



## Allgemeingültige FSPs

Version vom: Mittwoch, 4. Mai 2022, 12:02:00

## Inhaltsverzeichnis

1. Änderungsliste .....	1
2. Allgemeingültige FSPs .....	2
FSP001_ModuleStatus .....	3
0x01 <sub>H</sub> /1 <sub>D</sub> /0x3031 <sub>ASCII</sub>	
FSP002_ModuleWarnings .....	4
0x02 <sub>H</sub> /2 <sub>D</sub> /0x3032 <sub>ASCII</sub>	
FSP003_ModuleErrors.....	5
0x03 <sub>H</sub> /3 <sub>D</sub> /0x3033 <sub>ASCII</sub>	
FSP004_ModuleInterlocks.....	6
0x04 <sub>H</sub> /4 <sub>D</sub> /0x3034 <sub>ASCII</sub>	
FSP005_InterlocksEnable .....	7
0x05 <sub>H</sub> /5 <sub>D</sub> /0x3035 <sub>ASCII</sub>	
FSP006_InterlocksArrivalSequence .....	8
0x06 <sub>H</sub> /6 <sub>D</sub> /0x3036 <sub>AS</sub>	
FSP007_Reserved.....	9
0x07 <sub>H</sub> /7 <sub>D</sub> /0x3037 <sub>ASCII</sub>	
FSP008_Reserved.....	10
0x08 <sub>H</sub> /8 <sub>D</sub> /0x3038 <sub>ASCII</sub>	
FSP009_ModuleSerialNumber .....	11
0x09 <sub>H</sub> /9 <sub>D</sub> /0x3039 <sub>ASCII</sub>	
FSP010_ModuleCommands .....	12
0x0A <sub>H</sub> /10 <sub>D</sub> /0x3041 <sub>ASCII</sub>	
FSP011_ModuleInterlocksMask_n .....	13
0x0B <sub>H</sub> /11 <sub>D</sub> /0x3042 <sub>ASCII</sub>	
FSP012_ModuleUSIConfig.....	14
0x0C <sub>H</sub> /12 <sub>D</sub> /0x3043 <sub>ASCII</sub>	
FSP013_PeripheralConfig.....	15
0x0D <sub>H</sub> /13 <sub>D</sub> /0x3044 <sub>ASCII</sub>	
FSP014_CurrentScale .....	16
0x0E <sub>H</sub> /14 <sub>D</sub> /0x3045 <sub>ASCII</sub>	
FSP015_VoltageScale.....	17
0x0F <sub>H</sub> /15 <sub>D</sub> /0x3046 <sub>ASCII</sub>	
FSP016_BFieldScale.....	18
0x10 <sub>H</sub> /16 <sub>D</sub> /0x3130 <sub>ASCII</sub>	
FSP017_DataStorageStatus.....	19
0x11 <sub>H</sub> /17 <sub>D</sub> /0x3131 <sub>ASCII</sub>	
FSP018_DataStorageCfg.....	20
0x12 <sub>H</sub> /18 <sub>D</sub> /0x3132 <sub>ASCII</sub>	
FSP019_Reserved.....	21
0x13 <sub>H</sub> /19 <sub>D</sub> /0x3133 <sub>ASCII</sub>	
FSP020_ActualValue_A.....	22
0x14 <sub>H</sub> /20 <sub>D</sub> /0x3134 <sub>ASCII</sub>	
FSP021_ActualValue_B .....	23
0x15 <sub>H</sub> /21 <sub>D</sub> /0x3231 <sub>ASCII</sub>	
FSP022_ActualValue_C .....	24
0x16 <sub>H</sub> /22 <sub>D</sub> /0x3232 <sub>ASCII</sub>	
FSP023_ActualValue_D .....	25
0x17 <sub>H</sub> /23 <sub>D</sub> /0x3137 <sub>ASCII</sub>	
FSP024_Reserved.....	26
0x18 <sub>H</sub> /24 <sub>D</sub> /0x3138 <sub>ASCII</sub>	
FSP025_Reserved.....	27
0x19 <sub>H</sub> /25 <sub>D</sub> /0x3139 <sub>ASCII</sub>	
FSP026_Reserved.....	28
0x1A <sub>H</sub> /26 <sub>D</sub> /0x3141 <sub>ASCII</sub>	
FSP027_Reserved.....	29
0x1B <sub>H</sub> /27 <sub>D</sub> /0x3142 <sub>ASCII</sub>	
FSP028_Reserved.....	30
0x1C <sub>H</sub> /28 <sub>D</sub> /0x3143 <sub>ASCII</sub>	
FSP029_ActualValuePhysicalQuantities .....	31
0x1D <sub>H</sub> /39 <sub>D</sub> /0x3144 <sub>ASCII</sub>	
FSP030_SetValue_A.....	32
0x1E <sub>H</sub> /30 <sub>D</sub> /0x3145 <sub>ASCII</sub>	
FSP031_SetValue_B .....	33
0x1F <sub>H</sub> /31 <sub>D</sub> /0x3246 <sub>ASCII</sub>	
FSP032_SetValue_C.....	34

