCom_4 ist „FSP123_SerialCom_4_ErrorGenCfg“.

Der Status für SerialCom_3 ist im „FSP122_SerialCom_3_Status“.

Den Multiplexer für die zu sendenden Daten bedient „FSP096_SerialComDt2SendSel“.

[39..33] **Selected received payload ^{*)}**

[3..0]	Ausgang des Multiplexers
0x0	Disabled
0x1	PayloadRx_1_data[23..0]
0x2	PayloadRx_2_data[23..0]
0x3	PayloadRx_3_data[23..0]
0x4	PayloadRx_4_data[23..0]
0x5	n.u.
..	
0xF	

^{*)} Diese Signale stehen erst ab MFU LWL FW 7.1. zur Verfügung. Vorher n.u.

[35..33] n.u.

[32] Reset errors

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

[23..21] n.u.

[20] Error injection enable(active high)

[19..17] n.u.

[16] Change symbol flag

[15..13] n.u.

[12..8] Symbol mask

[7..0] Symbol number to corrupt

Name	FSP114_intScopeTFTSettings
Adresse	0x72_H/114_D/0x3732_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x01_210_0 _H

Repräsentiert die Einstellungen für die Anzeige des internen Scopes auf dem TFT Standardbildschirm.

[23..17] n.u.

[16] intScopeTFT_ShowClassicView
 Wenn ‚1‘ wird anstelle des internen Scopes auf dem TFT in jedem Fall die klassische grafische Ansicht des Stromistwertes angezeigt.
 Wenn ‚0‘ wird im ‚Remote‘-Betrieb und wenn KEIN PC mit der MFU verbunden ist im Triggerfall das interne Scope auf dem TFT dargestellt. In allen anderen Fällen die klassische grafische Ansicht des Stromistwertes.

[15..4] intScopeTFT_Channel_Selector

[15] n.u.
 [14..12] TFT Kanal **3** (grün)
 [11] n.u.
 [10..8] TFT Kanal **2** (rot)
 [7] n.u.
 [6..4] TFT Kanal **1** (blau)
 mit

Bitfolge	Aufzeichnung
000	Scope Eingang Kanal 1
001	Scope Eingang Kanal 2
010	Scope Eingang Kanal 3
011	Scope Eingang Kanal 4

[3] n.u.

[2..0] intScopeTFT_Y_Sculling

Skaliert die Auflösung der aufgezeichneten Kanäle für die TFT Darstellung. Jeder Scopeeingangskanal hat eine Auflösung von 12 Bit, die intern für die Darstellung auf dem TFT auf 7 Bit reduziert werden. Von den zwölf Bit werden bei bipolaren Geräten alle 12 Bit benutzt, das MSB bildet dabei das Vorzeichen. Bei unipolaren Geräten werden hingegen nur 11 Bit benutzt, das Vorzeichen bleibt unbenutzt. Bei bipolaren Geräten hat dies zur Folge, dass 6 Bit die eigentliche Spannung und 1 Bit das Vorzeichen repräsentieren. Bei unipolaren Geräten wird die Spannung hingegen auf allen 7 Bit abgebildet. Die Skalierung verschiebt nun diese Bitfolge nach rechts. Dies erhöht die Auflösung im kleinen Spannungsbereich, da die oberen Bit ignoriert werden. Bei bipolaren Geräten kann dadurch die Anzeige um den Faktor 25 = 32-fach und bei unipolaren Geräten 24 = 16-fach vergrößert werden.