0x20 <sub>H</sub> /32 <sub>D</sub> /0x3230 <sub>ASCII</sub>	
FSP033_SetValue_D .....	35
0x21 <sub>H</sub> /33 <sub>D</sub> /0x3231 <sub>ASCII</sub>	
FSP034_Reserved.....	36
0x22 <sub>H</sub> /34 <sub>D</sub> /0x3232 <sub>ASCII</sub>	
FSP035_Reserved.....	37
0x23 <sub>H</sub> /35 <sub>D</sub> /0x3233 <sub>ASCII</sub>	
FSP036_Reserved.....	38
0x24 <sub>H</sub> /36 <sub>D</sub> /0x3234 <sub>ASCII</sub>	
FSP037_Reserved.....	39
0x25 <sub>H</sub> /37 <sub>D</sub> /0x3235 <sub>ASCII</sub>	
FSP038_Reserved.....	40
0x26 <sub>H</sub> /38 <sub>D</sub> /0x3236 <sub>ASCII</sub>	
FSP039_SetValuePhysicalQuantities .....	41
0x27 <sub>H</sub> /39 <sub>D</sub> /0x3237 <sub>ASCII</sub>	
FSP040_RemoteUpdateStatus.....	42
0x28 <sub>H</sub> /40 <sub>D</sub> /0x3238 <sub>ASCII</sub>	
FSP041_RemoteUpdateCommands.....	43
0x29 <sub>H</sub> /41 <sub>D</sub> /0x3239 <sub>ASCII</sub>	
FSP042_RemoteUpdateData.....	44
0x2A <sub>H</sub> /42 <sub>D</sub> /0x3241 <sub>ASCII</sub>	
FSP043_FSP_Image_Generator .....	46
0x2B <sub>H</sub> /43 <sub>D</sub> /0x3242 <sub>ASCII</sub>	
FSP044_FSP_Image_Status.....	47
0x2C <sub>H</sub> /44 <sub>D</sub> /0x3243 <sub>ASCII</sub>	
FSP045_AlteraRemoteUpdateCmd.....	48
0x2D <sub>H</sub> /45 <sub>D</sub> /0x3244 <sub>ASCII</sub>	
FSP046_AlteraRemoteUpdateStatus.....	49
0x2E <sub>H</sub> /46 <sub>D</sub> /0x3245 <sub>ASCII</sub>	
FSP047_Reserved.....	50
0x2F <sub>H</sub> /47 <sub>D</sub> /0x3246 <sub>ASCII</sub>	
FSP048_Reserved.....	51
0x30 <sub>H</sub> /48 <sub>D</sub> /0x3330 <sub>ASCII</sub>	
FSP049_Reserved.....	52
0x31 <sub>H</sub> /49 <sub>D</sub> /0x3331 <sub>ASCII</sub>	
FSP050_ModuleSupplyValues .....	53
0x32 <sub>H</sub> /50 <sub>D</sub> /0x3332 <sub>ASCII</sub>	
FSP051_ModulePotIValues .....	54
0x33 <sub>H</sub> /51 <sub>D</sub> /0x3333 <sub>ASCII</sub>	
FSP052_ModuleComparatorValues.....	55
0x34 <sub>H</sub> /52 <sub>D</sub> /0x3334 <sub>ASCII</sub>	
FSP053_ModuleTemperatures.....	56
0x35 <sub>H</sub> /53 <sub>D</sub> /0x3335 <sub>ASCII</sub>	
FSP054_ModuleTemperaturesComparisonThresholds.....	57
0x36 <sub>H</sub> /54 <sub>D</sub> /0x3336 <sub>ASCII</sub>	
FSP055_WaterFlowMeasurement .....	58
0x37 <sub>H</sub> /55 <sub>D</sub> /0x3337 <sub>ASCII</sub>	
FSP056_Reserved.....	59
0x38 <sub>H</sub> /56 <sub>D</sub> /0x3338 <sub>ASCII</sub>	
FSP057_Reserved.....	60
0x39 <sub>H</sub> /57 <sub>D</sub> /0x3339 <sub>ASCII</sub>	
FSP058_ParameterChecksumValue .....	61
0x3A <sub>H</sub> /58 <sub>D</sub> /0x3341 <sub>ASCII</sub>	
FSP059_ParameterChecksumValueCalculated .....	62
0x3B <sub>H</sub> /59 <sub>D</sub> /0x3342 <sub>ASCII</sub>	