Bitfolge	Aufzeichnung
000	Bipolar: TFT Speicherbits[Eingangsbits 11(VZ), 10..5] Unipolar: TFT Speicherbits[Eingangsbits 10..4]
001	Bipolar: TFT Speicherbits[Eingangsbits 11(VZ), 9..4] Unipolar: TFT Speicherbits[Eingangsbits 9..3]
010	Bipolar: TFT Speicherbits[Eingangsbits 11(VZ), 8..3] Unipolar: TFT Speicherbits[Eingangsbits 8..2]
011	Bipolar: TFT Speicherbits[Eingangsbits 11(VZ), 7..2] Unipolar: TFT Speicherbits[Eingangsbits 7..1]
100	Bipolar: TFT Speicherbits[Eingangsbits 11(VZ), 6..1] Unipolar: TFT Speicherbits[Eingangsbits 6..0]
101..111	Bipolar: TFT Speicherbits[Eingangsbits 11(VZ), 5..0] Unipolar: TFT Speicherbits[Eingangsbits 6..0]

Name	FSP116_intScopeSettings
Adresse	0x74_H/116_D/0x3734_{ASCII}
Tiefe	9 Byte / 72 Bit
I/O	lesen / schreiben
Reset	0x0000_0_0_0000_0000_0_0_0 _H

Repräsentiert die interne Scope Parametrierung

- [71..60] TriggerLevel (12 Bit) Repräsentiert den Triggerlevel des internen Oszilloskops
- [59..56] n.u. (4 Bit), TriggerLevel unterstes Nibble zur Rundung
- [55..52] External RAM extra configuration(4 bits)

[3..0]	
[0]	Debug mode enable. Active high. It selects as data to store in the RAM, the output of a 32 bit counter and as external trigger signal the bit nr [1] of this nibble.
[1]	Debug External trigger signal. It enables also the debug 32 bits counter described above.Active high
[2]	Trigger signal selector: if 0 => internal trigger signal selected; else external trigger signal selected.
[3]	Incoming data sampler signal: if 0=> the incoming data are sampled using the internal Scope timebase pulse signal; else the incoming data are sampled using the HighSpeedPort_2_NewDataStrobe pulse signal.

- [51..48] TimeScale (4 Bit), gibt die Skalierung der Zeitbasis an

[3..0]	Skalierung
0x0	ns, 100ns kleinster Wert bei TimeBase = 0, bzw. 1, Schrittweite 100ns
0x1	µs, 1µs kleinster Wert bei TimeBase = 0, bzw. 1, Schrittweite 1µs
0x2	ms, 1ms kleinster Wert bei TimeBase = 0, bzw. 1, Schrittweite 1ms
0x3	s, 1s kleinster Wert bei TimeBase = 0, bzw. 1, Schrittweite 1s
0x4.. 0xF	n.u., Standardskalierung 100ns wird gewählt

- [47..32] TimeBase (16 Bit), repräsentiert die Zeitbasis (horizontal) des internen Oszilloskop. 'TimeBase' gibt dabei an, wie viele male 'TimeScale' für ein Gesamtbild benötigt wird. Bsp.: abhängig von 'TimeScale' gilt für die zeitliche Auflösung bei 500 Bildpunkten: $(TimeBase \times TimeScale \times 500) / 10 = Zeit / DIV$
TimeBase = 2, TimeScale = 1(us) => 2us => alle 2us wird ein Bildpunkt erfasst. Bei 500 Werten bedeutet dies 2us * 500 = 1000us = 1ms für das gesamte Bild.

- [31..25] n.u. (7 Bit)
- [24..16] TriggerOffset (9 Bit), gibt einen Triggeroffset an
- [15..12] n.u. (4 Bit)
- [11..8] TriggerChannel (4 Bit)

[3..0]	Trigger Kanal
0x0	Kanal 1
0x1	Kanal 2
0x2	Kanal 3
0x3	Kanal 4
0x4	n.u.
0x5	n.u.
0x6	n.u.

0x7	n.u.
0x8	Externe Triggerquelle über LEMO Buchse X4 , LEMO 1 IN
0x9	Interne Triggerquelle über den internen Funktionsgenerator
...	
0xF	

[7..4] Trigger Mode (4 Bit)

[3..0]	Trigger Modus
0x0	Triggerlevel > Wert, Messwertannäherung von unten durch Triggerlevel
0x1	Triggerlevel = Wert
0x2	Triggerlevel < Wert, Messwertannäherung von oben durch Triggerlevel
0x3	positive Triggerflanke über digitalen Eingang LEMO 1 oder den int. Funktionsgenerator
0x4	negative Triggerflanke über digitalen Eingang LEMO 1 oder den int. Funktionsgenerator
0x5	n.u.
...	
0xF	

[3..0] TriggerConfig (4 Bit)

[3..0]	Beschreibung
0x0	Trigger Konfigurieren, keine Datenaufzeichnung
0x2	auf Triggerereignis warten (Trigger scharf)
0x3	n.u.
...	
0xF	

Name	FSP117_intScopeTriggerReadOut
Adresse	0x75_H/117_D/0x3735_{ASCII}
Tiefe	4 Byte / 32 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Repräsentiert Informationen über den Triggerprozess des internen Oszilloskops. Die Werte in diesem FSP werden bei einem Triggerereignis aktualisiert.

[31..28] n.u. (immer 0)

[27..24] TriggerStatus[3..0]

[3..0]	Beschreibung
0x0	TriggerIdle (FSP116_intScopeSettings[3..0] FunctionMode auf 0x0, keine Datenaufzeichnung, Trigger kann konfiguriert werden)
0x1	PrefillingRAM (FSP116_intScopeSettings[3..0] FunctionMode wurde auf 0x2 gesetzt, Datenaufzeichnung noch <u>nicht</u> aktiv, warten bis RAM vorgefüllt ist)
0x2	WaitForTriggerEvent (RAM ist vorgefüllt, FSP116_intScopeSettings[3..0] FunctionMode ist auf 0x2, Datenaufzeichnung aktiv, warten auf Triggerereignis)
0x3	-
0x4	TriggerEventOccured (das in FSP116_intScopeSettings[7..4] TriggerMode gewählte Ereignis hat stattgefunden)
0x5	-
...	
0x7	
0x8	WaitForReadOutData (alle Daten rund um das Triggerereignis sind aufgezeichnet und zur Abholung bereit)
0x9	-
...	
0xF	

[23..21] n.u. (immer 0)

[20..12] TriggerOffsetEcho[8..0]
entspricht nach dem Triggerereignis dem Wert von FSP116_intScopeSettings[25..16]

[11..9] n.u. (immer 0)

[8..0] TriggerAddress[8..0]
entspricht nach dem Triggerereignis der Trigger-Adresse

Name	FSP118_intScopeDataReadOutAddress
Adresse	0x76_H/118_D/0x3736_{ASCII}
Tiefe	2 Byte / 16 Bit
I/O	lesen / schreiben
Reset	0x0000 _H

Repräsentiert die 9 Bit Adresse des internen Scopespeichers dessen Daten gelesen werden sollen

[15..9] n.u. (7 Bit)

[8..0] ScopeReadOutRAMAddress (9 Bit), repräsentiert die RAM Speicheradresse des internen Oszilloskop die angesprochen werden soll. Das interne Oszilloskop besitzt 4 Kanäle die in 4 RAM Speicher abgelegt werden können. Dies geschieht zeitgleich, d.h. die jeweils angelegte Adresse adressiert alle 4 RAM Speicher zeitgleich.

Name	FSP119_intScopeDataReadOut
Adresse	0x77_H/119_D/0x3737_{ASCII}
Tiefe	10 Byte / 80 Bit (ab MFU FW 7.5.0)
I/O	lesen
Reset	0x(siehe Beschreibung)

Repräsentiert die 9 Bit Adresse des internen Scopespeichers, und die darin befindlichen Daten (ScopeRamData_CHx) der Kanäle 1 bis 4.

[79..76] n.u. (4 Bit, immer 0)

[75..64] ScopeRamData_CH4[11..0] (12 Bit)

[63..60] n.u. (4 Bit, immer 0)

[59..48] ScopeRamData_CH3[11..0] (12 Bit)

[47..44] n.u. (4 Bit, immer 0)

[43..32] ScopeRamData_CH2[11..0] (12 Bit)

[31..28] n.u. (4 Bit, immer 0)

[27..16] ScopeRamData_CH1[11..0] (12 Bit)

[15..9] n.u. (7 Bit, immer 0)

[8..0] ScopeReadOutRAMAddress[8..0], (9 Bit) entspricht dem Eintrag in „FSP118_intScopeDataReadOutAddress“ und dient der Verifizierung ob die Daten auch zu dieser Adresse gehören.