**1. Änderungsliste**

Datum	Name	Kommentar
01.07.2021	D. Schupp	Dokument erstellt aus ACU-FSP mUSIc TFT

## **2. Allgemeingültige FSPs**

Die FSPs 0 bis 59 sind in ihrer Funktion weitestgehend vereinheitlicht und sofern notwendig in die Module zu integrieren. Ab FSP 60 sind diese modulspezifisch und deshalb auch in den einzelnen Modulkapiteln beschrieben.

Name	<b>FSP001_ModuleStatus</b>
Adresse	<b>0x01<sub>H</sub>/1<sub>D</sub>/0x3031<sub>ASCII</sub></b>
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

- [23..9] Module Status,  
Kommando Status  
z.B. Infos über die Ablaufsteuerungszustand oder Kommandos des Moduls
- [8] USIIsHighSpeed  
wenn ,1' ist USI im Highspeed Mode
- [7..6] Reserviert für zukünftige Anwendungen
- [5] NoInterlocks  
wenn ,1' stehen keine Interlocks an  
Im Modul sind keine Interlocks gespeichert und es stehen auch keine Interlocks an.
- [4] NoErrors  
wenn ,1' ist Modul fehlerfrei  
Im Modul sind keine Fehler gespeichert die den Betrieb stören.
- [3] NoWarnings  
wenn ,1' ist Modul ohne Warnungen  
Im Modul sind keine Warnmeldungen vorhanden die den Betrieb zwar nicht stören aber trotzdem überprüft werden müssten (Details im FSP für die Warnungsbits) z.B. Temperatur zu hoch.
- [2] ModuleReady  
wenn ,1' ist Modul betriebsbereit  
Das Modul ist voll betriebsbereit
- [1] ChecksumOK  
wenn ,1' Parameter Checksumme OK  
Die Prüfsumme für die Modulparameter ist bestätigt.
- [0] ParametersLoaded  
wenn ,1' sind die Parameter geladen  
Das Modul hat seine Konfigurationsparameter geladen.

---

Name	<b>FSP002_ModuleWarnings</b>
Adresse	<b>0x02<sub>H</sub>/2<sub>D</sub>/0x3032<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

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

Liegt eine Warnung vor ist das korrespondierende Bit ,0' andernfalls ,1'. Außerdem ist Bit [3] des FSP001\_ModuleStatus = ,0'.

---

Name	<b>FSP003_ModuleErrors</b>
Adresse	<b>0x03<sub>H</sub>/3<sub>D</sub>/0x3033<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

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

Liegt ein Fehler vor ist das korrespondierende Bit ,0' andernfalls ,1'. Außerdem ist Bit [4] des FSP001\_ModuleStatus = ,0'.



Name	<b>FSP004_ModuleInterlocks</b>
Adresse	<b>0x04<sub>H</sub>/4<sub>D</sub>/0x3034<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

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

#### **WICHTIG**

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

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

Liegt ein Interlock vor ist das korrespondierende Bit ,0' andernfalls ,1'. Außerdem ist Bit [5] des FSP001\_ModuleStatus = ,0'.

Nicht genutzte Interlockbits müssen ,1' sein!