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_000000_000000 _H

Repräsentiert die Parameter für den internen Funktionsgenerator

[127..124] ModeSelect

[3..0]	ModeSelect
0x0	Funktionsgenerator läuft frei, unabhängig ‚ExtSyncSource‘
0x1	Eingänge ‚ManualDirectionUP_nDown‘ und ‚ManualTriggerCounter‘ sind aktiv
0x2	Rampe aufwärts/abwärts startet nur mit steigender Flanke an ‚ExternalEnableSelect‘
0x3	
...	-
0xF	

[123..120] ExternalEnableSelect

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

[119..96] RampRiseTime (24 Bit), Wenn X"00_0000" pulst der Funktionsgenerator, andernfalls wird gerammt. RampTime repräsentiert dabei die Pausenzeit zwischen zwei Rampenstützpunkten während der Anstiegs-, bzw. Abstiegszeit der Rampe. Gibt die Zeit der Pausenintervalle zwischen den einzelnen Inkrementen zwischen dem ‚LowermostValueRampPulse‘ und dem ‚TopValueRampPulse‘ an. Das Pausenintervall entspricht „(RampRise * 10ns) + 10 ns“. Beispiel: LowermostValueRampPulse = -32768_D
 TopValueRampPulse = 32767_D
 RampTime = 1_D
 Es liegen 65535 Stützpunkte zwischen dem Minimal und Maximalwert.
 Es wird alle „(RampRise * 10ns)“ ein neuer Stützpunkt ausgegeben, d.h. alle 10ns. Eine Rampe dauert also 65535 * (1*10ns)

- [95..72]
- [95..89] n.u. (7 Bit)
- [88..72] LowermostValueDuration (17 Bit), repräsentiert die Zeitdauer die der untere (minimale) Pulswert/Scheitelwert der Rampe (Faltbottom) des internen Funktionsgenerators anstehen soll. Ist Bit [88] gesetzt gilt der Wert von Bit[87..72] in uSekunde. Ist Bit [88] nicht gesetzt gilt der Wert von Bit[87..72] in mSekunden.
-
- [71..48]
- [71..65] n.u. (7 Bit)
- [64..48] TopValueDuration (17 Bit), repräsentiert die Zeitdauer die der obere (maximale) Pulswert/Scheitelwert der Rampe (Flatop) des internen Funktionsgenerators anstehen soll. Ist Bit [64] gesetzt gilt der Wert von Bit [63..48] in uSekunde. Ist Bit [64] nicht gesetzt gilt der Wert von Bit [63..48] in mSekunden.
-
- [47..24]
- [47..28] LowermostValueRampPulse (20 Bit), repräsentiert den unteren (minimalen) Pulswert/Scheitelwert der Rampe des internen Funktionsgenerators. Vorzeichenbehafteter Wert im Bereich zwischen -11..+11 Volt.
- [27..24] n.u. (4 Bit)
-
- [23..0]
- [23..4] TopValueRampPulse (20 Bit), repräsentiert den oberen (maximalen) Pulswert/Scheitelwert der Rampe des internen Funktionsgenerators. Vorzeichenbehafteter Wert im Bereich zwischen -11..+11 Volt.
- [3..0] n.u. (4 Bit)

Name	FSP121_ControllerStatusBits LE/SE
Adresse	0x79_H/121_D/0x3739_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x01 _H

Hierbei handelt es sich um ein binäres Datenwort, bestehend aus 23 Bits, die verschiedene Steuerzustände innerhalb der MFU repräsentieren.