Definitionen für:

Name	<b>FSP005_InterlocksEnable</b>
Adresse	<b>0x05<sub>H</sub>/5<sub>D</sub>/0x3035<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

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

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

Interlockdefinitionen für:

Name	<b>FSP006_InterlocksArrivalSequence</b>
Adresse	<b>0x06<sub>H</sub>/6<sub>D</sub>/0x3036<sub>ASCII</sub></b>
Tiefe	Modulabhängig
I/O	lesen
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

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

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

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

Das FSP kann die Tiefe aller im Modul erfassbaren Interlocks besitzen, oder aber nur eine bestimmte Anzahl sequenziell auftretender Interlocks erfassen.

I.d.R. wird die Erfassung der 5 zuerst auftretenden Interlocks ausreichend sein.

---

Name	<b>FSP007_Reserved</b>
Adresse	<b>0x07<sub>H</sub>/7<sub>D</sub>/0x3037<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP008_Reserved</b>
Adresse	<b>0x08<sub>H</sub>/8<sub>D</sub>/0x3038<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP009_ModuleSerialNumber</b>
Adresse	<b>0x09<sub>H</sub>/9<sub>D</sub>/0x3039<sub>ASCII</sub></b>
Tiefe	6 Byte / 48 Bit
I/O	lesen
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

Der FSP enthält die Modul 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.

Name	<b>FSP010_ModuleCommands</b>
Adresse	<b>0x0A<sub>H</sub>/10<sub>D</sub>/0x3041<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

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

[7..4] n.u.

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

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

Name	<b>FSP011_ModuleInterlocksMask_n</b>
Adresse	<b>0x0B<sub>H</sub>/11<sub>D</sub>/0x3042<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

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

Wie beim FSP004\_ModuleInterlocks liegt die Grenze normalerweise in der Mitte des FSP (Ausnahme siehe weiter unten im Text), wobei immer ganze Bytes für die Gruppen verwendet werden müssen. d.h. für 3 Interlocks müssen trotzdem 2 Bytes verwendet werden. 1 Byte für den aktuellen Status und 1 Byte für die gespeicherte Meldung.

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

#### Sonderfunktion: **Zugelassene Interlocks nach der Reglerfreigabe**

Dieses FSP kann eine weitere Bitmaske enthalten, die sich noch oberhalb der Masken für aktuell anstehende und gespeicherte Interlocks befindet

Das FSP enthält dann drei Bitmasken, die zum einen nicht verwendete Interlocks vollständig ausmaskiert, d.h. alle nicht zu benutzenden Interlocks sind mit ,1' zu setzen. Dies wird bereits weiter oben beschrieben. Zum anderen lassen sich Interlocks mit diesem FSP und der zusätzlichen Maske so maskieren, dass diese erst nach Freigabe des Reglers aktiviert werden. Deren Erfassung wird also während einer Reglersperre ignoriert. Hierbei ist das zugehörige Bit für verzögerte Freigabe des Interlocks (zusammen mit der Reglerfreigabe) mit ,1' zu setzen.

Die Verteilung der Masken ist dann wie folgt:

[Maske verzögerte Freigabe][Maske gespeicherte Interlocks][Maske aktuelle Interlocks]



Name	<b>FSP012_ModuleUSIConfig</b>
Adresse	<b>0x0C<sub>H</sub>/12<sub>D</sub>/0x3043<sub>ASCII</sub></b>
Tiefe	1 Byte / 8 Bit
I/O	lesen / schreiben
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

Hiermit wird die USI Schnittstelle des Moduls konfiguriert.

- [7] USIIsHighSpeed  
wenn ,1' schaltet USI in HighSpeed Modus um
- [6..3] n.u.
- [2..0] Bitrate des Moduls [111 = 115k]

---

Name	<b>FSP013_PeripheralConfig</b>
Adresse	<b>0x0D<sub>H</sub>/13<sub>D</sub>/0x3044<sub>ASCII</sub></b>
Tiefe	1 Byte / 8 Bit
I/O	lesen / schreiben
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

Auf dem Modul befindliche Peripherie kann mit diesem FSP konfiguriert werden

---

Name	<b>FSP014_CurrentScale</b>
Adresse	<b>0x0E<sub>H</sub>/14<sub>D</sub>/0x3045<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

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

---

Name	<b>FSP015_VoltageScale</b>
Adresse	<b>0x0F<sub>H</sub>/15<sub>D</sub>/0x3046<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

Repräsentiert die Spannungskalierung des ACU.

---

Name	<b>FSP016_BFieldScale</b>
Adresse	<b>0x10<sub>H</sub>/16<sub>D</sub>/0x3130<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	Reset:0x(siehe Beschreibung) <sub>H</sub>

Repräsentiert die Feldskalierung des ACU Systems.

---

Name	<b>FSP017_DataStorageStatus</b>
Adresse	<b>0x11<sub>H</sub>/17<sub>D</sub>/0x3131<sub>ASCII</sub></b>
Tiefe	2 Byte / 16 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Enthält den DataStorage Status

Name	<b>FSP018_DataStorageCfg</b>
Adresse	<b>0x12<sub>H</sub>/18<sub>D</sub>/0x3132<sub>ASCII</sub></b>
Tiefe	6 Byte / 48 Bit
I/O	Lesen /schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

Enthält die DataStorage Konfiguration

---

Name	<b>FSP019_Reserved</b>
Adresse	<b>0x13<sub>H</sub>/19<sub>D</sub>/0x3133<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert



---

Name	<b>FSP020_ActualValue_A</b>
Adresse	<b>0x14<sub>H</sub>/20<sub>D</sub>/0x3134<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP liefert einen im Modul vorhandenen Istwert\_A zurück

---

Name	<b>FSP021_ActualValue_B</b>
Adresse	<b>0x15<sub>H</sub>/21<sub>D</sub>/0x3231<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP liefert einen im Modul vorhandenen Istwert\_B zurück

---

Name	<b>FSP022_ActualValue_C</b>
Adresse	<b>0x16<sub>H</sub>/22<sub>D</sub>/0x3232<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP liefert einen im Modul vorhandenen Istwert\_C zurück

---

Name	<b>FSP023_ActualValue_D</b>
Adresse	<b>0x17<sub>H</sub>/23<sub>D</sub>/0x3137<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP liefert einen im Modul vorhandenen Istwert\_D zurück

---

Name	<b>FSP024_Reserved</b>
Adresse	<b>0x18<sub>H</sub>/24<sub>D</sub>/0x3138<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP025_Reserved</b>
Adresse	<b>0x19<sub>H</sub>/25<sub>D</sub>/0x3139<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP026_Reserved</b>
Adresse	<b>0x1A<sub>H</sub>/26<sub>D</sub>/0x3141<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP027_Reserved</b>
Adresse	<b>0x1B<sub>H</sub>/27<sub>D</sub>/0x3142<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert



---

Name	<b>FSP028_Reserved</b>
Adresse	<b>0x1C<sub>H</sub>/28<sub>D</sub>/0x3143<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

Name	<b>FSP029_ActualValuePhysicalQuantities</b>
Adresse	<b>0x1D<sub>H</sub>/39<sub>D</sub>/0x3144<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Gibt die physikalische Größe der Istwerte an.

---

Name	<b>FSP030_SetValue_A</b>
Adresse	<b>0x1E<sub>H</sub>/30<sub>D</sub>/0x3145<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP dient zur Übermittlung eines Sollwertes\_A an das Modul

Name	<b>FSP031_SetValue_B</b>
Adresse	<b>0x1F<sub>H</sub>/31<sub>D</sub>/0x3246<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP dient zur Übermittlung eines Sollwertes\_B an das Modul

---

Name	<b>FSP032_SetValue_C</b>
Adresse	<b>0x20<sub>H</sub>/32<sub>D</sub>/0x3230<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP dient zur Übermittlung eines Sollwertes\_C an das Modul

---

Name	<b>FSP033_SetValue_D</b>
Adresse	<b>0x21<sub>H</sub>/33<sub>D</sub>/0x3231<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP dient zur Übermittlung eines Sollwertes\_D an das Modul

---

Name	<b>FSP034_Reserved</b>
Adresse	<b>0x22<sub>H</sub>/34<sub>D</sub>/0x3232<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP035_Reserved</b>
Adresse	<b>0x23<sub>H</sub>/35<sub>D</sub>/0x3233<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert



---

Name	<b>FSP036_Reserved</b>
Adresse	<b>0x24<sub>H</sub>/36<sub>D</sub>/0x3234<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP037_Reserved</b>
Adresse	<b>0x25<sub>H</sub>/37<sub>D</sub>/0x3235<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP038_Reserved</b>
Adresse	<b>0x26<sub>H</sub>/38<sub>D</sub>/0x3236<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP039_SetValuePhysicalQuantities</b>
Adresse	<b>0x27<sub>H</sub>/39<sub>D</sub>/0x3237<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

Gibt die physikalische Größe der Sollwerte an.