[23..20]	n.u., immer ,0'
[19]	Controller_1_I_Part_Disable
[18]	Controller_2_I_Part_Disable
[17]	Controller_1_P2_Part_Active
[16]	Controller_2_P2_Part_Active
[15]	Degaussing_Active
[14]	Controller_1_PWM_FDrive_Comparator
[13]	Controller_2_PWM_FDrive_Comparator
[12]	PWM_FDriveComparator_MuxOut
[11]	Controller_1_IGBT_V5_Active
[10]	Controller_2_IGBT_V5_Active
[9]	IGBT_V5_Active
[8]	n.u.
[7]	ACU_Controller_Enable
[6]	SlopeLimiterEnable C1
[5]	n.u.
[4]	SlopeLimiterEnable C2
[3..0]	immer ,0'

Name	FSP122_SerialCom_3_Status
Adresse	0x7A_H/122_D/0x3741_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Sammelt die Alarme für den LWL-Kanal **3**.

Das Äquivalent für SerialCom_1 ist „FSP109_SerialCom_1_Status“.

Das Äquivalent für SerialCom_2 ist „FSP112_SerialCom_2_Status“.

Das Äquivalent für SerialCom_4 ist „FSP124_SerialCom_4_Status“.

Das „Debugging“ der SerialCom_3 erfolgt im „FSP113_SerialCom_3_ErrorGenCfg“.

Den Multiplexer für die zu sendenden Daten bedient „FSP096_SerialComDt2SendSel“.

[23..17] n.u.

[16] PayloadAlarm_3

[15] LostConnection_3

[14] IllSymbComFailed_3

[13] PckgDis_3

[12] ParityComFailed_3

[11..8] ParityErrorCnt_3[3..0]

[7..0] CntError_3[7..0]

Name	FSP123_SerialCom_4_ErrorGenCfg
Adresse	0x7B_H/123_D/0x3742_{ASCII}
Tiefe	5 Byte / 40 Bit
I/O	lesen/schreiben
Reset	0x000000000 _H

Konfiguriert die Fehlerinjektion, die Bitrate und setzt die Fehler für den LWL-Kanal **4** zurück.

Das Äquivalent für SerialCom_1 ist „FSP108_SerialCom_1_ErrorGenCfg“.

Das Äquivalent für SerialCom_2 ist „FSP111_SerialCom_2_ErrorGenCfg“.

Das Äquivalent für SerialCom_3 ist „FSP113_SerialCom_3_ErrorGenCfg“.

Der Status für SerialCom_4 ist im „FSP124_SerialCom_4_Status“.

Den Multiplexer für die zu sendenden Daten bedient „FSP096_SerialComDt2SendSel“.

[39..33] **Selected received payload *)**

[3..0]	Ausgang des Multiplexers
0x0	Disabled
0x1	PayloadRx_1_data[23..0]
0x2	PayloadRx_2_data[23..0]
0x3	PayloadRx_3_data[23..0]
0x4	PayloadRx_4_data[23..0]
0x5	n.u.
..	
0xF	

***) Diese Signale stehen erst ab MFU LWL FW 7.1. zur Verfügung. Vorher n.u.**

[35..33] n.u.

[32] Reset errors

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

[23..21] n.u.

[20] Error injection enable(active high)

[19..17] n.u.

[16] Change symbol flag

[15..13] n.u.

[12..8] Symbol mask

[7..0] Symbol number to corrupt

Name	FSP124_SerialCom_4_Status
Adresse	0x7C_H/124_D/0x3743_{ASCII}
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) _H

Sammelt die Alarme für den LWL-Kanal **4**.

Das Äquivalent für SerialCom_1 ist „FSP109_SerialCom_1_Status“.

Das Äquivalent für SerialCom_2 ist „FSP112_SerialCom_2_Status“.

Das Äquivalent für SerialCom_3 ist „FSP122_SerialCom_3_Status“.

Das „Debugging“ der SerialCom_4 erfolgt im „FSP123_SerialCom_4_ErrorGenCfg“.

Den Multiplexer für die zu sendenden Daten bedient „FSP096_SerialComDt2SendSel“.

[23..17] n.u.

[16] PayloadAlarm_4

[15] LostConnection_4

[14] IllSymbComFailed_4

[13] PckgDis_4

[12] ParityComFailed_4

[11..8] ParityErrorCnt_4[3..0]

[7..0] CntError_4[7..0]