Name	<b>FSP040_RemoteUpdateStatus</b>
Adresse	<b>0x28<sub>H</sub>/40<sub>D</sub>/0x3238<sub>ASCII</sub></b>
Tiefe	1 Byte / 8 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Wenn ein Modul Remoteupdate unterstützt, ist dieser FSP für das Rücklesen der Statusinformationen des Fernupdates zu benutzen.

[7..3] n.u.

[2] FSP042\_Busy  
wenn ,1' ist FSP42 beschäftigt (z.B. weil gerade Flashsektoren gelöscht oder programmiert werden) und es sollten KEIN Zugriffe darauf erfolgen

[1] FSP042\_ReadyToSendData,  
wenn ,1' können Daten vom Host an FSP42 abgeholt werden

[0] FSP042\_ReadyToReceiveData,  
wenn ,1' können Daten vom Host an FSP42 gesendet werden

Name	<b>FSP041_RemoteUpdateCommands</b>
Adresse	<b>0x29<sub>H</sub>/41<sub>D</sub>/0x3239<sub>ASCII</sub></b>
Tiefe	1 Byte / 8 Bit
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP überträgt die Kommandos für das Fernupdate

[7..3] n.u.

[2..0] Kommandos für den RemoteUpdateHandler

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

Name	<b>FSP042_RemoteUpdateData</b>
Adresse	<b>0x2A<sub>H</sub>/42<sub>D</sub>/0x3241<sub>ASCII</sub></b>
Tiefe	256 Byte
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

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

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

### Zugriffe einleiten

Zugriffe auf FSP042 müssen generell über FSP041 eingeleitet werden.

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

### Lesen

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

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

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

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

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

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

### Schreiben

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

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

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

Jetzt erfolgt die Einleitung des Schreibkommandos.

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

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

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

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

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

**Abbrechen/Beenden**

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



Name	<b>FSP043_FSP_Image_Generator</b>
Adresse	<b>0x2B<sub>H</sub>/43<sub>D</sub>/0x3242<sub>ASCII</sub></b>
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x00_0000 <sub>H</sub>

Über diesen FSP wird ein FSP Image erzeugt und im seriellen Flash gespeichert, bzw. die Images der FSPs aus dem seriellen Flash geladen. Der serielle Flash ist hierdurch auch löscherbar. Statusrückmeldungen gibt das „FSP044\_FSP\_Image\_Status“.

#### Hinweis

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

- [23..16] FSPImgSectorAddress, sofern FSPImgUseSectorAddress; = ,1‘ wird der hier angegebene Sektor im ext. seriellen Flash gelöscht, zur Sicherung, bzw. zum Laden von FSP Daten genutzt.
- [15..8] FSPImgFSPNumber, gibt die FSP Nummer an, die mittels ‚FSPImgMakelImage‘ ins ext. serielle Flash gesichert werden soll.
- [7..4] n.u.
- [3] FSPImgUseSectorAddress, wenn ‚1‘ wird das Image in den ‚FSPImgSectorAddress‘ Sektor geschrieben, wenn ‚0‘ wird die Startadresse aus dem ‚M25P\_Access‘ Modulgeneric ‚gFSPImgStartAddress‘ verwendet.
- [2] FSPImgEraseAll, wenn ‚1‘ wird ein Sektor im ext. seriellen Flash gelöscht. Dieser ist abhängig von ‚FSPImgUseSectorAddress‘ und ‚FSPImgSectorAddress‘ oder ‚gFSPImgStartAddress‘. Zum Starten des Löschvorgangs muss ein ‚0‘ → ‚1‘ Übergang erfolgen.
- [1] FSPImgMakelImage, wenn ‚1‘ wird der Inhalt des FSP ‚FSPImgFSPNumber‘ ins ext. serielle Flash gesichert. Zum Starten des Sicherungsvorgangs muss ein ‚0‘ → ‚1‘ Übergang erfolgen. Sollen mehrere FSP gesichert werden ist ‚FSPImgMakelImage‘ mehrfach zu aktivieren und die jeweilige FSP Nummer in ‚FSPImgFSPNumber‘ anzugeben.
- [0] FSPImgLoadImage, wenn ‚1‘ wird der Inhalt des ext. seriellen Flashs in die FSPs geladen. Sind zuvor mehrere FSP gesichert worden, werden alle Sicherungen geladen. Zum Starten des Ladevorgangs muss ein ‚0‘ → ‚1‘ Übergang erfolgen.

Name	<b>FSP044_FSP_Image_Status</b>
Adresse	<b>0x2C<sub>H</sub>/44<sub>D</sub>/0x3243<sub>ASCII</sub></b>
Tiefe	1 Byte / 8 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Liefert Statusrückmeldungen bzgl. des Sicherungsvorgangs von FSP Daten. Die Sicherung wird über „FSP043\_FSP\_Image\_Generator“ gesteuert.

[7..1] n.u., immer ,0‘

[0] FSPImgReady, ist ,1‘ wenn das ext. Flash Zugriffe zum sichern/laden von FSP Daten gestattet, ist ,0‘ wenn Daten gesichert oder gelesen werden, bzw. ein Löschvorgang durchgeführt wird. Ist diese Bit ,0‘ und mittels FSP070 werden Zugriffe auf das ext. Flash gestartet ist das Ergebnis undefiniert.

Name	<b>FSP045_AlteraRemoteUpdateCmd</b>
Adresse	<b>0x2D<sub>H</sub>/45<sub>D</sub>/0x3244<sub>ASCII</sub></b>
Tiefe	6/7 Byte / 48/56 Bit
I/O	lesen / schreiben
Reset	Reset:0x00100000_00_0_0_0_0 <sub>H</sub>

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)

Mit Einführung des CycloneV änderte sich dieses FSP inhaltlich leicht.

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

#### **Altes Format**

[47..44] n.u.

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

[19..17] n.u.

#### **Neues Format**

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

[23..17] n.u.

#### **Gemeinsam unverändert**

[16] Reset WD Disable (only for debug)

[15..13] n.u.

[12] Start Write (steigende Flanke an diesem Bit startet die FSM zum 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	<b>FSP046_AlteraRemoteUpdateStatus</b>
Adresse	<b>0x2E<sub>H</sub>/46<sub>D</sub>/0x3245<sub>ASCII</sub></b>
Tiefe	10 Byte / 80 Bit
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

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	<b>FSP047_Reserved</b>
Adresse	<b>0x2F<sub>H</sub>/47<sub>D</sub>/0x3246<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP048_Reserved</b>
Adresse	<b>0x30<sub>H</sub>/48<sub>D</sub>/0x3330<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP049_Reserved</b>
Adresse	<b>0x31<sub>H</sub>/49<sub>D</sub>/0x3331<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP050_ModuleSupplyValues</b>
Adresse	<b>0x32<sub>H</sub>/50<sub>D</sub>/0x3332<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Liefert die Betriebsspannungen des Moduls.



---

Name	<b>FSP051_ModulePotiValues</b>
Adresse	<b>0x33<sub>H</sub>/51<sub>D</sub>/0x3333<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Liefert die Potispannungen des Moduls.

---

Name	<b>FSP052_ModuleComparatorValues</b>
Adresse	<b>0x34<sub>H</sub>/52<sub>D</sub>/0x3334<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Liefert die Komparatorspannungen des Moduls.

---

Name	<b>FSP053_ModuleTemperatures</b>
Adresse	<b>0x35<sub>H</sub>/53<sub>D</sub>/0x3335<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen
Reset	0x(siehe Beschreibung) <sub>H</sub>

Liefert verschiedenen Temperaturen des Moduls.

Sofern die Temperaturschwellen zur Laufzeit geändert werden sollen, werden diese im Standard FSP 054 hinterlegt.

---

Name	<b>FSP054_ ModuleTemperaturesComparisonThresholds</b>
Adresse	<b>0x36<sub>H</sub>/54<sub>D</sub>/0x3336<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

Stellt die Vergleichswerte (Schwellen) für verschiedene Temperaturen des Moduls zur Verfügung bei denen ein Alarm erfolgen soll. Hiermit lassen sich Standardwerte im Modul zur Laufzeit ändern.

Name	<b>FSP055_WaterFlowMeasurement</b>
Adresse	<b>0x37<sub>H</sub>/55<sub>D</sub>/0x3337<sub>ASCII</sub></b>
Tiefe	modulabhängig
I/O	lesen / schreiben
Reset	0x(siehe Beschreibung) <sub>H</sub>

Sind Wasserwächter im Modul vorhanden, werden diese über verschiedene Parameter beschrieben um deren aktuellen Durchfluss zu bestimmen und dazustellen.

FSPxxx\_WaterFlowThresholds, wird mit den unteren und oberen Schwellwerten für den Durchfluss in uSekunden zwischen zwei Pulsen beschrieben. Zu beachten ist, dass der untere Grenzwert die minimal zulässige Zeitdauer in uSekunden angibt, die zwischen zwei Pulsen liegen darf, damit aber den maximalen Durchfluss beschreibt. Je kleiner die Zeitspanne zwischen zwei Pulsen ist, desto mehr Wasser fließt durch den Wasserwächter. Umgekehrt beschreibt der obere Schwellwert mit maximal zulässiger Zeitdauer in uSekunden, die zwischen zwei Pulsen liegen darf und damit den minimalen Durchfluss. Je größer die Zeitspanne zwischen zwei Pulsen ist, desto weniger Wasser fließt durch den Wasserwächter.

FSPxxx\_InvertedWaterFlow\_TimePeriodeBetweenTwoPulses\_in\_us, gibt die Zeitdauer zwischen zwei Messimpulsen des Wasserwächters in uSekunden zurück.

Diese FSP können beliebig im Modul zu finden sein.

Daher beschreibt FSP055 u. A. deren FSP-Adressen.

- [...] ggf. Index-Typ \*) weitere Wasserwächter
- [31..28] ggf. Index-Typ \*) des 2. Wasserwächters
- [27..24] Index-Typ \*) des 1. Wasserwächters
- [23..16] Anzahl der verfügbaren Wasserwächterkanäle
- [15..8] Adresse des FSPxxx\_InvertedWaterFlow\_TimePeriodeBetweenTwoPulses\_in\_us
- [7..0] Adresse des FSPxxx\_WaterFlowThresholds

\*) Indextypen

0	Float_Type_or_norm_Interlock	(normaler Schaltkontakt, kein Durchfluss übermittelt)
1	Customized_Values	(Benutzerwerte)
2	RRI_010xxx020_2I	(GHM-Hornsberg)
3	RRI_010xxx050_12I	(GHM-Hornsberg)
4	RRI_010xxx070_14I	(GHM-Hornsberg)
5	RRI_025xxx180_36I	(GHM-Hornsberg)
6	RRI_025xxx120_72I	(GHM-Hornsberg)
7	RRI_010xxx160_120I	(GHM-Hornsberg)
8	DRS_9250XXF300	(Kobold)

---

Name	<b>FSP056_Reserved</b>
Adresse	<b>0x38<sub>H</sub>/56<sub>D</sub>/0x3338<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

---

Name	<b>FSP057_Reserved</b>
Adresse	<b>0x39<sub>H</sub>/57<sub>D</sub>/0x3339<sub>ASCII</sub></b>
Tiefe	-
I/O	-
Reset	0x(siehe Beschreibung) <sub>H</sub>

Dieser FSP ist für zukünftige Erweiterungen reserviert

Name	<b>FSP058_ParameterChecksumValue</b>
Adresse	<b>0x3A<sub>H</sub>/58<sub>D</sub>/0x3341<sub>ASCII</sub></b>
Tiefe	3 Byte / 24 Bit
I/O	lesen / schreiben
Reset	0x00_0000 <sub>H</sub>

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 zum Modul.



Name	<b>FSP059_ParameterChecksumValueCalculated</b>
Adresse	<b>0x3B<sub>H</sub>/59<sub>D</sub>/0x3342<sub>ASCII</sub></b>
Tiefe	3 Byte / 24 Bit
I/O	lesen
Reset	0x00_0000 <sub>H</sub>

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 Prüfsumme der Datenübertragung zum Modul